Advances in Deep Active Learning and Synergies with Semi-Supervision

Dissertation zur Erlangung des Doktorgrades
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

vorgelegt von

Sandra Gilhuber

München, den 27.08.2024

Tag der Einreichung: 27.08.2024

Erstgutachter: Prof. Dr. Thomas Seidl
Zweitgutachter: Prof. Dr. Bernhard Sick
Drittgutachter: Prof. Dr. Martin Ester

Tag der Disputation: 26.02.2025

Eidesstattliche Versicherung

(siehe Promotionsordnung vom 12.07.2011, § 8, Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig und ohne unerlaubte Beihilfe angefertigt wurde.

München, den 27.08.2024

Sandra Gilhuber

Abstract

Supervised deep learning models have successfully enabled the automation of processes and the discovery of valuable insights within large datasets, exceeding the capabilities of humans in analyzing and managing the constantly growing volumes of data. However, the effectiveness of these models largely depends on the availability of sufficient high-quality annotated training data. While collecting unlabeled data is often possible with comparably low effort, labeling it is laborious, time-consuming, and costly. In many domains, such as medical or industrial applications, providing accurate annotations requires specialized expertise, which is both scarce and expensive. It is, therefore, essential to reduce the necessity for manual labeling wherever possible.

In this thesis, we address the challenge of insufficient and costly annotations. In particular, we contribute to the field of deep active learning. Unlike traditional approaches that passively rely on pre-labeled data, active learning employs an iterative process alternating between training and labeling. By utilizing the model to decide which instances are most useful for its learning process, the performance is enhanced with a smaller amount of labeled data. Semi-supervised learning is a related field dealing with limited labeled data, which aims to improve models by leveraging both labeled and unlabeled data. Our contributions include new methods and insights into active learning as well as its combination with semi-supervised learning to exploit the strength of both.

Modern deep active learning strategies typically combine model uncertainty with sample diversity to avoid labeling data with redundant information. However, ensuring diversity by calculating distances in learned representations is computationally expensive, particularly for complex, high-dimensional neural networks. Our first contributions address this limitation. We propose using the prediction probabilities to simultaneously select diverse and uncertain instances, substantially accelerating query selection and returning a qualitative query set. Our method proves effective for both tabular and image classification, being superior to competitors in label and time efficiency.

Our next contribution focuses on active learning for node classification. The edges in a graph provide valuable insights into both the importance of individual nodes and the overall graph structure. Hence, it is essential to consider them when actively selecting the most useful instances for labeling. In our work, we introduce a novel active learning

method for node classification that leverages diffusion-based graph heuristics in multiple ways for graph learning as well as actively querying nodes for labeling. In contrast to existing methods, our approach demonstrates robust performance across diverse datasets and consistently surpasses random sampling. Moreover, due to pre-computations, it is faster than competitors.

Finally, we turn our attention to the task of image classification with a particular focus on the combination of techniques from semi-supervised learning and active learning. Our first contribution in this domain proposes a novel active pseudo-labeling approach. We show that false pseudo-labels often occur during the initial iterations where label information is particularly sparse, resulting in long-term negative effects due to confirmation bias. To mitigate this, we propose a solution to refine the pseudo-labels produced by a model based on their consistency with predictions of a second model, considerably improving prediction accuracy. In our last contribution, we analyze the effects of confirmation bias in semi-supervised learning when faced with datasets comprising challenging characteristics as they appear frequently in real-world data. In particular, we consider a high imbalance within and between classes as well as a high similarity between classes. We demonstrate the limitations of semi-supervised methods in overcoming confirmation bias when the data is randomly and passively labeled. By choosing better data samples through active learning, we discuss how confirmation bias can be mitigated, showcasing the potential of combining semi-supervised learning and active learning in the presence of common real-world data challenges.

Zusammenfassung

Supervised Deep Learning Modelle haben erfolgreich die Automatisierung von Prozessen und die Entdeckung wertvoller Erkenntnisse in großen Datensätzen ermöglicht. Sie übertreffen damit die Fähigkeiten des Menschen bei der Analyse und Verwaltung der ständig wachsenden Datenmengen. Die Effektivität dieser Modelle hängt jedoch weitgehend von der Verfügbarkeit einer ausreichenden Anzahl qualitativ hochwertiger annotierter Trainingsdaten ab. Während die Erhebung von nicht gelabelten Daten oft mit vergleichsweise geringem Aufwand möglich ist, ist das Labeling mühsam, zeitaufwändig und kostspielig. In vielen Bereichen, wie beispielsweise in medizinischen oder industriellen Applikationen, erfordert die Bereitstellung präziser Annotationen spezielles Fachwissen, welches sowohl knapp als auch teuer ist. Daher ist es wichtig, die Notwendigkeit der manuellen Annotationen so weit wie möglich zu reduzieren.

In dieser Arbeit befassen wir uns mit dem Problem von unzureichenden und kostspieligen Annotationen. Insbesondere leisten wir einen Beitrag zum Bereich des Deep Active Learning. Im Gegensatz zu traditionellen Ansätzen, die sich passiv auf zuvor gelabelte Daten verlassen, wird beim Active Learning ein iterativer Prozess eingesetzt, bei dem sich Training und Labeling abwechseln. Indem das Modell entscheidet, welche Instanzen für seinen Lernprozess am nützlichsten sind, wird die Leistung mit einer geringeren Menge an annotierten Daten verbessert. Semi-supervised Learning ist ein verwandtes Gebiet, das sich ebenfalls mit wenig annotatierten Daten befasst und darauf abzielt, Modelle zu optimieren, indem sowohl gelabelte als auch ungelabelte Daten genutzt werden. Unsere Beiträge umfassen neue Methoden und Einblicke in Active Learning sowie dessen Kombination mit Semi-supervised Learning, um die Stärken beider Verfahren zu nutzen.

Moderne Strategien für Deep Active Learning kombinieren in der Regel Modellunsicherheit mit Diversität, um die Annotation von Daten mit redundanten Informationen zu vermeiden. Die Sicherstellung der Diversität durch die Berechnung von Distanzen in gelernten Repräsentationen ist jedoch rechenintensiv, insbesondere bei komplexen, hochdimensionalen neuronalen Netzen. Unsere ersten Beiträge befassen sich mit dieser Einschränkung. Wir schlagen vor, die Vorhersagewahrscheinlichkeiten eines neuronalen Netzes zu verwenden, um gleichzeitig vielfältige und unsichere Instanzen auszuwählen,

was die Abfrageauswahl erheblich beschleunigt und eine qualitative Abfragemenge liefert. Unsere Methode erweist sich sowohl für die Klassifizierung von tabellarischen Daten als auch von Bildern als effektiv und ist in Bezug auf Labeling- und Zeiteffizienz der Konkurrenz überlegen.

Unser nächster Beitrag konzentriert sich auf Active Learning für die Klassifizierung von Knoten in einem Graphen. Die Kanten eines Graphen bieten wertvolle Einblicke sowohl in die Bedeutung einzelner Knoten als auch in die Gesamtstruktur des Graphen. Daher ist es wichtig, sie bei der aktiven Auswahl der nützlichsten Instanzen für das Labeling zu berücksichtigen. In unserer Arbeit stellen wir eine neuartige Active Learning Methode für Knotenklassifizierung vor, die diffusionsbasierte Graphenheuristiken auf vielfältige Weise für das Training und die aktive Abfrage von Knoten nutzt. Im Gegensatz zu existierenden Methoden zeigt unser Ansatz eine robuste Leistung über verschiedene Datensätze hinweg und übertrifft durchweg eine zufällige Auswahl. Außerdem ist er aufgrund von Vorberechnungen schneller als bisherige Verfahren.

Schließlich wenden wir uns der Aufgabe der Bildklassifizierung zu, wobei wir uns besonders auf die Kombination von Techniken des Semi-supervised Learning und des Active Learning konzentrieren. Unser erster Beitrag in diesem Bereich schlägt eine neuartige Kombination aus Active Learning und Pseudo-Labeling vor. Wir zeigen, dass falsche Pseudo-Labels oft während der ersten Iterationen auftreten, wenn die annotierten Daten besonders spärlich sind, was zu langfristigen negativen Effekten aufgrund des sogenannten Confirmation Bias führt. Um dies zu entschärfen, schlagen wir eine Lösung vor, bei der die Pseudo-Labels auf der Grundlage ihrer Konsistenz mit den Vorhersagen eines zweiten Netzwerks verfeinert werden, wodurch die Vorhersagegenauigkeit erheblich verbessert wird. In unserem letzten Beitrag analysieren wir die Auswirkungen von Confirmation Bias beim Semi-supervised Learning, wenn es mit Datensätzen konfrontiert wird, die schwierige Merkmale enthalten, wie sie häufig in realen Daten vorkommen. Insbesondere betrachten wir ein hohes Ungleichgewicht innerhalb oder zwischen den Klassen oder eine hohe Ähnlichkeit zwischen den Klassen. Wir zeigen die Grenzen von Semi-supervised Learning bei der Überwindung von Confirmation Bias, wenn die Daten zufällig und passiv annotiert sind. Durch die Auswahl besserer gelabelter Daten durch Active Learning erörtern wir, wie Confirmation Bias abgemildert werden kann, und zeigen das Potenzial der Kombination von Semi-supervised Learning und Active Learning in der Gegenwart von realistischen Datenherausforderungen auf.

Acknowledgements

This work would not have been possible without the help and guidance of many great people. I would therefore like to take this opportunity to say a special thank you to ...

... my supervisor, Prof. Dr. Thomas Seidl, for his constant support, encouragement, and advice over the past years. Thank you for giving me the opportunity to obtain my Ph.D. under your supervision.

... Prof. Dr. Bernhard Sick and Prof. Dr. Martin Ester for their willingness to spend their valuable time on reviewing this thesis.

... my colleagues, co-authors, and friends at the Database Systems and Data Mining group and other institutes who have accompanied and supported me in many different ways over the past years. Particularly, I want to thank Anna, Niklas, Rasmus, Philipp, Yunpu, Julian, Christian, Max, Maxi, Collin, Matthias, Joao, and Susanne.

... my loving and supporting family Grete, Sigi, Liza, and Anna, who have been by my side and supported me my whole life.

... my beloved husband Michael, who motivated and supported me, made me laugh when I needed it, and is always there for me. Thank you also to our unborn son, who reminded me to take breaks during the writing of this thesis but also encouraged me to complete it before his arrival. I am looking forward to our family and the start of a new journey.

Contents

Chapter 1

Introduction

Over the past decades, artificial intelligence (AI) has attracted substantial attention and has become a part of our daily lives. Particularly, the subfields of supervised machine learning and deep learning have demonstrated a remarkable ability to learn from large amounts of data to solve complex tasks, both in research and in various applications, including healthcare [72], automotive [67], finance [45], and agriculture [50]. However, their performance often depends on the availability of sufficient high-quality training data. This observation was also pointed out by Andrew Ng: "Data is the food of AI" [83]. Studies report that data collection, preparation, and cleansing can take up to 80% of the activities when developing machine learning methods for real applications [83, 104]. As a result, data-centric AI emerged as a whole branch of research. In contrast to the model-centric approach of developing new model architectures and algorithms, data-centric AI focuses on improving the data quality and quantity to enhance model performance [57, 117].

A central part of the data collection pipeline is data labeling. Supervised models learn from labeled training data consisting of input-output pairs to recognize patterns and make predictions on new, unseen data. The outputs are known as labels or annotations. To exploit the full potential of many deep learning methods, a sufficient amount of such labeled training data is required. In today's digital age, we experience rapid growth of data generated by individuals and industries, and in many scenarios, collecting unlabeled data is unproblematic and relatively inexpensive [23, 30]. However, humans usually have to assign annotations manually. This poses a severe challenge for developing and deploying deep learning since manual labeling is time-consuming and expensive, especially if domain experts are required to provide accurate annotations. For example, labeling medical images can only be done by qualified doctors, or the detection of defects in industrial machines requires the expertise of specialized engineers. The associated labor and costs of manual labeling limit the applicability of deep learning in real-world applications. Therefore, methods enabling the training of strong models with

fewer labels, such as active learning and semi-supervised learning, are essential.

The idea of active learning is to label only the most valuable data instances in a large unlabeled pool that bring the greatest benefit to the learning progress of a model. It alternates between model training and annotating the most informative data instances until a predefined budget or model performance is reached. Through this process, only a small portion of the available data is labeled, effectively reducing annotation costs. Over the past years, numerous different active learning strategies have been developed and successfully applied to various domains, such as computer vision [17, 48], graph learning [12, 69, 77], natural language processing [5, 93, 121], or audio processing [70, 84, 85]. Active learning research is largely concerned with designing new query methods that decide which instances are worth labeling and which are not. Active learning strategies can be divided into uncertainty-based, representativeness-based, diversity-based, and hybrid methods [61]. Uncertainty sampling assumes that instances with high model uncertainty contain a particularly large amount of new information and are, therefore, valuable for the model. Representativeness-based sampling is based on the assumption that typical instances are valuable because they are similar to many other unlabeled instances in the data. Diversity sampling focuses on minimizing redundancies in the labeled data to approximate the overall distribution of the data. Most of the research focuses on hybrid techniques combining several query types into one selection, as these have been shown to yield more robust results for varying and unique characteristics of datasets and tasks [1, 6, 24, 80, 82].

Semi-supervised learning is a related machine learning paradigm that addresses the challenge of high annotation costs. In addition to utilizing the limited labeled data, semi-supervised learning seeks to automatically extract information from a large pool of unlabeled data and integrate it into the training process to enhance model performance. Common techniques involve pseudo-labeling, where the model produces artificial labels for unlabeled data, or consistency regularization, where the model enforces slightly different versions of the same input to produce similar outputs [23]. Active learning and semi-supervised learning propose different strategies to address a similar problem and can naturally be combined. While active learning improves the quality of supervision by extending the labeled pool with a few particularly valuable instances, semi-supervised learning automatically derives knowledge from the large unlabeled pool to improve the model performance.

The scope of this thesis and the included contributions lie within the above-described areas of active and semi-supervised learning. In the following section, the research scope of the contributions is introduced in more detail.

1.1 Research Scope

The goal of a deep active learning method is to improve model performance with fewer labels, also referred to as label efficiency. However, existing methods often neglect the query time needed to determine the set of instances sent to the annotator. In particular, common diversity-based methods targeted at image classification often use high-dimensional latent features to measure distances between all instances in the unlabeled pool, which is computationally demanding [10, 56]. However, large query times can be prohibitive, for instance, when experts are only available for a limited time. To address this problem, we present two contributions in this thesis. First, we investigate the acceleration of established diversity-based methods by exchanging latent features with prediction probabilities [37]. Second, we propose a novel query method named FALCUN [41] that combines uncertainty and diversity in the probability space and yields time- and label-efficient results on various tabular and image classification tasks. The contributions are explained in more detail in Section 3.1.

Next, we consider active learning for node classification. In graph active learning, it is common to utilize the graph structure to derive information on the usefulness of nodes. To yield robust results on diverse datasets, an ongoing challenge is to combine graph-specific selection criteria with other powerful query types, such as uncertainty and diversity-based techniques. However, existing approaches for node classification often rely on hyperparameters, treat the data sampling and training as separate steps, or narrow their focus to limited selection criteria. As a result, they often do not yield consistent results across datasets. Moreover, utilizing multiple query types can negatively affect the runtimes of active learning methods. To address these limitations, we present a novel active learning method for node classification called DiffusAL [39] and present its key contributions in Section 3.2. DiffusAL utilizes diffusion-based heuristics for graph learning and for querying valuable instances for annotation. Experiments on node classification benchmarks show that our approach is more label-efficient and, due to various pre-computations, also more time-efficient than existing methods.

Finally, we turn our attention to the combination of active learning and semi-supervised learning for image classification. A known challenge in semi-supervised learning is confirmation bias, where the model repeatedly enforces information it has learned wrong, resulting in degraded model performance [3, 106]. The problem occurs when the model makes unreliable predictions. This can also hinder the effective combination with active learning since the initial labeling information is scarce [27]. In our contribution discussed in Section 3.3, we address this challenge and propose a novel algorithm called VERIPS that combines active learning with pseudo-labeling [38]. VERIPS uses a verification step to filter pseudo-labels based on a second network. This mechanism discards many wrong pseudo-labels at the beginning of the active learning loop, thereby

effectively mitigating confirmation bias.

Our final contribution [40], discussed in Section 3.4, investigates the applicability of active learning and semi-supervised learning when facing different kinds of challenging datasets. In our contribution, we identify three relevant challenges for combining active and semi-supervised methods that are present in many real-world data sets. Existing research mainly overlooked these challenges since it focused on specific benchmark data that is distinctly different from data encountered in many real-world applications. More specifically, we identified the following challenges: between-class imbalance (BCI), between-class similarity (BCS), and within-class imbalance (WCI). Then, we demonstrate how semi-supervised learning methods fail on these challenges because of confirmation bias when the labeled data is collected randomly. Moreover, we provide a proof-of-concept showcasing how the usage of active learning instead of random sampling can help to overcome confirmation bias.

1.2 Thesis Structure

The remainder of this dissertation is structured as follows: Chapter 2 introduces relevant background knowledge. First, the fundamentals of machine learning are presented in Section 2.1. This includes an introduction to the notation, a brief discussion of the tasks relevant to this thesis, some basics about deep neural networks, and an overview of machine learning with limited labeled data. In Section 2.2, we discuss semi-supervised learning and relevant approaches. Section 2.3 introduces active learning, addresses considerations relevant to deep active learning, and provides an overview of different query types and related work. Section 2.4 concludes the second chapter and discusses the combination of active and semi-supervised learning. Chapter 3 first gives an overview of the publications that are included in this thesis and subsequently explains their main contributions in more detail in separate sections (Sections 3.1 to 3.4). To conclude this thesis, we give a brief summary of our contributions and indicate directions for future research in Chapter 4. The original publications that are subject to this dissertation and corresponding supplements are included in the appendix.

Chapter 2

Background

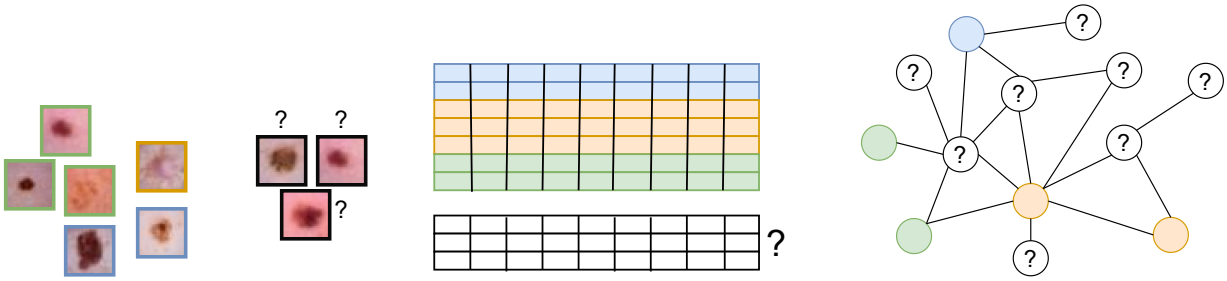
In this chapter, we provide an overview of the relevant background essential for understanding the main contributions and situating them within existing literature. We begin by giving a short introduction to machine learning in Section 2.1. In the remaining sections, we discuss different subfields of limited labeled learning. First, we give an overview of semi-supervised learning in Section 2.2. Then, we shift our focus to active learning in Section 2.3. Finally, we briefly discuss the combination of active and semi-supervised learning in Section 2.4.

2.1 Fundamentals of Machine Learning

This section first introduces the notation used throughout the thesis and provides an overview of the machine learning tasks addressed. Then, we present the basics of deep neural networks, which are primarily used to solve the targeted problems. Finally, we define the problem of machine learning with limited labeled data. In this broad field, we specifically focus on the two areas of active and semi-supervised learning.

2.1.1 Notation and Tasks

Machine learning tasks can be categorized by the amount of available label information or the data input types. In a fully *supervised learning* setting, the entire available data set consists of pairs $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the input data space and $\mathcal{Y}$ is the output data space. In other words, for each instance $\mathbf{x}$, we also have access to a label y. This contrasts with *unsupervised learning*, where no label information is given, and the goal is to reveal patterns in the unlabeled data without supervision. However, in this thesis, our focus lies on tasks formulated as supervised problems. One of the most common supervised tasks is classification. In this task, the label y can take one of C discrete



(a) Image classification. *(b) Tabular classification.* *(c) Node classification.*

Figure 2.1: *Multi-class classification tasks considered in the thesis. Exemplarily, we show a 3-class classification problem where the colors blue, orange, and green represent different classes. The goal is to learn from the given examples and give accurate predictions for the data marked by question marks. In image and tabular classification, the colored, labeled data is utilized for training, and the test data is unseen. In node classification, the whole graph comprising labeled and unlabeled nodes is utilized in training to predict labels of the nodes marked by a question mark.*

values with $\mathcal{Y} = \{1, 2, \ldots, C\}$. If $C = 2$, this would be called a binary classification problem. In this thesis, we focus on multi-class classification problems with $C > 2$. The goal in classification tasks is to learn a function, also referred to as classification model $f : \mathcal{X} \to \mathcal{Y}$ from the available set of observed training pairs that can accurately predict the label $\hat{y}$ for a given input $\mathbf{x}$. Formally, this can be expressed as: $\hat{y} = f(\mathbf{x}, \theta)$. We use θ to denote the classification model's parameters, learned from the data using an appropriate optimization algorithm. Classification tasks can be categorized based on which data is classified. In this thesis, we consider three tasks, namely *image classification*, *tabular classification*, and *node classification*.

Image classification In image classification, the inputs are images. Each image $\mathbf{x}$ is represented as a tensor of pixel values. For a colored image, this tensor is of the form $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$, where H is the height of the image, W is the width, and C is the number of color channels (usually 3 for RGB images). There are various application areas, such as the classification of X-rays [47] or CT [107] scans for disease diagnosis in a medical context or traffic sign recognition [71] as an aid for autonomous driving. We consider image classification tasks in an inductive setting, where the model is trained on a labeled dataset and then used to predict labels for new, unseen images. An illustration with examples of DermaMNIST, which consists of dermatoscopic images categorizing different diseases [113], is shown in 2.1a.

Tabular classification In tabular classification, the input features are structured in a tabular format. Each row in the table represents a sample $\mathbf{x}$, and columns correspond to different features $\mathbf{x} = (x^1, x^2, \ldots, x^M)$, where M is the number of features. This task is

common in many domains, such as finance, healthcare, or marketing, where data is often stored in relational databases [99]. Features can be numerical, categorical, or a mix of both. When considering tabular data, it might be necessary to pre-process features into a format that can be used by the learning model, such as transforming categorical features into numerical values. Similar to image classification, we consider tabular classification in an inductive setting, where the model is trained on known data and then applied to new samples during inference (See 2.1b).

Node classification Node classification is one of the core tasks within the context of graph data. In this task, the dataset is represented as a graph $G = (V, E)$, where V is the set of nodes and E denotes the set of edges connecting these nodes with $E \subseteq V \times V$. Additionally, we assume that each node $v \in V$ is associated with attributes represented as an M-dimensional feature vector $\mathbf{x} \in \mathbb{R}^M$. The objective is to predict the nodes' labels based on their features and the graph structure. This task is studied in various areas where relations between nodes are crucial, such as social network analysis or molecular biology [7]. Node classification is often conducted in a transductive setting, i.e., the model is trained and tested on the same graph, leveraging both labeled and unlabeled nodes during training to improve performance (See 2.1c). During training, a combination of supervised learning on labeled nodes and unsupervised learning using the edges to unlabeled nodes is utilized, and therefore, this setting can be seen as a special type of semi-supervised learning [15].

2.1.2 Deep Neural Networks

The classification models used in this dissertation are deep neural networks, so we briefly derive some important basics in the following. For a more detailed introduction, we refer the reader to the book of Goodfellow et al. [42]. Deep neural networks form a special group of machine learning models and are widely used to solve the above-described classification tasks due to their capability of processing high-dimensional data and the power of learning arbitrarily complex, non-linear functions.

The most essential architecture, often a building block for more complex network architectures, is the *Multi-Layer Perceptron* (MLP) [42]. An MLP is a parametric function that maps a set of inputs to outputs by chaining many parametric functions. Each sub-function is called a layer of the network, where the first layer is called the input layer, the intermediate layers are called hidden layers, and the last layer is called the output layer. Each layer consists of neurons, which are the most basic unit in an MLP, and each neuron in a layer is connected to every neuron in the subsequent layer. One neuron takes a vector $\mathbf{x}$ as input, multiplies it with a weight vector $\mathbf{w}$, and adds a bias b. This results in a scalar value $z' = \mathbf{x} \cdot \mathbf{w} + b$. $\mathbf{w}$ and b are trainable parameters, meaning they will be

optimized during the training process. To introduce non-linearity, an activation function σ is applied to the output z'. Popular choices are the sigmoid function given as $\frac{1}{1+e^{-z'}}$ or ReLu given as $\max(z, 0)$. The output of a neuron, which is then passed onto the next neuron, is given as:

$$z = \sigma(\mathbf{x} \cdot \mathbf{w} + b) \tag{2.1}$$

We can easily extend the output of a single neuron to a layer-wise expression. Given an MLP comprising K layers, the output of the k-th layer in the network consisting of d_k neurons is denoted as:

$$\mathbf{z}^{(k)} = \sigma(\mathbf{W}^{(k)}\mathbf{z}^{(k-1)} + \mathbf{b}^{(k)}), \tag{2.2}$$

where $\mathbf{z}^{(k-1)}$ is the output vector of the previous layer (or the input vector $\mathbf{x}$ in case of the input layer), $\mathbf{W}^{(k)} \in \mathbb{R}^{d_k \times d_{k-1}}$ is the weight matrix where the i-th row corresponds to the weight vector $\mathbf{w}$ of the i-th neuron, $\mathbf{b} \in \mathbb{R}^{d_k}$ is the bias vector of the k-th layer, and the activation function σ is applied element-wise. The dimensionality of the output layer corresponds to the number of classes C of the classification task, and the activation function is usually the softmax function that transforms the outputs into a probability distribution. We denote the predicted probabilities for an input $\mathbf{x}$ forwarded through the network as

$$\mathbf{p} = \text{softmax}(\mathbf{p}') = \frac{e^{p'_i}}{\sum_{j=1}^{C} e^{p'_j}}, \tag{2.3}$$

where $\mathbf{p}'$ is the output of the output layer before applying the softmax function. The final prediction $\hat{y}$ of the network is the class that received the highest predicted probability, i.e., $\hat{y} = f(\mathbf{x}, \theta) = \arg\max_C \mathbf{p}$, where $\theta = \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \ldots, \mathbf{W}^{(K)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \ldots, \mathbf{b}^{(K)}\}$ represents the model parameters.

The MLP is the simplest form of a neural network, and many other forms designed for certain tasks or data types have been introduced. Many of them try to utilize certain assumptions, also referred to as inductive bias, to better generalize to novel data and not only memorize the observations on which they are trained. For instance, *Convolutional Neural Networks* (CNNs) [59] are a specialized type of neural architecture particularly suited for processing grid-like data, such as images. A typical CNN architecture includes convolutional layers, pooling layers, and fully connected layers. The convolutional layer assumes that local areas of pixels in images are highly correlated and applies a set of learnable filters (or kernels) across the input matrix to produce feature maps. After the convolutional layer, it usually follows a pooling layer, which reduces the dimensionality of the feature map, emphasizing dominant features within specific regions. Other architectures are *recurrent neural networks* [92], for modeling sequential data, or *graph neural networks* (GNNs) [43, 9, 16], which are suited for graph data. However, our contributions are more concerned with a data-centric view rather than a model-centric one,

and therefore, giving a more detailed overview of different architectures is out of scope for this thesis. We refer to [42] for an in-depth overview of deep learning and different network architectures.

Training

Training a neural network usually involves forward propagation to compute predictions and backpropagation to update model parameters based on the prediction error. The forward step is passing an input $\mathbf{x}$ through the network $f(\mathbf{x}, \theta)$, which finally returns the prediction $\mathbf{p}$. Then, the prediction error between the model prediction $\mathbf{p}$ and the true target y is calculated by using a suitable, differentiable loss function $l(\mathbf{p}, y)$. A common choice for the loss function used in multi-class classification is the cross-entropy loss, which is defined as:

$$l_{CE}(\mathbf{p}, y) = -\sum_{i=1}^{C} \mathbb{1}(y = i) \log(p_i), \tag{2.4}$$

where $\mathbb{1}(y = i)$ is the indicator function which equals 1 if the true class label y is equal to class i and 0 otherwise. During the backward pass, the model adjusts its weights and biases according to the errors made in the prediction. More precisely, the partial derivatives (or gradients) of the loss function with respect to the model parameters θ comprising the weights and biases are calculated using the chain rule. Then, the parameters are updated by the negative gradient, thereby minimizing the loss function. This process is then repeated until convergence or reaching a predefined number of training iterations.

Learned Representations

Before the era of deep learning, most machine learning applications required manually designed features. However, manual feature engineering for complex tasks requires specialized knowledge and is time-consuming and laborious [42]. In contrast, a central and important advancement of neural networks is that representations are automatically learned in such a way that they represent the input space better for solving the specific task. The modular structure of neural networks allows us to extract intermediate outputs, which we can interpret as new representations of the original input data [42]. These features capture higher-level representations of the input data, and they can be useful on their own. This includes visualization, similarity calculations between objects, and further analysis. For example, in a CNN for image classification, the activations of the last convolutional layer often represent abstract features learned by the network, such as textures or object parts. These learned representations are commonly used to

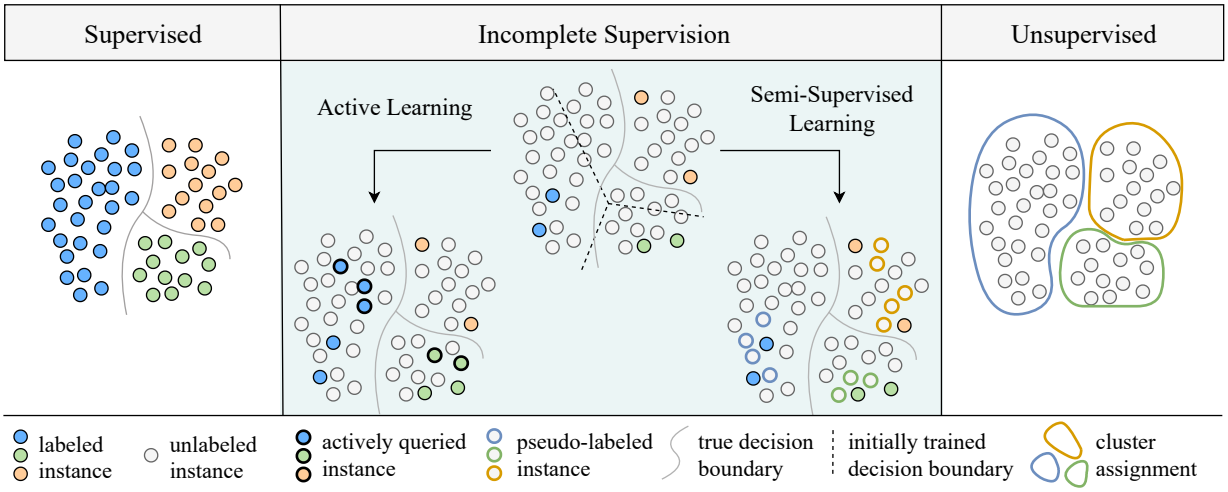


Figure 2.2: *Overview of the machine learning types covered in the thesis at the intersection of supervised and unsupervised learning, illustrated in an example. We consider settings of incomplete supervision where only a small subset of the available data is labeled: Active learning and semi-supervised learning. They lie at the intersection of supervised learning, where all data is labeled, and unsupervised learning, where no labels are given. Active learning increases the labeled pool by annotating the most informative instances. Besides labeled data, semi-supervised learning utilizes the unlabeled data for training, e.g., by pseudo-labeling instances of high model confidence.*

compare instances to make implications for enhanced training routines, e.g., consistency regularization in semi-supervised learning or to derive information on which instances are important to be labeled for active learning.

2.1.3 Machine Learning with Limited Labeled Data

One of the biggest limitations of fully supervised learning is that it requires access to a completely labeled dataset. However, labeling the complete training data is challenging in practice. For instance, certain tasks may require special domain expertise, such as finding potential diseases in medical images [17, 79]. Other annotation tasks are time-intensive due to more complex labeling types, such as box-level or pixel-level annotations, in contrast to image-level annotations [62]. Unlike images, other data types, such as tabular data or nodes in a graph, are not as visually intuitive and, hence, may be more difficult and time-consuming to annotate. Consequently, due to the high manual effort and resulting costs, annotations should be reduced whenever possible. To mitigate the necessity of vast amounts of manual annotations, weakly supervised learning aims to ”weaken” the dependence on supervision. Weakly supervised learning can be further categorized into *incomplete*, *inaccurate*, and *inexact* supervision [124]. Incomplete refers to a setting where labels are only partially available, inaccurate indicates that labels might

be faulty, and inexact refers to the setting where the label is coarse-grained [88]. In this thesis, we focus on incomplete supervision. Here, the goal is to train an accurate machine learning model f with only a limited amount of labeled data. In such scenarios, we distinguish between the labeled dataset $\mathcal{L}$ of size N_l consisting of labeled data pairs $(\mathbf{x}, y)$ and the unlabeled dataset $\mathcal{U} = \mathcal{X} \setminus \mathcal{L}$ of size N_u where only the data sample $\mathbf{x}$ is given without any label. Note that the labeled pool is usually much smaller than the unlabeled pool, i.e., $N_l \ll N_u$. The contributions presented in this thesis are mainly concerned with active learning and partly with combinations with semi-supervised learning, both paradigms within the broader field of limited labeled learning. In semi-supervised learning, a fixed number of labels is given in advance, and the aim is to utilize both labeled and unlabeled data in conjunction to train a strong classifier.[1] In contrast, active learning identifies the most beneficial instances and incrementally increases the number of labels over multiple rounds. The primary goal of both active learning and semi-supervised learning is to improve model performance with fewer labeled instances, i.e., to increase *label efficiency*. An overview of the addressed machine learning types and an illustrative example is shown in Figure 2.2. In the following sections, we will discuss semi-supervised learning and active learning, as well as their combination, in more detail.

2.2 Semi-Supervised Learning

Semi-supervised learning (SSL) lies at the intersection of unsupervised and supervised learning and leverages a small labeled pool together with a large unlabeled data pool to train models [23]. The goal is to reveal patterns within the unlabeled data to improve model performance beyond plain supervision without extending manual annotation efforts. A core aspect of most SSL techniques is that they build upon the following three main assumptions [109]:

1. *Smoothness assumption*: Two instances $\mathbf{x}_1, \mathbf{x}_2$ that are close within a region of high density have similar labels y_1, y_2.

2. *Low-density assumption*: The decision boundary of the classifier $f(\mathbf{x}, \theta)$ should pass through low-density regions.

3. *Manifold assumption*: The high-dimensional input space consists of multiple lower-dimensional manifolds and instances $\mathbf{x}_1, \mathbf{x}_2$ that lie on the same manifold and have similar labels y_1, y_2.

[1] Note that there is a wide field of semi-supervised tasks. For instance, in semi-supervised or constraint-based clustering, the task is formulated from an unsupervised perspective, and supervised information comes in the form of pairwise constraints. However, in this thesis, we focus on semi-supervised classification, where the labels are instance-level class assignments.

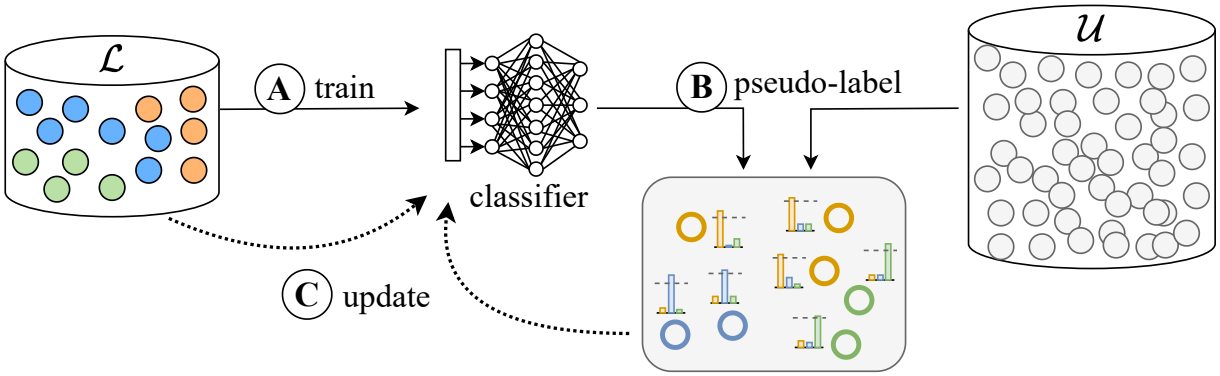


Figure 2.3: *Illustration of pseudo labeling. Ⓐ Train the classifier using the labeled pool $\mathcal{L}$. Ⓑ Pseudo-label unlabeled data where prediction exceeds threshold. Ⓒ Update classifier using labeled and pseudo-label data. Repeat pseudo-label selection and model-refitting until convergence.*

Existing SSL techniques utilize one or more of these assumptions to derive meaningful insights from the unlabeled data to gradually update the model without requiring more manual labels.

2.2.1 Methods

In this thesis, we focus on modern SSL approaches that are based on deep neural networks. In particular, we will explain pseudo-labeling, consistency regularization, and hybrid methods, which have been studied extensively over the past few years, especially in image classification, and are most relevant to our contributions. A thorough overview of other SSL methods can be found in the comprehensive SSL surveys of [23] and [109].

Pseudo-Labeling

Pseudo-labeling (sometimes referred to as self-training) is one of the earliest SSL techniques. Though the idea has been around for many decades [94, 28, 2] before the era of deep learning, it is still frequently used and combined in recent approaches due to its simplicity and effectiveness. In pseudo-labeling, the model uses its own predictions $\hat{y}$ on the unlabeled data $\mathcal{U}$ as if they were real annotated targets. These so-called pseudo-labels are then treated as ground truth and involved in further training of the model. 2.3 illustrates the general approach of pseudo-labeling. After fitting an initial model on the labeled dataset $\mathcal{L}$ (step Ⓐ), the model is used to pseudo-label some of the unlabeled data, for instance using a confidence threshold (step Ⓑ). Then, the model is updated using the labeled and pseudo-labeled data (step Ⓒ). After re-fitting the model, new pseudo-labels are generated, and the process is repeated until reaching a certain stopping criterion.

The general idea of pseudo-labeling has been translated to deep learning in the work

of [60]. Many variants and combinations to improve the pseudo-label selection and assignment have been proposed since then. A common approach is to pseudo-label instances $\mathbf{x} \in \mathcal{U}$ whose most likely prediction probability $\mathbf{p}(\hat{y}_1) = max(\mathbf{p})$ exceeds a predefined threshold τ. Many approaches gradually increase the impact of the pseudo-labels by adjusting the weighting of the pseudo-labeled loss [60, 3, 106]. Other approaches to select more reliable pseudo-labels involve utilizing more sophisticated uncertainty metrics [89] or class-specific, flexible thresholds [119].

Consistency Regularization

Consistency regularization is based on the smoothness assumptions or the manifold assumption [114], i.e., slightly perturbed versions of the same input should yield consistent predictions. Consistency regularization techniques enforce the network to model this relation. This can be achieved by applying perturbations such as data augmentations or noise to the input data and ensuring the model's predictions remain stable by adding a dedicated consistency loss to the final loss term. Concretely, given an unlabeled instance $\mathbf{x} \in \mathcal{U}$ and a perturbed version of the instance $\hat{\mathbf{x}}$, the objective is to minimize $d(f(\mathbf{x}), f(\hat{\mathbf{x}}))$, where $d(\cdot, \cdot)$ is a suitable distance function measuring the discrepancy between the two model outputs [76]. Popular choices are the mean squared error, the Kullback-Leibler divergence, or the Jensen-Shannon divergence [78].

The Π-Model [58] exploits the fact that neural networks can produce different outputs for the same instance during training due to common regularization techniques such as dropout or adding noise. There, an additional loss term weighted by a hyperparameter is added. This additional loss calculates the mean squared error between the different outputs for each instance in the unlabeled pool produced by the same model. However, the different stochastic predictions of the Π-Model may be unstable due to rapid changes over the training course [76]. To stabilize predictions, the mean teacher [106] approach utilizes a student and a teacher model to obtain different outputs for the same input and match the predictions for training. The student network is the primary network, and the teacher model maintains an exponential moving average of the parameters from previous training steps. Another popular and powerful technique to create different versions of the same input, especially in SSL for computer vision, is data augmentation. Common data augmentations are scaling, rotation, random noise, or flipping. For example, VAT [74] applies a perturbation to the original input instance to obtain different predictions instead of relying on the stochasticity of the neural network and enforcing consistency between them. MixUp [120] proposes to generate new training samples by linearly interpolating between pairs of inputs and their corresponding labels. However, data augmentations may not be straightforwardly applicable to other data types, such as text or tabular data, or special domain expertise might be required to maintain the

meaning of the original data [89].

Hybrid

Hybrid methods combine different concepts into a single, holistic framework to improve performance. MixMatch [14] guesses labels for each unlabeled instance by averaging several predictions of their augmented versions and subsequent sharpening. Then it uses MixUp [120] to create augmented versions of the instances and their guessed labels, and includes these mixed-up samples in a standard supervised training. ReMixMatch [13] is an improvement of MixMatch. The advancement involves distribution alignment, promoting the marginal distribution of predictions on unlabeled data to closely match the marginal distribution of true labels, and augmentation anchoring, which ensures that the model produces similar outputs for multiple strongly augmented versions of an input instance as a weakly augmented version of it. One of the most popular methods of the past years is FixMatch [102]. FixMatch obtains pseudo-labels for weakly augmented images whose most likely prediction exceeds a threshold and assigns the pseudo-label to strongly augmented versions of the same image using cross-entropy. Using a fixed user-defined threshold for all classes can be suboptimal. Therefore, FlexMatch [119] employs a curriculum pseudo-labeling technique which flexibly adjusts class-specific thresholds as an extension to FixMatch.

2.2.2 Confirmation Bias

Confirmation bias in SSL refers to the model's tendency to reinforce its own incorrect predictions during the learning process [3, 106]. For instance, pseudo-labels are hypothetical labels produced by the model and, therefore, may be wrong. This wrong information is then propagated to subsequent training iterations and can lead to a downward spiral. In severe cases, confirmation bias can even lead to SSL models being less effective than plain supervised learning without using unlabeled data. The problem of confirmation bias specifically occurs in SSL since the training procedure builds upon certain assumptions introduced at the beginning, and these assumptions can be broken. Reasons for that involve a too small labeled pool, the lack of high-quality labeled samples, or generally if the model is overconfident [3]. Another challenge is the class distribution mismatch when the labeled data does not contain all classes that are part of the unlabeled data [76]. To avoid the inclusion of incorrect information, several strategies can be employed, such as confidence thresholding [60, 102], class-specific thresholds [119], ensuring pseudo-labels are consistent under different data augmentations and perturbations [3], or curriculum learning [20]. However, there is a trade-off between fully exploiting the unlabeled data and cautiously avoiding the risk of reinforcing incorrect

predictions through confirmation bias. Developing effective methods that automatically adapt to diverse datasets with varying levels of complexity is difficult. Thus, it remains an ongoing challenge to effectively mitigate confirmation bias without diminishing the potential of leveraging unlabeled data.

2.3 Active Learning

Active learning (AL) addresses the issue of high annotation costs and limited labeled data by intelligently allocating annotation efforts. A central assumption is that not all data is equally important for training. Therefore, the learner can be improved more quickly if it is allowed to choose the data it learns from [96]. Annotation efforts should be focused on instances that contribute the most to model performance, while irrelevant samples should be left unlabeled. Three typical active learning scenarios are considered in the literature: *pool-based active learning*, *stream-based active learning*, and *membership query synthesis* [96]. In *pool-based active learning*, a large unlabeled pool is given in advance, and the active learner queries instances by searching for the most useful ones in that pool. In *stream-based active learning*, the instances continuously arrive at different time stamps, and one must individually decide whether to label an instance or to discard it. *Membership query synthesis* refers to the generation of synthetic instances to augment the labeled pool rather than relying exclusively on instances that are already part of the given data source.

In this thesis, we focus on *pool-based active learning*, the primary area of ongoing research, and detail the general framework in the following subsection. Then, we will highlight key differences between traditional AL and AL in a deep learning context. The last part of this section categorizes different query types, i.e., the function of an active learner that determines which data is labeled and gives an overview of related work.

2.3.1 Pool-based Active Learning

Figure 2.4 depicts the general framework of pool-based active learning. Initially, two distinct pools of data are available for use: an unlabeled pool, denoted as $\mathcal{U}$, and a labeled pool, denoted as $\mathcal{L}$. In an iterative process, the objective is to label a subset of the unlabeled pool that best enhances the classifier's generalization performance. The first step is to train a classifier using $\mathcal{L}$ (step A). Subsequently, the query strategy, which comprises the logic of which instances are considered most beneficial, returns a set of Q query objects from the unlabeled pool by extracting valuable knowledge of the previously trained classifier. The selected query objects are then forwarded to the oracle, which is usually a human annotator (step B). The annotator subsequently provides a label for each instance, and the objects are moved to the labeled pool (step C). This process is

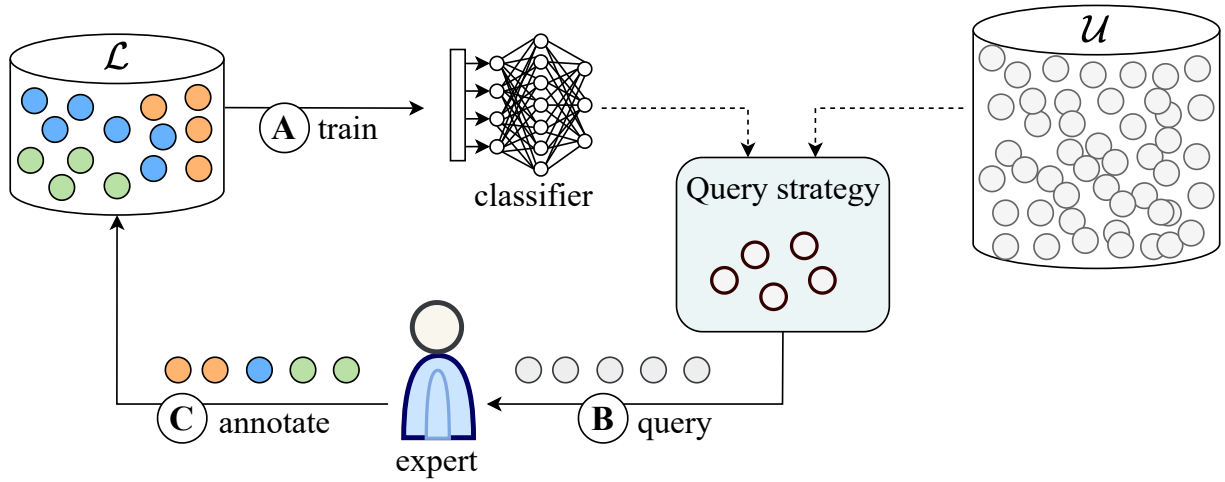


Figure 2.4: *Overview of pool-based active learning. Three steps are repeated until a certain stopping criterion is reached: Ⓐ Train the classifier on the labeled pool $\mathcal{L}$. Ⓑ Use the classifier to query instances from the unlabeled pool $\mathcal{U}$ and send them to the annotator. Ⓒ Receive labels for queried instances and move them to the labeled pool $\mathcal{L}$.*

repeated until a predefined stopping criterion is met. In practice, this criterion is usually either the exhaustion of a predefined labeling budget B or the achievement of a desired model accuracy.

2.3.2 Deep Active Learning

Compared to traditional active learning methods, some aspects are specifically relevant to active learning being applied in conjunction with deep neural networks. In the following, we will explain the most important differences.

Dynamic features

In traditional AL, the features used to measure similarity between inputs are usually fixed or pre-processed. They do not change throughout the course of AL selection. In contrast, in deep AL, features are jointly learned with the classification model. Thus, they dynamically change during the AL process [87]. These learned representations are more expressive and discriminative towards the downstream task than the original input features [42]. Consequently, they are preferred to calculate similarity for representativeness or diversity-based methods. However, this involves similarity calculations that are not pre-computed. Instead, they are re-computed between AL selections, resulting in increased computational costs [61, 118].

Pre-training and Continual Training

Many AL methods fail to outperform random sampling at the beginning of the AL process where the labeling budget is really small. This effect is also referred to as the cold start problem [108, 61]. To avoid the cold start, pre-trained models trained on related domains [32] or using self-supervised pre-training [22] can be used to improve AL in early iterations [75, 61]. Instead of re-training the deep learning model from scratch between AL rounds, concepts of transfer learning can be used to fine-tune the model with the novel instances [80, 61]. While this can effectively reduce training times between AL selections, undesired effects can arise that make learning unstable. For instance, early learned concepts can be forgotten over the course [25]. Therefore, continual training must be considered cautiously.

Batch-Mode

Earlier works often assume the model is updated after selecting and labeling only one single instance [108]. The batch-mode setting refers to selecting multiple instances in batches and sending them to the annotator in parallel. The batch-mode setting became particularly popular in deep active learning since updating the model serially after each annotated instance individually is computationally prohibitive for data-greedy and training-intensive deep learning procedures [87, 6]. Moreover, sending batches of data also allows the labeling process to be parallelized, enabling the simultaneous work of multiple annotators [56]. Although batch acquisition has many benefits, it also introduces new challenges. Since the model is not updated after every chosen instance, AL methods need to make assumptions about how different combinations of chosen samples might influence the importance of other query points. Otherwise, the chosen query set might contain highly similar instances with repetitive information. For example, consider two identical images that both result in the highest uncertainty scores. A strategy based solely on uncertainty would select both for labeling, but doing so would waste valuable annotation resources. Therefore, diversity-enhancing mechanisms to reduce redundancy and information overlap within the query batch are important [6]. We will detail common diversity-based and hybrid techniques in Section 2.3.3 and in Section 2.3.3, respectively.

2.3.3 Query Types

The main distinguishing characteristic between active learning strategies is their acquisition functions, i.e., the function that determines which instances are worth consuming the labeling budget and which are left unlabeled. In the literature, different categorizations of query types (or acquisition types) exist. In the following, we will discuss the

most important concepts used throughout the literature, which are closely aligned with the taxonomy used in [108] and [61]. Namely, we distinguish between uncertainty-based, representativeness-based, diversity-based, and hybrid techniques. An illustrative example for comparing the query types is shown in Figure 2.5.

Uncertainty-based Sampling

The idea of uncertainty-based sampling is based on principles of information theory and assumes that instances about which the model is most uncertain regarding its prediction will provide the most novel information when added to the training data [96]. The naive approach computes an uncertainty score u given the model's predicted probability distribution $\mathbf{p}$ for each instance in the unlabeled pool. Then the most uncertain or the top-k most uncertain instances are selected. Given the prediction probability $\mathbf{p}$ for an instance $\mathbf{x} \in \mathcal{U}$, typical uncertainty estimates are:

- *Least confidence* selects points with the smallest posterior probability for their most likely label $\mathbf{p}(\hat{y_1})$. Least confidence uncertainty is defined as: $u_{lc}(\mathbf{p}) := 1 - \mathbf{p}(\hat{y_1}) \in [\frac{1}{C}, 1]$ [96].

- *Margin* (also known as breaking ties (BT) or best-vs-second-best (BvSB)) selects points that have the smallest difference between the two most likely classes $\mathbf{p}(\hat{y_1})$ and $\mathbf{p}(\hat{y_2})$. Margin uncertainty is defined as: $u_m(\mathbf{p}) := 1 - (\mathbf{p}(\hat{y_1}) - \mathbf{p}(\hat{y_2})) \in [0, 1]$ [90]. Larger values indicate higher uncertainty of the model that its predicted class $\mathbf{p}(\hat{y_1})$ is correct.

- *Entropy* selects points that maximize Shannon entropy [98]. Entropy uncertainty is defined as: $u_e(\mathbf{p}) := -\sum_{i=1}^{C} \mathbf{p}(\hat{y_i}) \log(\mathbf{p}(\hat{y_i})) \in [0, \log C]$ [96]. The higher the entropy, the more uncertain the model is about the prediction for a sample $\mathbf{x}$.

These uncertainty-based techniques are easy to understand and implement. Due to their effectiveness and simplicity, they are often combined with other query types and thus play an important role in many state-of-the-art approaches [82, 49].

Query-by-committee (QBC) approaches measure uncertainty by combining predictions from an ensemble of classifiers (the "committee"). A high disagreement among the predictions of an instance indicates a high uncertainty. QBC methods aim to reduce the reliance on potentially unreliable predictions from a single network [97, 29, 11]. Typical approaches include selecting instances with the largest mean standard deviation over all classes [51], calculating the Kullback-Leibler divergence [118], or applying one of the previously introduced uncertainty heuristics to the mean prediction of all committee members [11]. However, training multiple networks to form a committee greatly increases computational costs.

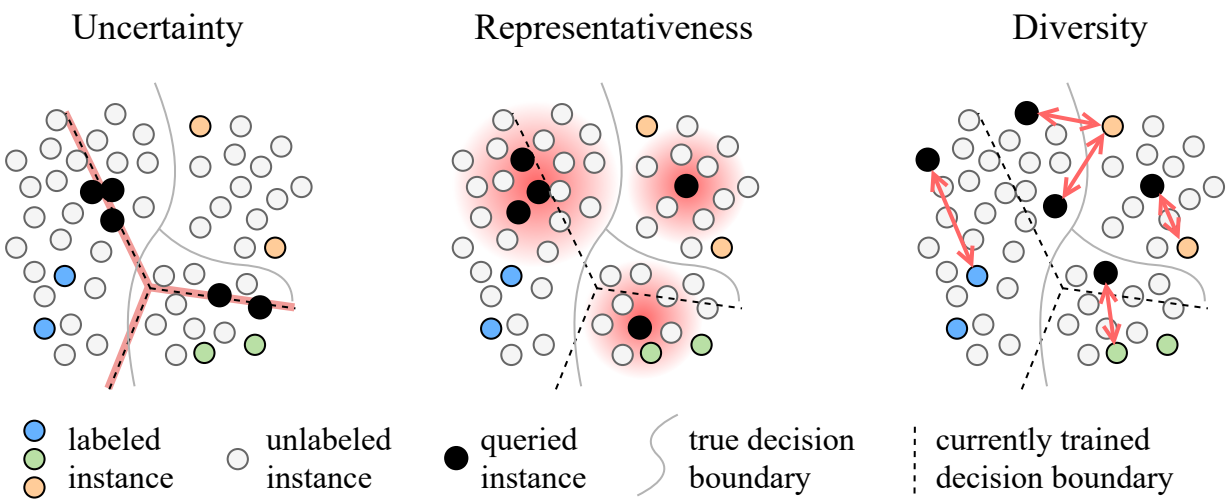


Figure 2.5: *Illustration of different query types. Uncertainty sampling (left) favors instances close to the decision boundary. (b) Representativeness sampling (middle) favors instances in dense/central regions. (c) Diversity sampling (right) favors instances that are different from each other or other labeled instances.*

Bayesian methods use probabilistic modeling through Bayesian inference to model uncertainty instead of using multiple classifiers. A prominent Bayesian method is BALD [32], which uses dropout at inference to obtain a distribution over the model weights (called Monte Carlo dropout [31]) and selects samples that are expected to maximize the information gained from the model parameters. BatchBALD [55] and PowerBALD [56] are diversity-aware extensions of BALD to adapt it to batch mode selection. However, since Monte Carlo dropout requires multiple forward passes of the unlabeled pool and convergence of dense dropout layers, it does not scale well to large learning problems and is drastically slower than simple uncertainty metrics [116].

Representativeness-based/Density-based Sampling

Representativeness- or density-based methods target instances from highly populated areas that are typical and representative of the underlying data distribution (see also Figure 2.5) [108]. The queried objects should reflect the main characteristics and patterns within the data to better represent other unlabeled instances. By learning from these representative examples, the model often generalizes better to unseen instances from the same data distribution. In contrast to uncertainty-based methods, which might select outliers or rare edge cases, the primary goal is to choose instances in central or dense regions.

Choosing typical instances instead of uncertain ones can be particularly beneficial in the low-budget regime where neural networks are not sufficiently trained and produce

unreliable uncertainty scores [44, 118]. A crucial aspect of measuring representativeness is having access to expressive features that accurately capture the underlying structure of the data. The raw input features might not always be suitable for this purpose [22]. The learned internal features are also not very expressive when the network is poorly trained in low-budget situations. Some methods for image classification propose leveraging self-supervised representation learning to generate more meaningful features, which can then be utilized to select instances from dense regions of the feature space [22, 44, 115]. In the context of graph active learning, where input data is not i.i.d., it is common to exploit the inherent graph topology to explore the relationship between instances [30]. For instance, the degree of a node or PageRank centrality can be used to focus on nodes with higher connectivity [19, 34]. Representativeness-based methods are often paired with diversity-enhancing mechanisms to select a single instance from a certain dense or central region as representative of many instances in their neighborhood.

Diversity-based Sampling

Diversity-based methods encourage annotating instances that are different from each other to cover a broad and varied subset of the data instead of concentrating on small regions [61]. A common way to ensure diversity is to consider the interaction between points, as shown in the example in Figure 2.5. Especially in batch mode selection, diversity is important to minimize the likelihood of choosing similar instances [6]. One of the most important baselines here is the Coreset approach [95], which continuously selects the instance furthest away from already labeled instances in the latent feature space until the query budget is exhausted as a greedy solution to constructing a coreset (also known as k-center-greedy). The aim is that the labeled set yields maximum coverage to form a good surrogate of the data distribution. The work inspired several other diversity-aware approaches [1, 53, 115]. Another approach is training an adversarial network [101, 52] to discriminate between labeled and unlabeled data and query instances more similar to unlabeled data to ensure diversity. However, these methods are computationally very expensive due to the additional training costs [10]. To diversify the query batch within each AL iteration, many methods perform k-means clustering on learned representations. Then samples are drawn from each cluster to ensure decent diversity between the candidates [80, 82, 123, 65, 19, 34, 22]. Since k-means does not directly return a real instance, k-medoid can be used as an alternative [65, 111]. Another approach is to use the k-means++ initialization [4], which iteratively accounts for the interaction of selected query candidates to ensure diversity. Such diversity-based methods that explicitly consider the interaction of selected query objects are popular but often have high computational complexity [56]. To address the high computational costs, stochastic approaches, such as [8, 56], have been introduced.

Hybrid Approaches

Hybrid methods combine multiple of the above-described query types to form the query batch [61]. Each of the introduced query types has certain weaknesses when considered in isolation. For instance, uncertainty sampling might result in a highly biased set differing from the actual data distribution, or in the low-label regime, uncertainty scores may be unreliable [44]. Representativeness sampling can be advantageous in ambiguous datasets or low-label regimes. However, spending labels only on typical instances may waste annotation efforts on easy-to-learn and less informative examples [118]. Diversity-based selection might also sample insufficient informative instances, select many irrelevant outliers, or, in general, not consider task-specific aspects such as high imbalance or that some classes are generally harder to learn and require more labeling information. However, since it is hard to know which selection scheme works best for a new dataset, many state-of-the-art AL strategies unify several query criteria for a more robust selection [118].

In the following, we will explain the most important approaches, which are mainly evaluated on image classification, one of the most active areas of deep AL. Combining uncertainty with clustering-based diversity is very popular [82, 123, 80, 24, 105, 24]. For instance, CLUE [82] is an established method that uses entropy uncertainty as sample weights for k-means clustering. Then it selects instances closest to the centroids as queries for labeling. The sample weights ensure that the clusters are moved toward uncertain regions, resulting in a diverse and uncertain query set. Another popular approach is Alfamix [80], which considers instances as candidates for selection if their predictions for interpolated features are inconsistent. Similar to CLUE, Alfamix clusters the candidate set and selects instances closest to the centroids as final query objects for labeling. Other techniques first filter the most uncertain objects based on a hyperparameter before selecting instances from each cluster as query objects [105, 123]. However, pre-filtering by a hyperparameter is problematic as performance is sensitive to its choice. Moreover, these methods perform k-means clustering on the latent features, which is computationally expensive for large unlabeled pools and high-dimensional latent features. One of the most established approaches over the past years is BADGE [6], which uses k-means++ initialization on the gradients of the last layer to query instances. The idea is that the gradients of uncertain instances have larger magnitudes and are favored by the k-means++ selection, thus selecting uncertain and diverse instances. However, as the size of the gradient-based embeddings is proportional to the number of classes times the dimensionality of the latent features, BADGE's runtimes can be really high [10].

Determining which subset of data will provide the greatest benefit when labeled to enhance the model's performance is challenging, particularly because datasets can vary significantly, and their unique characteristics might be unknown in advance in real-world

scenarios [61, 108, 68]. Thus, many state-of-the-art active learning methods rely on the combination of different query types to increase the robustness for varying settings and transferability to new datasets [61, 118].

2.4 Combining Active and Semi-Supervised Learning

Active learning and semi-supervised learning are naturally related, since they both address the challenge of high annotation efforts by learning with limited labeled data. Both concepts try to extract knowledge from unlabeled data to make informed decisions on how to best improve the model in further steps. However, they tackle the problem from opposite directions [96]. On one side, semi-supervised learning directly utilizes unlabeled data as an additional unsupervised training source to improve the model beyond plain supervision. On the other side, active learning successively increases the labeled pool by integrating a human in the loop and allocating the labeling efforts to the most valuable instances. An advantage of SSL is that it assumes a fixed labeled pool and extends training capabilities beyond labeled information without requesting new labels. On the other hand, AL offers greater flexibility to improve the quality of the labeled dataset tailored to the task-specific learning challenges faced by a model. The concepts are compatible, and it is worth exploring their combination into a unified framework to exploit the strengths of both.

To integrate SSL into the AL loop, we extend the training phase by leveraging the labeled and unlabeled data. In the training phase, we can utilize any of the previously explained semi-supervised methods to train the classifier. Then the normal AL procedure continues, where the query strategy returns the most relevant instances from the unlabeled pool and sends them to the annotator. After they are annotated, the model is re-trained in a semi-supervised manner as in the beginning. The overall framework of combining AL and SSL is visualized in Figure 2.6. Within the framework, the model is jointly optimized by utilizing labeled and unlabeled data and increasing the labeled pool with instances from which it most benefits.

Combining these two paradigms has been subject to research in several works [33, 46, 64, 91, 95, 125]. CEAL [110] combines threshold-based pseudo-labeling and entropy sampling and shows its effectiveness over other active learning methods for image classification. The authors of [103] show that combining Mixmatch [14] and margin sampling yields better results than random labeling. [35] proposes a consistency-based approach not only for training, as is often done in SSL, but also for measuring uncertainty for active selection. They demonstrate the effectiveness of their proposed combination of SSL and AL on various image classification benchmarks.

However, given the high computational complexity of many SSL approaches [76], us-

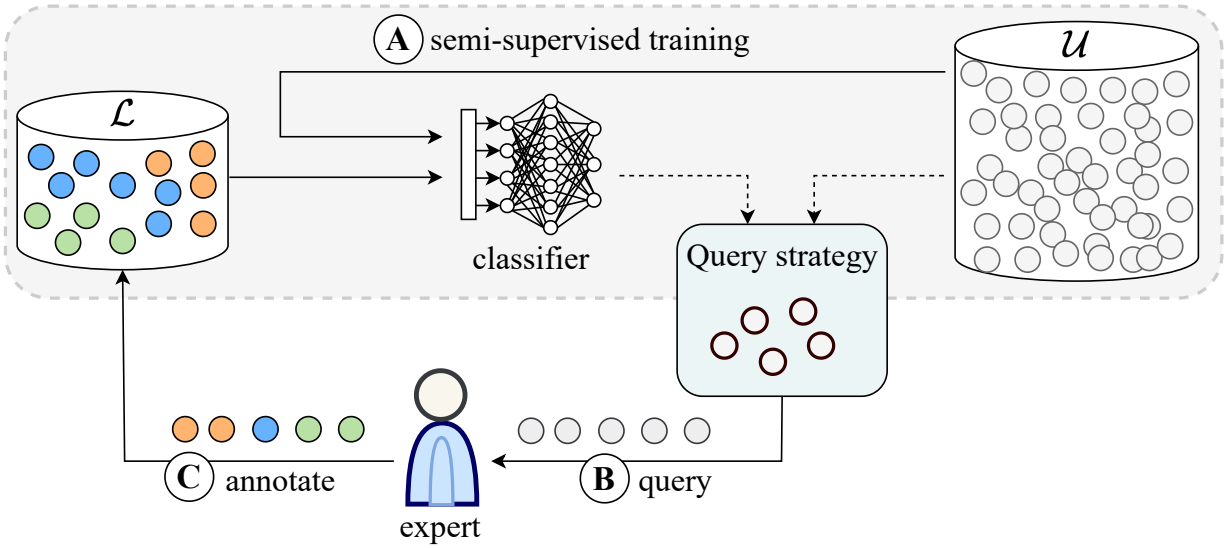


Figure 2.6: *Overview of combining active and semi-supervised learning. Ⓐ Train the classifier using the initially labeled pool $\mathcal{L}$ and the unlabeled pool $\mathcal{U}$ in a semi-supervised way. Ⓑ Select instances from the unlabeled pool via a query strategy and send them to the human annotator. Ⓒ Receive labels for queried instances and move them to the labeled pool $\mathcal{L}$. Repeat until the stopping criterion is reached.*

ing SSL in every AL round may be impractical or prohibitive in scenarios with time limitations [64]. Moreover, without careful consideration, the combination may not always yield better results than standard AL, which does not incorporate unlabeled data [27]. This can be attributed to the fact that SSL methods may suffer from confirmation bias, especially when supervision is really scarce or the learning task is very difficult [106, 3].

The authors of [100] report that AL generally benefits from utilizing unlabeled data at model training, but the ranking of the best query methods is not consistent. In [73], the marginal benefit of AL compared to the relatively large performance gain of SSL is criticized. The authors argue that data augmentations are one of the key drivers of the success of SSL. However, designing label-preserving data augmentations often requires domain expertise [89]. Moreover, SSL techniques are often evaluated using a reasonably large validation set to tune their hyperparameters, which is contrary to the limited label scenario [76]. Contrary to the claims of [73], the results reported in [68] show that AL is particularly useful in class-imbalanced settings, where popular SSL methods such as Fixmatch [102] struggle.

To summarize, existing research in this field is not always consistent, making it difficult to determine the practical effectiveness of combining SSL and AL in real-world scenarios. While the potential benefits have been demonstrated, it remains challenging for practitioners to assess under which conditions the combination is truly advantageous.

Chapter 3

Contributions

In this chapter, we provide an overview of the contributions included in this thesis and discuss each of them in more detail in the following sections. The published works, including supplements and a description of the author's contributions, can be found in Appendices A to E. This thesis involves five publications [37, 41, 39, 38, 40]. The code for all our contributions is publicly available. All publications are situated in the field of deep active learning. Overall, we propose three new AL algorithms for diverse multi-class classification tasks, which are explained in Sections 3.1 to 3.3. An overview of the key features of each of the proposed algorithms is given in Table 3.1. Moreover, the contribution discussed in Section 3.4 investigates the combination of AL and SSL for image classification on three realistic data challenges, providing valuable insights to researchers and practitioners.

Table 3.1: *Overview of proposed algorithms and distinguishing characteristics ("Unc"=Uncertainty, "Rep"=Representativeness, "Div"=Diversity).*

Algorithm	Data Type			Efficiency		Query Type			SSL
	Image	Tabular	Graph	Label	Time	Unc	Rep	Div	
FALCUN [41]	✓	✓		✓	✓	✓		✓	
DiffusAL [39]			✓	✓	✓	✓	✓	✓	✓
VERIPS [38]	✓			✓		✓			✓

Section 3.1 comprises two publications. In [37], we show that diversity sampling in the prediction probability space benefits time efficiency. We utilize these findings and propose a novel AL method named FALCUN [41] which leverages the probability space to query uncertain and diverse instances for labeling. Our experiments on image and tabular benchmarks demonstrate FALCUN's superiority in terms of label and time efficiency. Section 3.2 summarizes the key contributions of DiffusAL [39], an active learning method for node classification on graph data. DiffusAL combines diversity, representa-

tiveness, and uncertainty by exploiting information about the graph topology. Besides strong results regarding label efficiency, DiffusAL is also time efficient due to several precomputations that accelerate training and the calculation of metrics for the acquisition. The training involves utilizing the labeled and unlabeled nodes and is, therefore, considered semi-supervised. In the remaining sections, we look closer at the combination of AL with SSL in the context of image classification. First, we discuss our proposed algorithm named VERIPS [38] in Section 3.3. VERIPS combines uncertainty sampling with semi-supervised pseudo-labeling. 3.4 summarizes our contribution on how AL can help mitigate issues related to confirmation bias in SSL when facing three types of challenging datasets [40]. In the following, we will explain each contribution in more detail.

3.1 Sampling in the Probability Space for Faster Acquisitions

To reduce computational overhead, deep active learning is usually conducted in a batch mode, where multiple samples are queried and forwarded to the annotator at once, as explained in Section 2.3.2. Popular diversity-sampling techniques account for the interaction of instances within each AL round to avoid selecting instances with high information overlap. These methods commonly utilize latent feature vectors to measure similarities and ensure that the instances chosen for labeling are distant in the latent space. However, using these latent features to measure instance similarity can be computationally intense due to the high dimensionality of the hidden layers of neural networks. This becomes even worse when the available unlabeled data pool is very large, which is often the case since the collection of unlabeled data can usually be automated and requires less manual effort [23].

Accelerating Diversity Sampling

In this work, we address the problem of slow query times of diversity-based techniques by proposing a simple yet effective modification: we operate on the output probability vector instead of the latent features to measure instance similarity [37]. Since the output probabilities with a size equal to the number of classes usually have a much smaller dimensionality than the latent embeddings, the acquisition time is drastically reduced. We empirically demonstrate this by conducting experiments on MNIST [59] and using an MLP with a hidden dimensionality of 256 in its final layer, a commonly used architecture for this task [6]. We utilize three particularly popular diversity-enhancing techniques: (1) k-means-center (e.g., [82, 80, 123]), which performs k-means clustering with k equal to the query size Q and subsequently selects the closest point to each cluster center. (2)

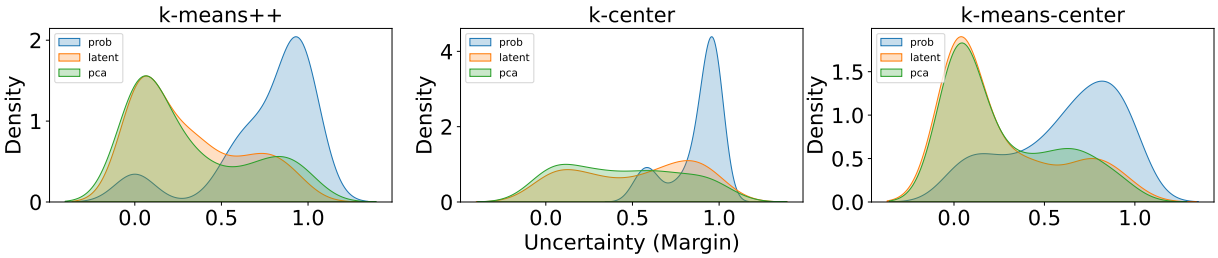


Figure 3.1: *Distribution of margin uncertainty of queried instances returned by several diversity-based methods when taking the prediction probability (blue), the latent features (orange), or the latent features after reducing dimensionality with PCA (green), demonstrated on MNIST. Without a dedicated uncertainty-enhancing mechanism, diversity sampling in the probability space automatically returns instances of higher uncertainty. This behavior influenced the design of our novel AL method called FALCUN.*

k-center (e.g., [95, 1, 53]), which iteratively selects instances that are furthest away from currently labeled data. (3) k-means++ (e.g., [6]), which iteratively selects instances with a probability proportional to their distance from previously selected samples. We apply each of the methods using the latent features, a reduced version of the features using Principal Component Analysis (PCA), and the output probabilities. As expected, the query time for each method is greatly reduced by utilizing the output probabilities. For instance, k-means-center is accelerated by a factor of 8 with our approach. A common assumption is that latent features are more expressive and contain important information that is lost when replacing them with the prediction probabilities, resulting in decreased label efficiency. However, our empirical findings show contrary effects and that this modification positively influences label efficiency. In fact, we show that all methods yield roughly 4% better accuracy on average when performing them on the prediction probabilities. We assume that diversity sampling in the probability space automatically also returns informative objects. We investigate this hypothesis in more detail in a follow-up work, which is explained in the next paragraph.

FALCUN

Figure 3.1 shows the distribution of uncertainty scores among 100 queried objects when sampled with different diversity sampling techniques operating on the output probabilities (“prob”, blue), the latent embeddings of the penultimate layer (“latent”, orange), or on a PCA-reduced version thereof (“pca”, green) as a baseline to accelerate the diversity sampling. The experiment was performed on MNIST with an experimental setting similar to that explained above. As the illustration shows, applying diversity sampling techniques in the probability space automatically selects instances with higher uncertainty. This is because instances far apart in the latent space are close in the probability space if the model is highly confident that they belong to the same class. Hence, by

focusing on diversity in the probability space, we automatically target diverse and uncertain areas, avoiding oversampling confident and irrelevant instances since these are naturally grouped together. In contrast, measuring diversity in the latent space would require an extra step to filter out these uninteresting samples, adding additional complexity to the process. We utilize these findings and propose a novel hybrid AL method named FALCUN [41].

In one AL round, FALCUN iteratively selects instances based on an additive relevance score $r(\mathbf{x}) = u(\mathbf{x}) + d(\mathbf{x})$, where $u(\mathbf{x})$ and $d(\mathbf{x})$ denote an uncertainty score and an adaptive diversity score, respectively. We use margin for the uncertainty score and reuse it to initialize the diversity score, i.e., $u(\mathbf{x}) := 1 - (\mathbf{p}(\hat{y_1}) - \mathbf{p}(\hat{y_2}))$ and $d_{init}(\mathbf{x}) := u(\mathbf{x})$, where $\mathbf{p}(\hat{y_1})$ and $\mathbf{p}(\hat{y_2})$ are the probabilities of the most and second most likely class. After every chosen query object $\mathbf{x}_q$, we dynamically update the current distance score of an instance $\mathbf{x}$ with $d(\mathbf{x}) \leftarrow \min(d(\mathbf{x}), dist(\mathbf{p}_\mathbf{x}, \mathbf{p}_{\mathbf{x}_q}))$, where the distance $dist(\cdot, \cdot)$ is the L1 norm. We apply min-max normalization to scale the values into the range $[0, 1]$, aligning them with $u(\mathbf{x})$. Initializing the diversity score with margin uncertainty ensures that confident samples are assigned low priority. As diversity scores can only decrease, these instances maintain a low selection priority throughout the query round. Moreover, we show that margin uncertainty is better suited for initializing diversity as it naturally emphasizes more diverse regions than entropy or least confidence uncertainty. Since a completely deterministic selection could potentially be less robust across varying datasets, we choose the next query instance $\mathbf{x}_q$ based on the probability $x_q \sim \frac{r(x)^\gamma}{\sum_{x \in \mathcal{U}} r(x)^\gamma}$, where y is a parameter controlling the degree of randomness ($\gamma = 0$ would resemble random sampling). In our experiments, we show that FALCUN is largely insensitive to the choice of γ for values larger than or equal to 5. As the query batch fills, the algorithm gradually shifts from the most uncertain instances toward exploring more diverse concepts.

FALCUN has some advantages over other hybrid methods that also account for the interaction of instances in their diversity sampling. For instance, previous methods often treat diversity and uncertainty estimation separately. To unify both metrics, they require certain predefined parameters, such as certain thresholds or other parameters balancing the trade-off between uncertainty and diversity [80, 82, 105]. Moreover, as explained previously, existing methods measuring distances in the latent space are often computationally expensive due to the high dimensionality of the latent embeddings [56]. Concretely, the time complexity of the acquisition is often dominated by $\mathcal{O}(N_u \cdot D)$, where N_u is the size of the unlabeled pool and D is the dimensionality of the penultimate layer. For instance, the time complexities for the popular methods BADGE [6] and CLUE [82] are $\mathcal{O}(Q \cdot N_u \cdot C \cdot D)$ and $\mathcal{O}(Q \cdot N_u \cdot i \cdot D)$, respectively, where Q is the query batch size, i is the number of cluster rounds, and C is the number of classes. In contrast, FALCUN's time complexity is $\mathcal{O}(Q \cdot N_u \cdot C)$, where usually $C \ll D$. Our experiments on several benchmarks for tabular classification and image classification demonstrate that FALCUN

outperforms compared methods in terms of label and time efficiency. Furthermore, we show in a separate experiment that FALCUN achieves stable results even on highly redundant datasets despite the omission of comparisons in the feature space.

3.2 DiffusAL

When considering AL for graph data, a major difference compared to image or tabular data, where data is usually i.i.d., is the availability of relations between the instances [34]. These edges, i.e., the connection between nodes in the graph, provide meaningful information about the graph structure and the relevance of each node. As such, they not only offer possibilities for developing special graph models but also provide valuable information for developing dedicated graph AL strategies [19, 34, 111, 122]. However, it remains challenging to use this information to design an approach that consistently outperforms random selection. Some existing approaches focus on limited selection aspects [65, 86, 111, 122] and outperform random sampling only on certain graphs. Other methods combine query types but are sensitive to hyperparameters [19, 34]. In addition, many methods use a graph convolutional network (GCN)[54] for training and acquisition. However, GCNs couple the learning of latent node features with neighborhood aggregation, which increases the time complexity of the active learning procedure. To address these limitations, we propose a novel graph AL method called DiffusAL [39] that leverages diffusion-based graph learning [36] for training and active selection. Graph diffusion overcomes certain disadvantages of conventional graph neural networks, such as the restriction to k-hop neighborhoods [18] or oversmoothing problems [63, 112]. Moreover, the neighborhood aggregation step can be pre-computed and is decoupled from learning latent node features, which makes them more efficient [18].

Model architecture Following previous works [18, 36], we pre-compute diffused features by calculating the personalized page rank (PPR) matrix over multiple scales denoted by P and multiplying it by the original node features X. These diffused features are then used as input to our classification model. To stabilize predictions, especially in early training phases or when labeling information is sparse, we propose to use a query-by-committee (QBC) comprising an ensemble of MLPs. Since committee approaches require the training of multiple classifiers, they are usually less desirable in terms of training efficiency. However, since the expensive diffusion step only needs to be performed once in advance and not repeatedly within the AL framework and training, our ensemble is trained faster than GCNs, which are commonly used for other graph AL methods [19, 111], as shown in the experiments. Each member of the ensemble has the

same architecture, and they only differ in their initializations. For the final prediction, we sum over the individual predictions of all members and apply softmax to get the predicted probability distribution.

Active selection We propose a combination of uncertainty, diversity, and representativeness to query instances for annotation since a single query type might not be capable of yielding robust results for many datasets. As the uncertainty score $s_{\text{unc}}(\mathbf{x})$, we utilize Shannon entropy [98] over the prediction probability of the model committee and apply L1-normalization over all unlabeled nodes. As a result, the range is bound between $[0, 1]$ and aligns with the other scores. For diversity, we perform k-means clustering on the pre-computed diffused features with k equal to the query batch size. Within each iteration, the diversity score keeps track of how many labeled instances are in each cluster and assigns higher weights to instances of currently underrepresented clusters. More precisely, the diversity score is defined as $s_{\text{div}}(\mathbf{x}) = 1 - \frac{|c_{train}|}{|V_{train}|}$, where $|c_{train}|$ and $|V_{train}|$ denote the number of labeled nodes in the cluster c that instance $\mathbf{x}$ was assigned to, $|V_{train}|$ represents the number of labeled training nodes in the current AL round. After every selection round, the number of $|c_{train}|$ and $|V_{train}|$ change. However, since we utilize the pre-computed diffused features, we do not recalculate the clustering in every iteration in contrast to other methods [65]. For representativeness, we compute a node importance score for each instance $\mathbf{x}$ by reusing the PPR matrix P. An entry P_{ij} in the PPR matrix can be interpreted as the importance of node j for node i, and the column-wise sum gives a proxy for the overall influence of the node in the graph. Concretely, the importance score of the j-th instance $\mathbf{x}$ is given by: $s_{\text{imp}}(\mathbf{x}) = \sum_{i \in V} P_{ij}$. Again, our approach does not require any recalculations during the AL rounds. The final score $s(\mathbf{x})$ of an instance $\mathbf{x}$ is defined as the product of the individual scores, i.e., $s(\mathbf{x}) = s_{\text{unc}}(\mathbf{x}) \cdot s_{\text{div}}(\mathbf{x}) \cdot s_{\text{imp}}(\mathbf{x})$. For the final acquisition, we calculate $s(\mathbf{x})$ for all instances in the unlabeled pool and select the nodes with the largest values for annotation.

Utilizing diffusion-based heuristics for training and active acquisition of nodes yields strong results, as experiments on several benchmarks demonstrate. Statistically, DiffusAL is the only method that outperforms random sampling in all experiments. We conduct ablation studies to understand the impact of the individual components and reveal that the performance of DiffusAL does not depend on just one single component but is the result of combining several components. Moreover, despite the combination of many selection criteria and a QBC approach, DiffusAL has fast acquisition and training times. In fact, it is often among the fastest approaches.

3.3 VERIPS

In AL, uncertain instances are considered to carry novel and valuable information and are thus favored for labeling. In contrast, pseudo-labeling selects the most confident samples to artificially increase the labeled pool. Both techniques can be combined into a unified framework to fully exploit the perceived knowledge of a model. For example, CEAL [110] is an established method integrating pseudo-labeling into the AL framework. It chooses the most uncertain instances estimated by entropy for labeling and the most certain ones for pseudo-labeling. CEAL uses two hyperparameters: a pseudo-label threshold and a decay rate that updates the threshold over the AL course to balance the selection of pseudo-labels. While the potential of combining these two paradigms is large and it is very appealing due to the simplicity and natural fit, naive approaches, like CEAL, fail when the initial model performs poorly [27]. Especially early in the AL cycle, where labeled information is particularly sparse, there is a high risk of producing many wrong pseudo-labels, resulting in confirmation bias.

To overcome this problem and stabilize the combination of pseudo-labeling and AL, we propose VERIPS (VERIfied Pseudo-label Selection for active learning). In particular, our method consists of a two-step approach where pseudo-labels are first selected based on a threshold following common pseudo-labeling standards and then only retained if the model's prediction matches the prediction of a second, similar model trained without pseudo-labeling. The second network provides an additional view to refine pseudo-labels. Concretely, given two networks, where the task network f_T is trained on $\mathcal{L}$ and $\mathcal{U}$ by pseudo-labeling and the verifier network f_V is only trained supervised on $\mathcal{L}$, we simply discard a pseudo-labeled instance $\hat{y} = f_T(\mathbf{x}, \theta_T)$ if it does not match the prediction of the verifier network $\tilde{y} = f_V(\mathbf{x}, \theta_V)$, i.e., $\hat{y} \neq \tilde{y}$. Our experiments show that CEAL often performs worse than baselines that do not use pseudo-labeling at all. In contrast, VERIPS significantly outperforms CEAL in the first AL cycles (by up to 27%). Even in the last round, VERIPS demonstrates up to 10% better accuracy. The performance of VERIPS is consistent among the tested data sets.

Furthermore, we show that VERIPS consistently achieves a higher proportion of correct pseudo-labels across all iterations. Especially in early rounds, VERIPS effectively discards many wrong pseudo-labels, yielding up to 20% higher pseudo-label correctness. Since VERIPS employs a filter mechanism to discard pseudo-labels, one could assume that CEAL involves significantly more pseudo-labels during training. However, since the training is stabilized after a few rounds, the proportion of instances falling over the threshold increases rapidly. As a result, after exhibiting the whole labeling budget, VERIPS not only has a better correctness ratio of pseudo-labels but also incorporates a similarly high amount of pseudo-labels as CEAL. Moreover, VERIPS only requires a single hyperparameter, i.e., the pseudo-label threshold, and we show that VERIPS is less

sensitive to its choice.

3.4 How to Overcome Confirmation Bias in Semi-Supervised Image Classification by Active Learning

Although AL has succeeded in various tasks and can significantly outperform random sampling, concerns have been raised about its trustworthiness and applicability in real-world situations for image classification [73, 66, 21]. When implementing AL on a new, largely unlabeled dataset, it is particularly difficult to assess which query method works best since one would have to label and compare data points along each AL trajectory. This is contrary to the ambition of reducing labeling costs. This challenge is known as the validation paradox [68]. Moreover, the authors in [73] state that SSL yields a greater relative performance improvement than deciding on the best query method. Furthermore, some works show the effectiveness and potential of combining AL and SSL [14, 110]. In our contribution [40], we aim to gain more insights into the applicability of AL and its combination with SSL. We find that most of the existing research has been conducted on benchmark datasets that do not resemble the challenges that are often present in real-world datasets. To provide insight into the topic, we analyze the combination of SSL and AL on realistic dataset challenges.

Therefore, we first identify three common real-world data challenges: between-class imbalance (BCI), between-class similarity (BCS), and within-class imbalance (WCI). BCI refers to a high imbalance concerning the number of examples over the different classes. It is the most well-known and studied problem among the three challenges, but in terms of combining SSL and AL, it is understudied. BCS refers to datasets that exhibit a high proportion of instances that are hard to distinguish between classes. The reason might be that two or more classes share similar or overlapping concepts or that the inherent noise in the data indicates a high aleatoric uncertainty [26]. WCI refers to the imbalance that can occur within classes in real-world datasets. Uncleaned real-world data might have huge amounts of very similar instances, but only a few instances carry diverse and rare aspects that could help the model better distinguish between classes. An illustrative example of these data challenges is shown in Figure 3.2.

To shed some light on how SSL performs under the aforementioned conditions, we then conduct experiments to study these challenges. We construct three versions of the MNIST task exhibiting the introduced data challenges. Then, we randomly label a few data points with varying budgets and perform pseudo-labeling [60], Fixmatch [102], and Flexmatch [119]. Our results show that these powerful SSL methods suffer from

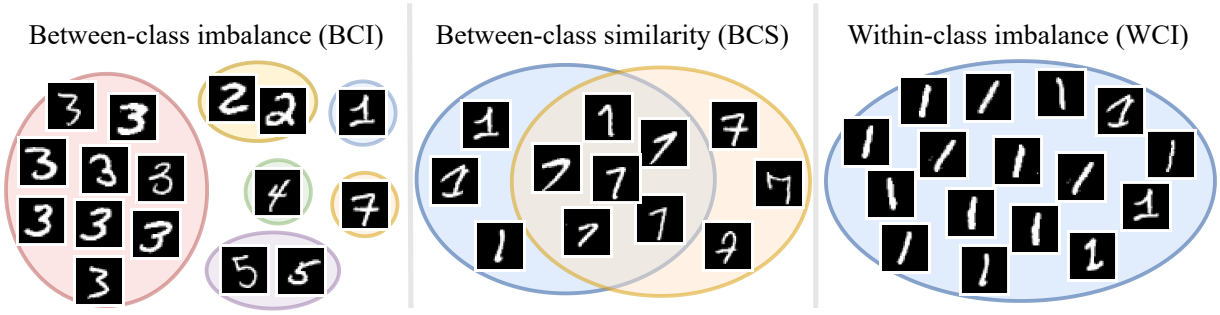


Figure 3.2: *Illustrative example of the three data challenges between-class imbalance (BCI), between-class similarity (BCS), and within-class imbalance (WCI) visualized by sample images taken from MNIST. Different colors indicate different classes. For BCI, the class distribution is imbalanced. In the case of BCS, the class of instances in the overlapping region is ambiguous. For WCI, there are many similar versions of the digit "1" and only a few different ones.*

confirmation bias in all studied challenges. In the case of BCI, the imbalance is confirmed repeatedly, increasing the class distribution mismatch, and the SSL methods partially even perform worse than the fully supervised baseline. For BCS, wrong pseudo-labels are produced frequently, resulting in worse performance than supervised learning. For WCI, the performance stagnates, and only already known concepts are confirmed.

With the insights of the failing SSL methods, we lastly investigate if AL could help to overcome the observed confirmation bias problems. We repeat the experiments on BCI, WCI, and BCS with the SSL methods, but instead of sampling data randomly, we use several AL methods to choose the data. We use four baseline AL methods in our experiments as representatives of different query categories: uncertainty-based, coverage-based (or diversity-based), representativeness-based, and a baseline that enhances class balance. Uncertainty and coverage-based sampling can improve the performance on BCI by up to 20%. BCS can be improved by up to 10% by annotating representative and typical instances rather than confusing, uncertain ones. On WCI, by sampling diverse instances, the SSL method almost yields the same performance as on the original MNIST (increasing accuracy by almost 10%).

To summarize, existing SSL techniques often suffer from confirmation bias, especially in the presence of the aforementioned real-world challenges. To overcome this problem, SSL research mainly focused on developing techniques where wrong model outputs have less impact or are identified as more reliable. In our work, we take a different perspective and show that guiding the quality of supervised information is a promising option to improve results. Our results indicate that confirmation bias can be mitigated by selecting more valuable samples through AL.

Chapter 4

Conclusion

To conclude this thesis, we give a short summary of our contributions in Section 4.1 and discuss limitations as well as potential future research directions in Section 4.2.

4.1 Summary

In this thesis, we presented various advances in the field of deep active learning. Furthermore, we also contributed to successfully integrating active and semi-supervised learning.

We proposed three deep AL algorithms for image, tabular, and node classification tasks. FALCUN is an AL method for image and tabular classification. It yields competitive and robust results on different datasets by carefully balancing uncertainty and diversity in the probability space. For node classification, we proposed DiffusAL, a novel method that unifies diffusion-based heuristics for model training and for actively selecting uncertain, representative, and diverse instances for labeling. Our experiments demonstrate that DiffusAL is the only method consistently better than random sampling. VERIPS combines uncertainty sampling and pseudo-labeling for image classification. By only considering pseudo-labels that are verified by a separate network, VERIPS discards false pseudo-labels and mitigates confirmation bias. As a result, VERIPS yields higher model accuracy than existing approaches.

Besides label efficiency, some of our contributions focus on time efficiency. For instance, FALCUN performs diversity sampling in the usually lower-dimensional probability space instead of the latent space. As a result, FALCUN has faster acquisition times than existing diversity sampling methods that consider the interaction between query objects. DiffusAL uses various pre-calculations that are reused for training and acquisition, and thus do not burden the runtime during the AL rounds. Saving time in the training or acquisition phase of the AL loop helps to deliver results quicker, making approaches more

appealing, and potentially saving annotator costs.

The problem of confirmation bias is further addressed in our contribution in Section 3.4. Here, we show that confirmation bias hinders the effectiveness of popular SSL methods when trained on a randomly labeled pool and facing three particularly difficult data challenges. We further demonstrate that actively selecting the labeled dataset is a viable tool to mitigate confirmation bias in SSL for the evaluated challenging datasets. Our research reveals important insights into the potential of combining AL and SSL and the conditions under which their integration is beneficial.

4.2 Limitations and Future Work

Most existing works on graph active learning, including our algorithm DiffusAL [39], have concentrated on graphs with high homophily, where the assumption is that closely connected nodes belong to similar classes. However, many real-world graphs have the opposite property, called heterophily, where connected nodes are dissimilar in features or labels [81]. Heterophilous graphs may demand fundamentally different strategies for labeling instances. Suboptimal labeling could potentially hinder the model from capturing the heterogeneous patterns. DiffusAL combines multiple selection criteria that may make it more robust to such a setting than other AL methods. However, so far, DiffusAL has been evaluated on datasets with a high degree of homophily. Investigating the performance of heterophily settings presents an interesting avenue for future research.

Our last contribution [40] provides valuable insights into the potential and difficulties of using active semi-supervised learning when faced with challenges present in many datasets. In our research, we focused on evaluating baseline AL methods to understand which query types are generally beneficial for which challenge. Moreover, the evaluation is limited to variants of MNIST to specifically examine the effect of each challenge. In future work, we aim to do an extensive evaluation on real-world datasets that exhibit the introduced challenges. This will allow us to give concrete recommendations for practitioners regarding the choice of AL and SSL for certain data challenges.

Moreover, we aim to include our methods, FALCUN [41] and VERIPS [38], in this benchmark. Since both emphasize uncertainty in their selection, they might struggle with datasets exhibiting high between-class similarity, as many confusing instances may be chosen. A possible way to overcome this could be to extend them to methods that distinguish between aleatoric and epistemic uncertainty. Aleatoric uncertainty refers to the inherent noise or randomness in the data that can not be reduced by adding more data. In contrast, epistemic uncertainty refers to uncertainty that comes from the lack of knowledge of a model [26]. Moving the focus towards epistemic uncertainty might help identify uncertain instances from which the model can learn better. However, even if it

is known which algorithms perform well for which data characteristics, it might be challenging to make an appropriate choice in settings where the exact dataset characteristics are unknown in advance. Therefore, another interesting direction for future work is to focus on designing robust methods that perform well across all challenges rather than performing well only on datasets exhibiting one of the challenges.

References

[1] Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 137–153. Springer, 2020.

[2] A Agrawala. Learning with a probabilistic teacher. *IEEE Transactions on Information Theory*, 16(4):373–379, 1970.

[3] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2020.

[4] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[5] Nabiha Asghar, Pascal Poupart, Xin Jiang, and Hang Li. Deep active learning for dialogue generation. *arXiv preprint arXiv:1612.03929*, 2016.

[6] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=ryghZJBKPS`.

[7] Davide Bacciu, Federico Errica, Alessio Micheli, and Marco Podda. A gentle introduction to deep learning for graphs. *Neural Networks*, 129:203–221, 2020.

[8] Dara Bahri, Heinrich Jiang, Tal Schuster, and Afshin Rostamizadeh. Is margin all you need? an extensive empirical study of active learning on tabular data. *arXiv preprint arXiv:2210.03822*, 2022.

[9] Peter W Battaglia et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[10] Nathan Beck, Durga Sivasubramanian, Apurva Dani, Ganesh Ramakrishnan, and Rishabh Iyer. Effective evaluation of deep active learning on image classification tasks. *arXiv preprint arXiv:2106.15324*, 2021.

[11] 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.

[12] Max Berrendorf, Evgeniy Faerman, and Volker Tresp. Active learning for entity alignment. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I 43*, pages 48–62. Springer, 2021.

[13] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019.

[14] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

[15] Smriti Bhagat, Graham Cormode, and S Muthukrishnan. Node classification in social networks. *Social network data analytics*, pages 115–148, 2011.

[16] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE SPM*, 34(4):18–42, 2017.

[17] Samuel Budd, Emma C Robinson, and Bernhard Kainz. A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical image analysis*, 71:102062, 2021.

[18] Julian Busch, Jiaxing Pi, and Thomas Seidl. Pushnet: Efficient and adaptive neural message passing. In *ECAI 2020*, pages 1039–1046. IOS Press, 2020.

[19] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. Active learning for graph embedding. *arXiv preprint arXiv:1705.05085*, 2017.

[20] Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6912–6920, 2021.

[21] Yao-Chun Chan, Mingchen Li, and Samet Oymak. On the Marginal Benefit of Active Learning: Does Self-Supervision Eat Its Cake? *arXiv:2011.08121 [cs]*, November 2020.

[22] Akshay L Chandra, Sai Vikas Desai, Chaitanya Devaguptapu, and Vineeth N Balasubramanian. On initial pools for deep active learning. In *NeurIPS 2020 Workshop on Pre-registration in Machine Learning*, pages 14–32. PMLR, 2021.

[23] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[24] Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. Batch active learning at scale. *Advances in Neural Information Processing Systems*, 34:11933–11944, 2021.

[25] Arnav Das, Gantavya Bhatt, Megh Bhalerao, Vianne Gao, Rui Yang, and Jeff Bilmes. Accelerating batch active learning using continual learning techniques. *arXiv preprint arXiv:2305.06408*, 2023.

[26] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

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

[28] S Fralick. Learning to recognize patterns without a teacher. *IEEE Transactions on Information Theory*, 13(1):57–64, 1967.

[29] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28:133–168, 1997.

[30] Yifan Fu, Xingquan Zhu, and Bin Li. A survey on instance selection for active learning. *Knowledge and information systems*, 35:249–283, 2013.

[31] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[32] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. 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 1183–1192. PMLR, 2017. URL http://proceedings.mlr.press/v70/gal17a.html.

[33] Fei Gao, Zhenyu Yue, Jun Wang, Jinping Sun, Erfu Yang, and Huiyu Zhou. A novel active semisupervised convolutional neural network algorithm for sar image recognition. *Computational intelligence and neuroscience*, 2017(1):3105053, 2017.

[34] Li Gao, Hong Yang, Chuan Zhou, Jia Wu, Shirui Pan, and Yue Hu. Active discriminative network representation learning. In *IJCAI*, 2018.

[35] Mingfei Gao, Zizhao Zhang, Guo Yu, Sercan Ö Arık, Larry S Davis, and Tomas Pfister. Consistency-based semi-supervised active learning: Towards minimizing labeling cost. In *Computer vision–ECCV 2020: 16th European conference, glasgow, UK, August 23–28, 2020, proceedings, part x 16*, pages 510–526. Springer, 2020.

[36] Johannes Gasteiger, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. *Advances in neural information processing systems*, 32, 2019.

[37] Sandra Gilhuber, Max Berrendorf, Yunpu Ma, and Thomas Seidl. Accelerating diversity sampling for deep active learning by low-dimensional representations. In *IAL@ PKDD/ECML*, pages 43–48, 2022.

[38] Sandra Gilhuber, Philipp Jahn, Yunpu Ma, and Thomas Seidl. Verips: verified pseudo-label selection for deep active learning. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 951–956. IEEE, 2022.

[39] Sandra Gilhuber, Julian Busch, Daniel Rotthues, Christian MM Frey, and Thomas Seidl. Diffusal: Coupling active learning with graph diffusion for label-efficient node classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 75–91. Springer, 2023.

[40] Sandra Gilhuber, Rasmus Hvingelby, Mang Ling Ada Fok, and Thomas Seidl. How to overcome confirmation bias in semi-supervised image classification by active learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 330–347. Springer, 2023.

[41] Sandra Gilhuber, Anna Beer, Yunpu Ma, and Thomas Seidl. Falcun: A simple and efficient deep active learning strategy. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 421–439. Springer, 2024.

[42] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[43] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE international joint conference on neural networks, 2005.*, volume 2, pages 729–734. IEEE, 2005.

[44] Guy Hacohen, Avihu Dekel, and Daphna Weinshall. Active learning on a budget: Opposite strategies suit high and low budgets. *arXiv preprint arXiv:2202.02794*, 2022.

[45] James B Heaton, Nick G Polson, and Jan Hendrik Witte. Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1):3–12, 2017.

[46] Siyu Huang, Tianyang Wang, Haoyi Xiong, Jun Huan, and Dejing Dou. Semi-supervised active learning with temporal output discrepancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3447–3456, 2021.

[47] Aras M Ismael and Abdulkadir Şengür. Deep learning approaches for covid-19 detection based on chest x-ray images. *Expert Systems with Applications*, 164: 114054, 2021.

[48] Sen Jia, Shuguo Jiang, Zhijie Lin, Nanying Li, Meng Xu, and Shiqi Yu. A survey: Deep learning for hyperspectral image classification with few labeled samples. *Neurocomputing*, 448:179–204, 2021.

[49] Heinrich Jiang and Maya R Gupta. Bootstrapping for batch active sampling. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3086–3096, 2021.

[50] Andreas Kamilaris and Francesc X Prenafeta-Boldú. Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, 147:70–90, 2018.

[51] Michael Kampffmeyer, Arnt-Borre Salberg, and Robert Jenssen. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1–9, 2016.

[52] Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. Task-aware variational adversarial active learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 8166–8175. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00807. URL

https://openaccess.thecvf.com/content/CVPR2021/html/Kim_Task-Aware_Variational_Adversarial_Active_Learning_CVPR_2021_paper.html.

[53] Yeachan Kim and Bonggun Shin. In defense of core-set: A density-aware core-set selection for active learning. In Aidong Zhang and Huzefa Rangwala, editors, *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 804–812. ACM, 2022. doi: 10.1145/3534678.3539476. URL https://doi.org/10.1145/3534678.3539476.

[54] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2017.

[55] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32, 2019.

[56] Andreas Kirsch, Sebastian Farquhar, Parmida Atighehchian, Andrew Jesson, Frederic Branchaud-Charron, and Yarin Gal. Stochastic batch acquisition: A simple baseline for deep active learning. *arXiv preprint arXiv:2106.12059*, 2021.

[57] Sushant Kumar, Sumit Datta, Vishakha Singh, Sanjay Kumar Singh, and Ritesh Sharma. Opportunities and challenges in data-centric ai. *IEEE Access*, 2024.

[58] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

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

[60] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Atlanta, 2013.

[61] Dongyuan Li, Zhen Wang, Yankai Chen, Renhe Jiang, Weiping Ding, and Manabu Okumura. A survey on deep active learning: Recent advances and new frontiers. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[62] Jiajia Li, Dong Chen, Xinda Qi, Zhaojian Li, Yanbo Huang, Daniel Morris, and Xiaobo Tan. Label-efficient learning in agriculture: A comprehensive review. *Computers and Electronics in Agriculture*, 215:108412, 2023.

[63] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[64] Jaeseung Lim, Jongkeun Na, and Nojun Kwak. Active semi-supervised learning by exploring per-sample uncertainty and consistency. *arXiv preprint arXiv:2303.08978*, 2023.

[65] Juncheng Liu, Yiwei Wang, Bryan Hooi, Renchi Yang, and Xiaokui Xiao. Lscale: latent space clustering-based active learning for node classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 55–70. Springer, 2022.

[66] David Lowell, Zachary C Lipton, and Byron C Wallace. Practical obstacles to deploying active learning. *arXiv preprint arXiv:1807.04801*, 2018.

[67] Andre Luckow, Matthew Cook, Nathan Ashcraft, Edwin Weill, Emil Djerekarov, and Bennie Vorster. Deep learning in the automotive industry: Applications and tools. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 3759–3768. IEEE, 2016.

[68] Carsten Lüth, Till Bungert, Lukas Klein, and Paul Jaeger. Navigating the pitfalls of active learning evaluation: A systematic framework for meaningful performance assessment. *Advances in Neural Information Processing Systems*, 36, 2024.

[69] Kaushalya Madhawa and Tsuyoshi Murata. Active learning for node classification: An evaluation. *Entropy*, 22(10):1164, 2020.

[70] Karan Malhotra, Shubham Bansal, and Sriram Ganapathy. Active learning methods for low resource end-to-end speech recognition. In *Interspeech*, pages 2215–2219, 2019.

[71] Smit Mehta, Chirag Paunwala, and Bhaumik Vaidya. Cnn based traffic sign classification using adam optimizer. In *2019 international conference on intelligent computing and control systems (ICCS)*, pages 1293–1298. IEEE, 2019.

[72] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246, 2018.

[73] Sudhanshu Mittal, Maxim Tatarchenko, Özgün Çiçek, and Thomas Brox. Parting with illusions about deep active learning. *arXiv preprint arXiv:1912.05361*, 2019.

[74] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8): 1979–1993, 2018.

[75] Eduardo Mosqueira-Rey, Elena Hernández-Pereira, David Alonso-Ríos, José Bobes-Bascarán, and Ángel Fernández-Leal. Human-in-the-loop machine learning: a state of the art. *Artificial Intelligence Review*, 56(4):3005–3054, 2023.

[76] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. *Advances in neural information processing systems*, 31, 2018.

[77] Natalia Ostapuk, Jie Yang, and Philippe Cudré-Mauroux. Activelink: deep active learning for link prediction in knowledge graphs. In *The world wide web conference*, pages 1398–1408, 2019.

[78] Yassine Ouali, Céline Hudelot, and Myriam Tami. An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*, 2020.

[79] Sarah Parisot, Sofia Ira Ktena, Enzo Ferrante, Matthew Lee, Ricardo Guerrero, Ben Glocker, and Daniel Rueckert. Disease prediction using graph convolutional networks: application to autism spectrum disorder and alzheimer's disease. *Medical image analysis*, 48:117–130, 2018.

[80] Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Reza Haffari, Anton van den Hengel, and Javen Qinfeng Shi. Active learning by feature mixing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 12227–12236. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01192. URL https://doi.org/10.1109/CVPR52688.2022.01192.

[81] Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? *arXiv preprint arXiv:2302.11640*, 2023.

[82] Viraj Prabhu, Arjun Chandrasekaran, Kate Saenko, and Judy Hoffman. Active domain adaptation via clustering uncertainty-weighted embeddings. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 8485–8494. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00839. URL https://doi.org/10.1109/ICCV48922.2021.00839.

[83] Gil Press. Andrew ng launches a campaign for data-centric ai. https://www.forbes.com/sites/gilpress/2021/06/16/andrew-ng-launches-a-campaign-for-data-centric-ai/, 2021. Accessed: August 19, 2024.

[84] Kun Qian, Zixing Zhang, Alice Baird, and Björn Schuller. Active learning for bird sounds classification. *Acta Acustica united with Acustica*, 103(3):361–364, 2017.

[85] Lukas Rauch, Denis Huseljic, Moritz Wirth, Jens Decke, Bernhard Sick, and Christoph Scholz. Towards deep active learning in avian bioacoustics. *arXiv preprint arXiv:2406.18621*, 2024.

[86] Florence Regol, Soumyasundar Pal, Yingxue Zhang, and Mark Coates. Active learning on attributed graphs via graph cognizant logistic regression and preemptive query generation. In *ICML*, pages 8041–8050. PMLR, 2020.

[87] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021.

[88] Zeyu Ren, Shuihua Wang, and Yudong Zhang. Weakly supervised machine learning. *CAAI Transactions on Intelligence Technology*, 8(3):549–580, 2023.

[89] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*, 2021.

[90] Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *Machine Learning: ECML 2006: 17th European Conference on Machine Learning Berlin, Germany, September 18-22, 2006 Proceedings 17*, pages 413–424. Springer, 2006.

[91] Matthias Rottmann, Karsten Kahl, and Hanno Gottschalk. Deep bayesian active semi-supervised learning. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 158–164. IEEE, 2018.

[92] Hojjat Salehinejad, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee. Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078*, 2017.

[93] Christopher Schröder and Andreas Niekler. A survey of active learning for text classification using deep neural networks. *arXiv preprint arXiv:2008.07267*, 2020.

[94] Henry Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.

[95] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=H1aIuk-RW.

[96] Burr Settles. Active learning literature survey. 2009.

[97] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294, 1992.

[98] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[99] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

[100] Oriane Siméoni, Mateusz Budnik, Yannis Avrithis, and Guillaume Gravier. Rethinking deep active learning: Using unlabeled data at model training. In *2020 25th International conference on pattern recognition (ICPR)*, pages 1220–1227. IEEE, 2021.

[101] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 5971–5980. IEEE, 2019. doi: 10.1109/ICCV.2019.00607. URL https://doi.org/10.1109/ICCV.2019.00607.

[102] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

[103] Shuang Song, David Berthelot, and Afshin Rostamizadeh. Combining mixmatch and active learning for better accuracy with fewer labels. *arXiv preprint arXiv:1912.00594*, 2019.

[104] Michael Stonebraker and El Kindi Rezig. Machine learning and big data: What is important? *IEEE Data Eng. Bull.*, 42(4):3–7, 2019.

[105] Tao Sun, Cheng Lu, and Haibin Ling. Local context-aware active domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18634–18643, 2023.

[106] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[107] Mohammad A Thanoon, Mohd Asyraf Zulkifley, Muhammad Ammirrul Atiqi Mohd Zainuri, and Siti Raihanah Abdani. A review of deep learning techniques for lung cancer screening and diagnosis based on ct images. *Diagnostics*, 13(16): 2617, 2023.

[108] Alaa Tharwat and Wolfram Schenck. A survey on active learning: State-of-the-art, practical challenges and research directions. *Mathematics*, 11(4):820, 2023.

[109] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020.

[110] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2016.

[111] Yuexin Wu, Yichong Xu, Aarti Singh, Yiming Yang, and Artur Dubrawski. Active learning for graph neural networks via node feature propagation. *arXiv preprint arXiv:1910.07567*, 2019.

[112] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pages 5453–5462. PMLR, 2018.

[113] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

[114] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(9): 8934–8954, 2022.

[115] Ofer Yehuda, Avihu Dekel, Guy Hacohen, and Daphna Weinshall. Active learning through a covering lens. *Advances in Neural Information Processing Systems*, 35: 22354–22367, 2022.

[116] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 93–102, 2019.

[117] Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, and Xia Hu. Data-centric ai: Perspectives and challenges. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 945–948. SIAM, 2023.

[118] Xueying Zhan, Qingzhong Wang, Kuan-hao Huang, Haoyi Xiong, Dejing Dou, and Antoni B Chan. A comparative survey of deep active learning. *arXiv preprint arXiv:2203.13450*, 2022.

[119] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021.

[120] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[121] Pei Zhang, Xueying Xu, and Deyi Xiong. Active learning for neural machine translation. In *2018 International Conference on Asian Language Processing (IALP)*, pages 153–158. IEEE, 2018.

[122] Wentao Zhang, Zhi Yang, Yexin Wang, Yu Shen, Yang Li, Liang Wang, and Bin Cui. Grain: Improving data efficiency of graph neural networks via diversified influence maximization. *Proc. VLDB Endow.*, 14(11):2473–2482, 2021.

[123] Fedor Zhdanov. Diverse mini-batch active learning. *CoRR*, abs/1901.05954, 2019. URL `http://arxiv.org/abs/1901.05954`.

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

[125] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, volume 3, pages 58–65, 2003.

Appendix

A Accelerating Diversity Sampling for Deep Active Learning By Low-Dimensional Representations.

Authors

Sandra Gilhuber, Max Berrendorf, Yunpu Ma, and Thomas Seidl

Venue

Proceedings of the Workshop on Interactive Adaptive Learning co-located with European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (IAL@ PKDD/ECML), pages 43–48, 2022.

Link

`https://ceur-ws.org/Vol-3259/ialatecml_paper4.pdf`

Code

`https://github.com/sobermeier/low-dim-div-sampling`

Declaration of Authorships

Sandra Gilhuber proposed the research idea, developed and conceptualized it, and discussed it with Max Berrendorf and Yunpu Ma. Sandra Gilhuber did the implementation. Sandra Gilhuber designed and conducted the experiments and analyzed their results. Sandra Gilhuber and Max Berrendorf wrote and revised the manuscript.

Accelerating Diversity Sampling for Deep Active Learning By Low-Dimensional Representations

Sandra Gilhuber[1], Max Berrendorf[1], Yunpu Ma[1], and Thomas Seidl[1]

Ludwig-Maximilians-Universität München, Munich, Germany
{gilhuber,berrendorf,ma,seidl}@dbs.ifi.lmu.de

Abstract. Selecting diverse instances for annotation is one of the key factors of successful active learning strategies. To this end, existing methods often operate on high-dimensional latent representations. In this work, we propose to use the low-dimensional vector of predicted probabilities instead, which can be seamlessly integrated into existing methods. We empirically demonstrate that this considerably decreases the query time, i.e., time to select an instance for annotation, while at the same time improving results. Low query times are relevant for active learning researchers, which use a (fast) oracle for simulated annotation and thus are often constrained by query time. It is also practically relevant when dealing with complex annotation tasks for which only a small pool of skilled domain experts is available for annotation with a limited time budget. Our code is available at: https://github.com/sobermeier/low-dim-div-sampling.

Keywords: Active Learning · Diversity Sampling

1 Introduction

Deep neural networks are the dominant choice for solving complex tasks, such as image classification. Their great success depends in large part on the availability of a sufficient amount of labeled data. Especially in domains with scarce publicly available data, such as medical or industrial applications, annotations can become prohibitively expensive due to the need for skilled domain experts. The field of active learning thus aims at reducing the number of required annotations by intelligently selecting instances for labeling. Since modern networks require a significant amount of training time, the traditional setting where instances are selected one after the other [13,15,20] has become infeasible [17], and a batch-setting is commonly applied, where a fixed number of instances is selected for annotation.

State-of-the-art approaches [3,9,18,19,16] follow two different paradigms (or a mixture thereof): In *uncertainty*-based methods [4,5,10], those instances are selected for which the model is the least certain about the prediction. In contrast, *diversity* methods [3,6,7,16,18,19,22] focus on selecting a representative subset of instances and avoid re-labeling similar instances. In this work, we focus on the second class.

Diversity-based methods often rely on high-dimensional representations extracted from the model's last layers [3,6,7,8,11,16,18,22,21]. In the presence of a large pool of unlabeled data, processing these representations can become a bottleneck of the approaches resulting in increased query times. While these can often be neglected when the annotation is delegated to a large pool of on-demand crowd workers, in settings where domain experts are required, there is often only a small number of available annotators with tight schedules. In these settings, it is desirable to reduce the query time in addition to only requesting useful instances for annotation. Similarly, in active learning research, where a simulated oracle is used for annotation, the computational bottleneck is often the instance selection.

2 Diversity Sampling on Low-Dimensional Representations

In this work, we present a simple yet effective approach to accelerate diversity-based methods, which replaces the high-dimensional latent features $\mathbf{x} \in \mathbb{R}^d$ by the vector of predicted class probabilities $\mathbf{p} \in \mathbb{R}^c$, where usually $c \ll d$. The approach can be applied to most diversity-based methods without large modifications and effectively reduces the instance selection times.

We empirically evaluate our approach with multiple different diversity-based active learning heuristics. Note that we do not consider uncertainty in this work and focus only on underlying diversity concepts. However, the selected diversity methods are key concepts of various popular active learning strategies, such as [1,3,16,18,22].

1. `KMeansCenter` selects the points closest to the centroids of k-means clustering [14] with $k = q$ clusters for annotation, where q denotes the query size. As a recent example, CLUE [16] uses k-means clustering as diversity concept enriched by uncertainty weighting.
2. `KCenterGreedy` iteratively selects the sample with the largest minimum distance to any already labeled instance. It is also known as CoreSet [18] and one of the first solely diversity-based active learning methods.
3. `KMeans++` [2] iteratively samples instances with probability proportional to the minimum distance to already selected points in the current acquisition round. BADGE [3] is a prominent example using `KMeans++` on high-dimensional vectors.

For the iterative `KCenterGreedy` and `KMeans++` algorithms, we keep an array of minimum distance to already labeled samples, and update it whenever we add another sample for labeling. The time complexity of selecting one batch of queries is given in Table 1. Notice that for all heuristics, the time complexity linearly depends on the vector dimension.

We empirically evaluate the MNIST [12] dataset of handwritten digits with 10 classes and a simple 2-layer fully-connected network with embedding dimensionality 256 as in [3] for a proof-of-concept. The learning rate is set to 0.01,

Table 1. Time complexity of a single acquisition round of the different diversity-based heuristics. q denotes the query size, i.e., number of instances to select for labeling, n_l/n_u the number of labeled/unlabeled samples ($n_l \ll n_u$), d the vector dimensionality, and i the number of iterations until convergence.

Algorithm	Time Complexity
KMeansCenter	$\mathcal{O}(q \cdot n_u \cdot i \cdot d)$
KCenterGreedy	$\mathcal{O}(n_l \cdot n_u \cdot d + q \cdot n_u)$
KMeans++	$\mathcal{O}(q \cdot n_u \cdot d)$

and we train the network from scratch for 10 epochs in each iteration. The initial pool contains 100 randomly chosen samples, and we select additional 100 instances per active learning iteration until a budget of 2,500 samples is exhausted. We investigate three different input features $\mathbf{x}$ of the samples as input to the heuristics:

1. the full-dimensional latent features, i.e., $\mathbf{x} \in \mathbb{R}^d$,
2. the vector of predicted class probabilities, i.e., $\mathbf{x} \in \mathbb{R}^c$, where $c = 10$ denotes the number of classes,
3. PCA-reduced features, i.e., $\mathbf{x} \in \mathbb{R}^{d'}$, where $d' \ll d$ is the reduced dimension. For comparability, we use the same dimensionality $d' = c = 10$ for PCA.

Our results are shown in Fig. 1. The first column shows the accuracy vs. the number of acquired labels. We observe that using the vector of predicted probabilities not only maintains the performance of full-dimensional latent features but also surpasses it for all three investigated diversity-based heuristics. In contrast, PCA-reduced latent features result in comparable performance. The third column compares the number of acquired labels against the cumulative query time. Using the vector of predicted probabilities generally shows the lowest cumulative runtime. Compared to using the output vectors, PCA requires an extra step and is therefore somewhat weaker in terms of query times. However, using full-dimensional latent features can lead to more than four-fold increased cumulative query time depending on the heuristic, even in this relatively small toy setting. The second column then combines both plots and shows the accuracy vs. the cumulative query time, demonstrating that both label efficiency and query times benefit from our proposed method.

3 Conclusion

In this paper, we proposed to use the vector of predicted probabilities instead of the high-dimensional latent features as input to diversity-based active learning methods. As a proof-of-concept, we demonstrated on one dataset that for several diversity-based heuristics, we could strongly reduce the query time while at the same time improving the performance. Since the predicted probabilities of the unlabeled data are usually exploited anyway during the active learning process, no additional computations are required.

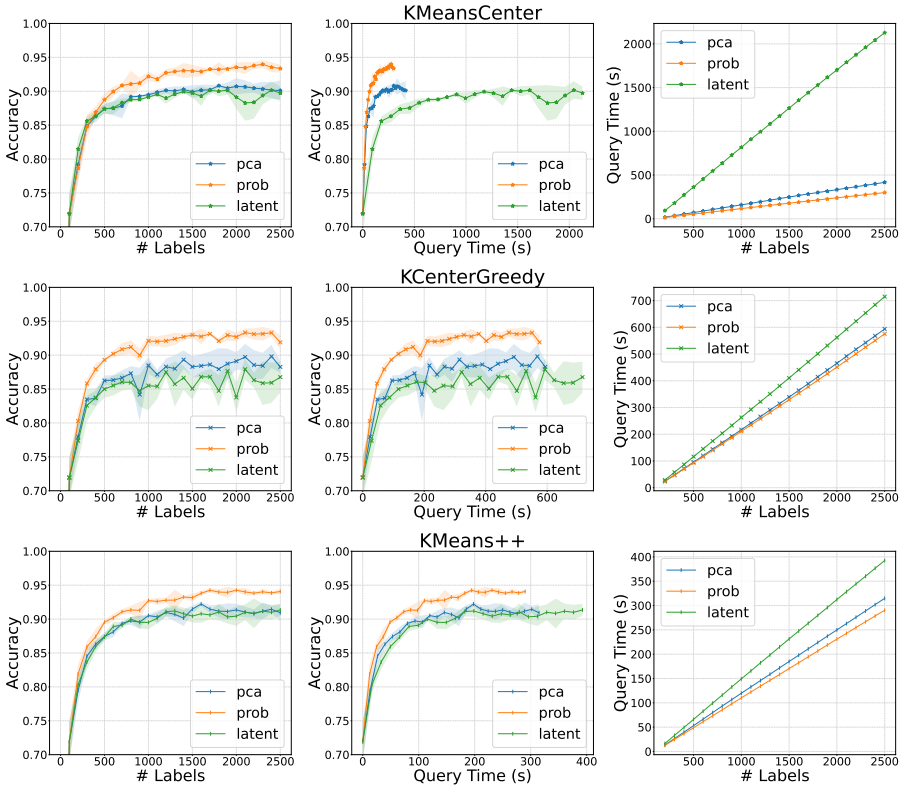


Fig. 1. Comparison of the different techniques for three different acquisition functions. The first column shows the accuracy w.r.t. the number of labels, the second column accuracy vs. cumulative query time, and the last column the cumulative query time vs. the number of acquired labels.

For future work, we would like to investigate this promising direction further, particularly how well the insights transfer to other datasets and how to best combine it with uncertainty-based methods. As an interesting observation, using samples with diverse predicted probabilities might also implicitly lead to selecting points of diverse uncertainty.

Acknowledgements

This work has been funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibilities for its content.

References

1. Abraham, A., Dreyfus-Schmidt, L.: Sample noise impact on active learning. arXiv preprint arXiv:2109.01372 (2021)
2. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. Tech. rep., Stanford (2006)
3. Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds. In: ICLR (2020)
4. Gal, Y., Islam, R., Ghahramani, Z.: Deep bayesian active learning with image data. In: ICML. pp. 1183–1192 (2017)
5. Gao, M., Zhang, Z., Yu, G., Arık, S.Ö., Davis, L.S., Pfister, T.: Consistency-based semi-supervised active learning: Towards minimizing labeling cost. In: ECCV. pp. 510–526 (2020)
6. Geifman, Y., El-Yaniv, R.: Deep active learning over the long tail. arXiv preprint arXiv:1711.00941 (2017)
7. Gissin, D., Shalev-Shwartz, S.: Discriminative active learning. arXiv preprint arXiv:1907.06347 (2019)
8. Killamsetty, K., Sivasubramanian, D., Ramakrishnan, G., Iyer, R.: Glister: Generalization based data subset selection for efficient and robust learning. vol. 35, pp. 8110–8118 (2021)
9. Kim, K., Park, D., Kim, K.I., Chun, S.Y.: Task-aware variational adversarial active learning. In: CVPR. pp. 8166–8175 (2021)
10. Kirsch, A., Van Amersfoort, J., Gal, Y.: BatchBALD: Efficient and diverse batch acquisition for deep bayesian active learning. NeuRIPS pp. 7026–7037 (2019)
11. Kothawade, S., Beck, N., Killamsetty, K., Iyer, R.: Similar: Submodular information measures based active learning in realistic scenarios. NeuRIPS **34**, 18685–18697 (2021)
12. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791, `https://doi.org/10.1109/5.726791`
13. Li, X., Guo, Y.: Multi-level adaptive active learning for scene classification. In: ECCV. pp. 234–249 (2014)
14. Lloyd, S.: Least squares quantization in PCM. IEEE Transactions on Information Theory **28**(2), 129–137 (1982)
15. McCallum, A.K., Nigam, K.: Employing EM and pool-based active learning for text classification. In: ICML. pp. 359–367 (1998)
16. Prabhu, V., Chandrasekaran, A., Saenko, K., Hoffman, J.: Active domain adaptation via clustering uncertainty-weighted embeddings. In: ICCV. pp. 8505–8514 (2021)
17. Ren, P., Xiao, Y., Chang, X., Huang, P., Li, Z., Gupta, B.B., Chen, X., Wang, X.: A survey of deep active learning. ACM Comput. Surv. **54**(9), 180:1–180:40 (2022). https://doi.org/10.1145/3472291
18. Sener, O., Savarese, S.: Active learning for convolutional neural networks: A core-set approach. In: ICLR (2018)
19. Sinha, S., Ebrahimi, S., Darrell, T.: Variational adversarial active learning. In: ICCV. pp. 5972–5981 (2019)
20. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. JMLR **2**(Nov), 45–66 (2001)
21. Wu, T.H., Liu, Y.C., Huang, Y.K., Lee, H.Y., Su, H.T., Huang, P.C., Hsu, W.H.: Redal: Region-based and diversity-aware active learning for point cloud semantic segmentation. In: ICCV. pp. 15510–15519 (2021)

22. Zhdanov, F.: Diverse mini-batch active learning. arXiv:1901.05954 (2019)

B FALCUN: A Simple and Efficient Deep Active Learning Strategy

Authors

Sandra Gilhuber, Anna Beer, Yunpu Ma, and Thomas Seidl

Venue

Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD), pages 421-439. Springer, 2024.

DOI

https://doi.org/10.1007/978-3-031-70352-2_25

Code

https://github.com/sobermeier/falcun

Declaration of Authorships

Sandra Gilhuber proposed the research idea, developed and conceptualized it with Yunpu Ma, and discussed it with Thomas Seidl. Sandra Gilhuber did the implementation. Sandra Gilhuber designed and conducted the experiments. Sandra Gilhuber and Anna Beer discussed the results and wrote the manuscript.

Copyright Notice

FALCUN: A Simple and Efficient Deep Active Learning Strategy

Sandra Gilhuber[1,2( )], Anna Beer[3], Yunpu Ma[1], and Thomas Seidl[1,2]

[1] LMU Munich, Munich, Germany
{gilhuber,seidl}@dbs.ifi.lmu.de
[2] Munich Center for Machine Learning (MCML), Munich, Germany
[3] University of Vienna, Vienna, Austria
anna.beer@univie.ac.at

Abstract. We propose FALCUN, a novel deep batch active learning method that is label- and time-efficient. Our proposed acquisition uses a natural, self-adjusting balance of uncertainty and diversity: It slowly transitions from emphasizing uncertain instances at the decision boundary to emphasizing batch diversity. In contrast, established deep active learning methods often have a fixed weighting of uncertainty and diversity, limiting their effectiveness over diverse data sets exhibiting different characteristics. Moreover, to increase diversity, most methods demand intensive search through a deep neural network's high-dimensional latent embedding space. This leads to high acquisition times when experts are idle while waiting for the next batch for annotation. We overcome this structural problem by exclusively operating on the low-dimensional probability space, yielding much faster acquisition times without sacrificing label efficiency. In extensive experiments, we show FALCUN's suitability for diverse use cases, including medical images and tabular data. Compared to state-of-the-art methods like BADGE, CLUE, and AlfaMix, FALCUN consistently excels in quality and speed: while FALCUN is among the fastest methods, it has the highest average label efficiency.

Keywords: Deep Active Learning · Supervised Learning · Diversity and Uncertainty Sampling

1 Introduction

Deep neural networks have proven their worth in various fields and are widely used for solving complex tasks. Their great success depends largely on the availability of labeled data. However, while large volumes of unlabeled data are often easily accessible, the labeling process remains time-consuming and costly, particularly in domains like medicine and industry, where experts are essential.

Active learning (AL) strategies mitigate annotation efforts by iteratively selecting and labeling the most informative instances to enhance model performance. However, the batch setting in deep AL, where multiple instances are

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-70352-2_25.

A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14943, pp. 421–439, 2024.
https://doi.org/10.1007/978-3-031-70352-2_25

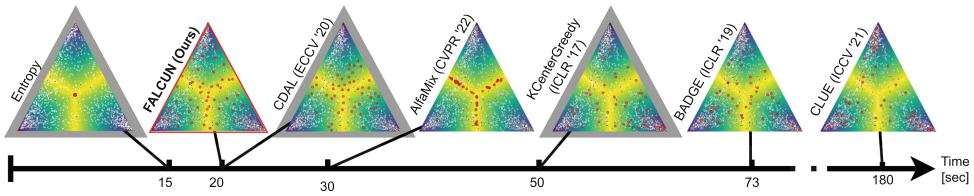


Fig. 1. Each simplex illustrates the probability space of a three-class subset of MNIST. The highest probabilities are in the corners (implied by darker colors). Small black and white dots are objects in $\mathcal{L}$ and $\mathcal{U}$, respectively. Red dots are instances selected by an AL method. FALCUN acquires objects very fast and returns a meaningful selection: gray borders imply worse quality than FALCUN.

sent to the annotator simultaneously to meet the higher data demands of deep learning and reduce re-training times, poses new challenges [2]. Specifically, the question of how to select the most informative instances while minimizing redundancy is an ongoing research topic.

To assess *diversity* and *uncertainty*, established approaches often treat the probability and latent spaces separately [14,15], requiring an additional step to merge the extracted information into a coherent acquisition. However, achieving a smooth combination of these disparate aspects can be difficult, potentially overemphasizing either uncertainty or diversity. Furthermore, a subsequent combination may rely on additional parameters [25] that are hard to select in advance. As a result, such methods might not outperform random sampling consistently, which is crucial for active learning approaches. Lastly, merging information from distinct spaces may result in highly complex methodologies, undermining their practical applicability in active learning contexts.

Moreover, using the latent representations of a deep neural network to measure diversity [2,15,18,25] can be computationally intensive due to the high dimensionality of learned features. E.g., the dimensionality of the last hidden layer for commonly used architectures (see [2,10,14]) is 512 in ResNet18, 2048 in ResNet50, and 4096 in VGG16. Thus, searching the feature space can be very time-intensive, leading to acquisition times of up to several days. Starting the labeling process on multiple days instead of requiring only one session can drive up costs immensely, e.g., if domain experts or laboratory equipment are required. Unnecessarily long computation times are also prohibitive from an ecological point of view.

We address these challenges and propose FALCUN (**F**ast **A**ctive Learning by **C**ontrastive **UN**certainty). As illustrated in Fig. 1, FALCUN queries instances that yield high-quality results for deep learning while also being faster than comparative methods. Our method exclusively operates on the output probabilities to calculate uncertainty and batch diversity. In a unified and coherent acquisition, FALCUN begins by proposing instances around the decision boundary and gradually shifts focus to diverse areas as regions of high uncertainty are increasingly explored.

The main benefits of FALCUN are:

- Label efficiency and robustness: Across varying datasets, AL settings, and model architectures, FALCUN is always among the most label-efficient methods. Among all experiments, FALCUN outperforms random sampling most often ($> 70\%$) while never performing statistically worse.
- Speed and scalability: Among competitors reaching similar accuracy, FALCUN is the fastest. FALCUN is more scalable than methods operating on the latent embeddings of a neural network.
- Diversity: Even on high-redundancy data sets, FALCUN finds a **diverse** set of instances.
- Explainability and simplicity: FALCUN is **easy** to understand and implement and, therefore, attractive for practitioners and researchers. Our code is available under https://github.com/sobermeier/falcun.

2 Related Work

AL techniques can be grouped into the following categories.

Uncertainty-based methods estimate the informativeness of an instance based on the model's predictive ambiguity. Common uncertainty estimates are margin uncertainty [16], entropy [20] or least confidence [19]. Labeling such instances should help to effectively refine the decision boundary and enhance generalization performance if included in the training [19]. Uncertainty-based sampling is widely used for its simplicity and effectiveness, especially when querying single instances or small batches at once. However, in the batch setting common for deep AL, where multiple instances are queried simultaneously, simple rank-based techniques become less label-efficient since they tend to select redundant instances. E.g., in Fig. 1, Entropy [22] as a non-diversity aware method selects highly repetitive instances.

Query-by-committee (QBC) refers to using a committee of classifiers and calculating statistical information over the varying outputs [4]. Due to the need for multiple classifiers, QBC approaches have a computational overhead and are less attractive for deep neural networks and big datasets. Deep Bayesian AL methods can be seen as a more elegant way to imitate a QBC. By using stochasticity in the prediction of a network, diverse outputs can be produced and used to calculate variations in the differing predictions for the same input. For instance, BALD [5] uses Monte-Carlo Dropout over multiple inference steps and calculates mutual information to assess the worthiness of an object. Still, such an approach requires multiple forward passes, which do not scale well to large unlabeled pools. Moreover, QBC methods also suffer from problems similar to uncertainty-based sampling in batch-setting.

Diversity-based techniques [18,21] minimize the information overlap within a batch. KCenterGreedy [18] iteratively selects the sample with the largest minimum distance to any labeled instance in the latent space to achieve decent coverage over the data space. However, only focusing on coverage can lead to selecting outliers or uninteresting instances that do not improve the performance.

Lastly, *hybrid* approaches [2,10,15] combine paradigms to overcome the challenges of solely uncertainty or diversity-based methods. Many methods perform a thorough search in the latent feature space to determine a sufficiently diverse set. E.g., BADGE [2] performs k-Means++ sampling on so-called gradient embeddings where large gradients indicate uncertainty. However, these gradient embeddings depend on the number of classes and the hidden dimensionality of the penultimate layer and thus get very high-dimensional. Other methods perform weighted k-Means clustering on the latent representations [15,25] where the weights are an uncertainty estimate and select the most central point from each cluster for annotation. Due to the repeated clustering, these methods are also computationally expensive.

AlfaMix [14] also performs k-means clustering on latent representations. In contrast to other methods, only clusters on a candidate pool determined by interpolating features in the latent space are considered. Depending on the size of the candidate pool, this increases the computational efficiency. However, as shown in Fig. 1, AlfaMix has a strong emphasis on the decision boundary, which can be problematic for highly repetitive datasets.

CDAL [1] uses a similar approach as KCenterGreedy but works on the output probabilities. It selects instances where the predicted probability is furthest away from already labeled instances. However, a problem is that some concepts in the data might be harder to learn than others. If instances get labeled, but the model needs more information in such a region, CDAL would not choose instances in the region. Task-specific hard-to-learn concepts might be ignored.

BatchBALD [10] extends BALD to the batch-setting, but has exponential time-complexity [17], making it unsuitable for our setting. Sampling from the power distribution of an uncertainty score [3,9] instead of a deterministic top k selection to increase diversity is a faster alternative. However, finding the optimal power value is hard. Small values are close to random sampling and too large values lead to a redundant selection. Thus, these methods are highly dependent on a good parameter choice.

In contrast, FALCUN uses the powering method to stay close to the original distribution instead of increasing diversity in general: it uses a dedicated diversity mechanism to be robust against parameter selection.

In summary, the main direction of deep AL research focuses on hybrid methods in the practically relevant batch setting, finding a set of informative instances with small information overlap. However, *how* to best combine uncertainty and diversity is an ongoing challenge.

3 Methodology of FALCUN

3.1 Notation

Our task is multi-class classification on an input space $\mathcal{X}$ of size N and a set of labels $\mathcal{Y} = \{1, \ldots, C\}$ for C classes. We consider pool-based AL, where a small initial labeled set $\mathcal{L} \subset \mathcal{X}$ is uniformly drawn from the unlabeled data distribution. The remaining data objects belong to the unlabeled set $\mathcal{U} = \mathcal{X} \backslash \mathcal{L}$ of

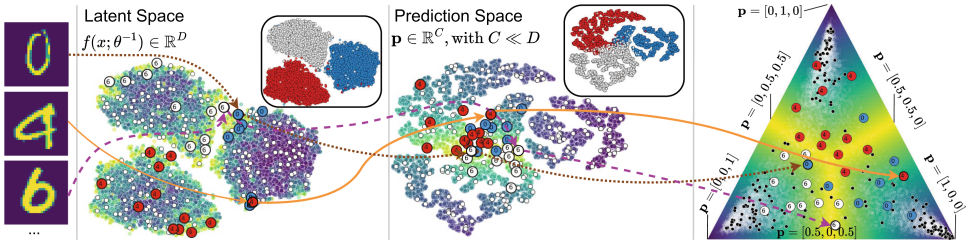


Fig. 2. FALCUN selects diverse and uncertain instances (colored circles) in the probability space (see 3-class simplex on the right). In the latent space on the left, they cover the most informative regions (yellow) while being highly diverse and stemming from different clusters. Red, white, blue imply ground truth classes. (Color figure online)

size N_u. At each AL round, Q samples are selected for annotation and retraining of the model. A classification model $f(x;\theta) \rightarrow \mathbb{R}^C$ with parameters θ maps a given input $x \in \mathcal{X}$ to a C-dimensional vector. Correspondingly, $f(x;\theta^{-1}) \rightarrow \mathbb{R}^D$ denotes the D-dimensional latent representation w.r.t. the penultimate layer of the classifier. The softmax function applied on the model output given by $f(x;\theta)$ for an object x returns the output probability vector $\mathbf{p}(x) \in [0,1]^C$. We use a standard cross-entropy loss to optimize the parameters over the labeled pool, denoted by $\mathbb{E}_{\mathcal{L}}[l_{ce}(f(x;\theta), y)]$.

3.2 Overview

Figure 2 gives an overview of FALCUN. Instead of exploiting the latent space for diversity and the probability space for uncertainty independently, FALCUN directly uses the probabilities to select diverse *and* uncertain instances. The original data inputs (left) are forwarded through the network. The second and third columns visualize the latent and the probability space in a 2D t-SNE visualization. The colors indicate uncertainty, with yellow, lighter regions indicating higher uncertainty. On the right, the 3-dimensional simplex S is given by $S = \{(p_1, p_2, p_3) | p_i \geq 0, p_1 + p_2 + p_3 = 1\}$, where p_1, p_2, p_3 denote the posterior probability for classes 1, 2, and 3, respectively. The corners indicate a high confidence for a certain class, as reflected by a darker color. The center corresponds to a uniform posterior distribution over all classes. Small black and white dots indicate objects in $\mathcal{L}$ and $\mathbf{U}$, respectively. Larger blue, red, and white circles indicate instances selected by FALCUN: they are prevalently in very informative regions in the latent space while being highly diverse.

3.3 Acquisition

Uncertainty Component. For uncertainty, we use the margin uncertainty, i.e., the difference between the probabilities of its two most probable classes:

$$u(x) := 1 - (\mathbf{p}(x)[c_1] - \mathbf{p}(x)[c_2]) \in [0,1], \tag{1}$$

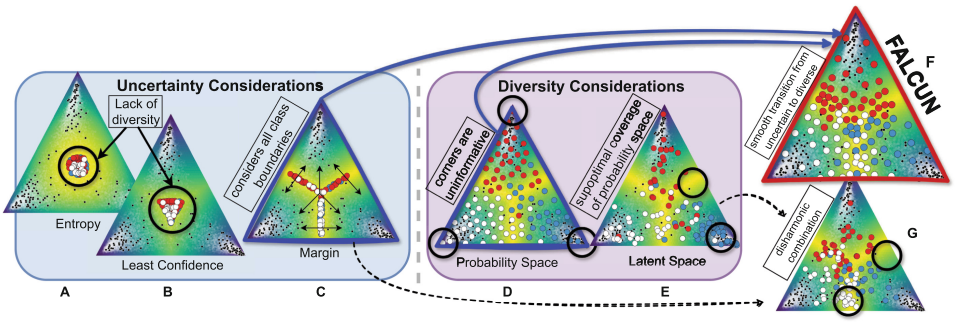


Fig. 3. Uncertainty Considerations (Left): In contrast to least confidence and entropy, the margin estimate focuses on the class boundaries between all class pairs, covering a more diverse spectrum. **Diversity Considerations** (Middle): Maximizing diversity in the probability space automatically covers diverse and uncertain regions, whereas using latent features for diversity makes a harmonic combination with uncertainty harder. **Final** (Right): **FALCUN** prefers instances at the decision boundary with a smooth transition to diverse regions.

where $0 \leq u(x) \leq 1$. Margin is a common choice for uncertainty [3,8,16] and naturally captures class boundaries. As illustrated in Fig. 3, margin (C) emphasizes diverse regions to be of equal interest and naturally captures more dissimilar concepts than comparable other uncertainty estimates such as entropy (A) or least confidence (B) [19]. The reason is that the margin's extremal function has no global optimum, but its optima lie on the pairwise class boundaries in the probability space. Thus, margin uncertainty is powerful [3,8,25] and allows an intuitive combination with diversity, as we show in the following.

Diversity Component. To estimate diversity, we follow a similar notion as [1], measuring class-wise, contextual diversity in the probability space rather than feature-wise diversity in the possibly very high-dimensional embedding space where we might run into curse-of-dimensionality issues or computational overhead. More precisely, we measure the distances between two instances x_1 and x_2 based on their probabilities using the L1 norm $|| \cdot ||_1$:

$$dist(\mathbf{p}(x_1), \mathbf{p}(x_2)) := ||\mathbf{p}(x_1) - \mathbf{p}(x_2)||_1 = \sum_{i=1}^{C} |p_i(x_1) - p_i(x_2)| \tag{2}$$

Calculating distance in the probability space accelerates computation without neglecting generalization performance [6]. Moreover, maximizing diversity in the probability space as visualized in Fig. 3 - D, automatically covers diverse and uncertain regions. In contrast, using latent features for diversity makes a harmonic combination with uncertainty harder, potentially resulting in suboptimal coverage of the probability space (see Fig. 3 E and G).

However, without careful initialization, which is hard when the query batch is still empty, maximizing diversity in the probability space also targets uninfor-

Algorithm 1. Our AL Algorithm FALCUN

Input: Unlabeled data pool $\mathcal{U}$, initially labeled data pool $\mathcal{L}$, number of acquisition rounds R, query-size Q, model $f(x;\theta)$, relevance factor γ

Train initial weights θ_0 on $\mathcal{L}$ by minimizing $\mathbb{E}_{\mathcal{L}}[l_{ce}(f(x;\theta),y)]$
for $r = 1, 2, \ldots, R$ **do**
  Initialize empty query set: $\mathcal{Q} = \{\}$
  $\forall x \in \mathcal{U}$: Compute class probabilities $\mathbf{p}(x)$
  $\forall x \in \mathcal{U}$: Initialize $u(x)$ and $d(x)$ with Equations (1) and (3)
  for $q = 1, \ldots, Q$ **do**
    $\forall x \in \mathcal{U}$: Calculate relevance score $r(x)$ with Equation (5)
    Sample according to Equation (6)
    $\mathcal{Q} = \mathcal{Q} \cup x_q$
    $\forall x \in \mathcal{U}$: Update diversity values $d(x)$ using Equation (4)
  end for
  Receive new labels from oracle for instances in $\mathcal{Q}$
  $\mathcal{L} = \mathcal{L} \cup \mathcal{Q}, \mathcal{U} = \mathcal{U} \setminus \mathcal{Q}$
  Train new model θ_r from scratch on $\mathcal{L}$ by minimizing $\mathbb{E}_{\mathcal{L}}[l_{ce}(f(x;\theta),y)]$
end for
return Final parameters θ_R obtained in round R

mative samples in the class corners. A good starting point is to focus on instances that provide different context-specific information to already well-distinguishable concepts. This can be seen as a way of diversity to the confident class corners in the simplex. The margin estimate gives us a good starting point for such diversity. Instances that receive the highest scores are (1) farthest away from the highly confident corners and (2) close to other classes. Without the second proximity consideration, focusing solely on maximizing distance to corners could bias towards the central region where all classes are equally probable (Revisit A, B, and C in Fig. 3). Margin *uncertainty* is high for instances from concepts that are *diverse* from concepts that the model can already classify confidently and, thus, naturally incorporates a diversity aspect.

Further details on the correlation between margin uncertainty and the distance to confident classes can be found in the supplementary material. Thus, we initialize the diversity score with the pre-calculated margin uncertainty and iteratively update it with each selected sample x_q:

$$d'_{init}(x) := u(x) \qquad (3) \qquad d'(x) \leftarrow \min(d'(x), dist(\mathbf{p}(x), \mathbf{p}(x_q)) \qquad (4)$$

As diversity values can only decrease, the initialization in Eq. (3) ensures that the closer objects are to the confident corners, the less likely they will be selected. By updating the diversity score using Eq. (4), instances near objects in the current query batch receive lower scores and are less likely to be selected. Finally, we linearly normalize the values to $[0, 1]$ to align them with the uncertainty scores using min-max-normalization.

Final Relevance Score. For every point x, we calculate a relevance score $0 \leq r(x) \leq 2$, which changes over the course of each AL round. We combine the uncertainty and the diversity component by defining $r(x)$ as the sum of the uncertainty $u(x)$ and the normalized adaptive diversity score $d(x)$:

$$r(x) := u(x) + d(x). \tag{5}$$

Note that the values in $u(x)$ are static within one acquisition, but the diversity scores $d(x)$ are updated with every chosen query instance. Thus, the diversity slightly overshadows when the regions with the highest uncertainty are exhausted. When there is decent coverage in the probability space and diversity scores denote a uniform distribution, the focus is more on uncertainty. Hence, there is always a natural balance between uncertain and diverse selection depending on the current query batch. We choose x as the next query sample x_q with probability

$$x_q \sim \frac{r(x)^\gamma}{\sum_{x \in \mathcal{U}} r(x)^\gamma}, \tag{6}$$

where γ is a parameter that controls the influence of the relevance scores. $\gamma = 0$ corresponds to a uniform selection, and larger values for γ result in a stronger focus on the calculated relevance scores getting more and more deterministic (rich values get richer). Thus, γ controls the trade-off between exploration (more randomness) and exploitation (more focus on larger values in $r(x)$). See also Fig. 4, which shows the selection probabilities of points depending on their relevance scores for different values of γ. Note that we do not need γ to ensure diversity as in [3]. We use it to reduce the risk of an overly biased selection. We analyze the effect of γ and show the importance of a dedicated diversity scheme in our ablation study in Fig. 13. By combining uncertainty and diversity with our initialization, we can exploit the probability space in a harmonic way as shown in Fig. 3 F. One AL round stops when the batch $\mathcal{Q}$ contains B samples and returns the query batch $\mathcal{Q}$, which will be sent to the oracle for annotation. The pseudo-code is shown in Algorithm 1.

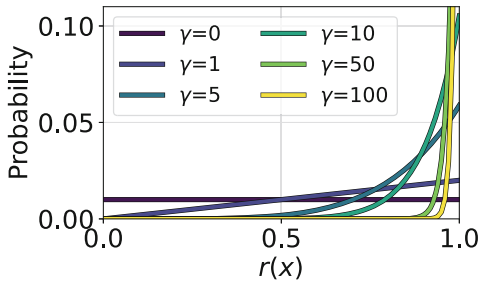


Fig. 4. Selection probability of an instance x for different γ values as a function of its relevance score $r(x)$.

Table 1. Data set properties: number of points N, number of classes C, and number of input features F.

Type	Data set	N	C	F
Image (Gray)	MNIST	60,000	10	28x28
	RMNIST	60,000	10	28x28
	FashionMNIST	60,000	10	28x28
	EMNIST	131,600	47	28x28
Image (Color)	SVHN	73,257	10	32x32x3
	BloodMNIST	11,959	8	28x28x3
	DermaMNIST	7,007	7	28x28x3
	CIFAR10	60,000	10	28x28x3
Tabular	OpenML-6	16,000	26	17
	OpenML-156	800,000	5	11
	OpenML-155	829,201	10	11

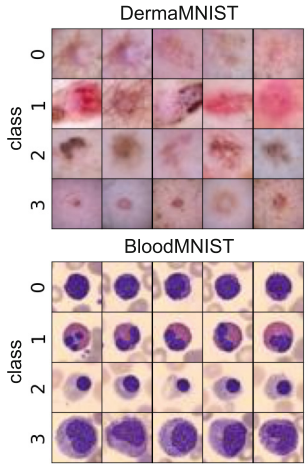


Fig. 5. Exemplary images of the two medical datasets.

4 Experiments

We evaluate the effectiveness of established AL methods and FALCUN regarding quality and acquisition runtime in isolation as well as in combination to get a complete picture. We use a broad range of datasets including grayscale images (MNIST [12], FashionMNIST [23], and EMNIST), colored images (CIFAR10 [11], SVHN [13], BloodMNIST, DermaMNIST [24]), and tabular datasets from the OpenML benchmark[1] suite (Ids: 6, 155, 156). BloodMNIST and DermaMNIST are challenging medical image datasets showcasing a task where labeling experts are limited and costly. Figure 5 shows some examples. Within a class, images can be very similar, s.t. their information is redundant. A good AL strategy should avoid selecting such repetitive instances to optimize label efficiency. To further assess the capabilities to sample a diverse subset, we include redundant versions of MNIST named RMNIST containing duplicate images (comparable to [10]). We randomly keep 10% unique original images and fill the rest with duplicated versions with added Gaussian noise. We vary the redundancy ratio in an extra experiment. Table 1 summarizes the data properties. For grayscale data we use a LeNet, a learning rate of 0.01 and train for 20 epochs. For colored data we use pre-trained Resnet18, and ResNet50, a learning rate of 0.001 and stop when a training accuracy of 99% is reached. We investigate whether the results are similar without pre-trained weights and when initializing the model with the weights from the previous round as proposed in [14]. For tabular data we use a simple multi-layer-perceptron (MLP) with two layers as proposed in [2] (hidden

[1] https://www.openml.org/.

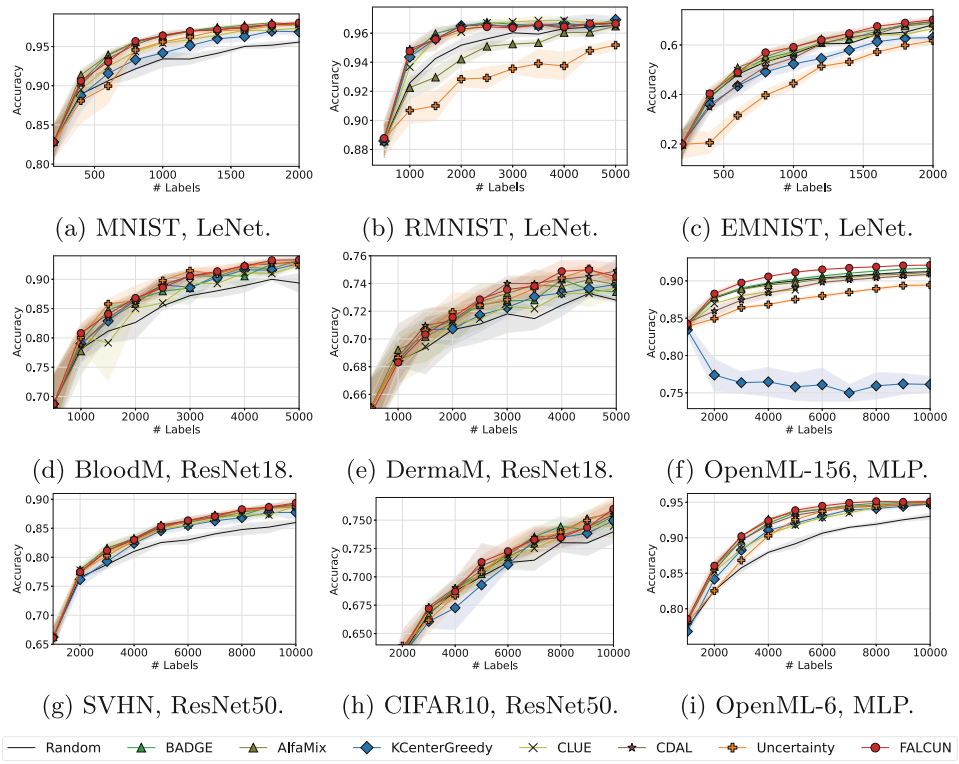


Fig. 6. Average test accuracy vs labeling budget for all active learning methods evaluated on greyscale (a, d), RGB (b, e) and tabular data (c, f).

dimensionality 1024), a learning rate of 0.0001 and use early stopping when a training accuracy of 99% is reached. We use an Adam optimizer. All experiments are performed five times with different seeds. We compare to state-of-the-art hybrid methods: BADGE [2], CDAL [1], CLUE [15], and AlfaMix [14]. We include a diversity baseline: KCenterGreedy [18], an uncertainty baseline: entropy sampling [19], and the passive baseline random sampling. For FALCUN, we set $\gamma = 10$. Further details are given in the publicly available code base.

4.1 Label Efficiency

Figure 6 shows the learning curves of diverse architectures and query sizes for evaluated datasets. The x-axis depicts the labeling budget, and the y-axis gives the average accuracy for varying AL methods. We see that FALCUN is among the best-performing methods for varying query sizes, data types, and model architectures. FALCUN also yields the strongest results on the tabular data: in contrast to all other competitors, it consistently outperforms random sampling on the Openml-156 dataset. Note that the ranking of the best-performing meth-

Table 2. Avg. Accuracy on CIFAR10 with varying architectures and settings. BB = backbone model, P = Pre-trained weights are used, Ctl = Continual setting where weights are not reset after each AL round, B=Labeling budget. FALCUN has most often **best** (bold) or second best performance (underlined).

BB	Ctl	P	B	CLUE	BADGE	CDAL	AlfaMix	Random	FALCUN
Resnet50		✓	6000	71.7	72.1	71.9	71.8	71.3	**72.3**
		✓	10000	74.5	75.3	75.5	75.6	74.0	**76.0**
	✓		6000	**52.0**	51.9	51.4	51.6	51.1	**52.0**
	✓		10000	57.5	58.6	**59.3**	58.3	57.4	58.5
Resnet18		✓	6000	70.1	69.9	69.8	69.9	69.4	**70.2**
		✓	10000	73.6	**74.0**	73.5	73.6	72.3	73.5
	✓		6000	54.8	55.6	55.2	55.8	54.7	**55.9**
	✓		10000	60.5	60.7	**61.0**	60.5	59.2	60.9

Table 3. Avg. Accuracy on CIFAR10 with pre-trained Resnet50 using initial pool sizes (I) and query sizes (QS). We report budgets (B) after the first and last acquisition. FALCUN performs well with varying AL settings.

I	QS	B	CLUE	BADGE	CDAL	AlfaMix	Random	FALCUN
1000	1000	2000	63.4	63.4	63.4	62.9	63.2	**63.7**
		10000	74.5	75.3	75.5	75.6	74.0	**76.0**
2000	2000	4000	68.4	68.4	68.5	69.3	**69.5**	68.8
		10000	74.9	75.3	75.0	**75.7**	71.7	75.0
5000	5000	10000	74.6	75.0	74.2	75.0	74.0	**75.3**
		20000	78.6	78.8	79.3	79.3	76.9	**79.3**
5000	7500	12500	75.2	75.9	75.2	76.4	75.1	**76.4**
		22500	78.0	79.6	**79.8**	79.3	77.9	79.6

ods is not the same over varying settings. E.g., Entropy, an only uncertainty-based technique, yields good results on BloodMNIST but underperforms on certain other datasets such as EMNIST, RMNIST or Openml-156. In contrast, KCenterGreedy, a solely diversity-based approach, only yields fairly good results on the highly redundant dataset RMNIST but performs poorly on Openml-156. Not surprisingly, some datasets and settings benefit more from uncertainty, and others might work better with diversity. Table 2 show results on CIFAR10 when varying the backbone (BB), using pre-training or not (P) and using continual training instead of starting from scratch after every AL round (Ctl) for varying budgets (B). Most often, FALCUN yields best or second best results. Table 3 shows results when varying the initial pool size (I) and query size (QS) for different Budgets (B). Again, FALCUN yields best or second best results frequently. All in all, FALCUN is robust across varying settings.

Dealing with Redundancy. We especially want to emphasize that though only operating on the output probabilities, FALCUN's success is not diminished on RMNIST. Figure 8 shows how the performance of all AL methods drops for varying redundancy ratios of the RMNIST dataset. Besides Entropy sampling, AlfaMix's quality decreases rapidly for highly redundant datasets. We hypothesize this is due to oversampling the decision boundary, as visualized in Fig. 1. We provide all learning curves in the supplementary materials.

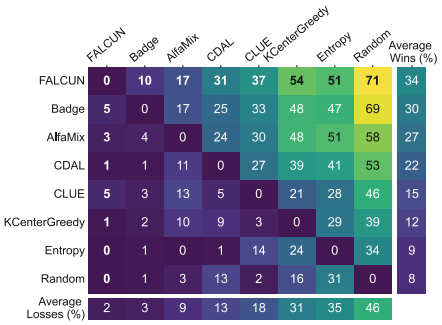


Fig. 7. Dueling matrix: The last column gives the percentage of wins of the respective method. The last row gives the percentage of losses.

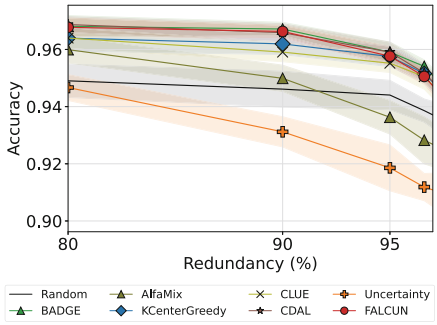


Fig. 8. Final average test accuracy for varying redundancy ratios.

Dueling Matrix Over All Experiments. Designing a robust method is hard when the characteristics of a dataset are unknown in advance. Moreover, in AL, it is hard to compare all learning curves from all experiments, and sometimes, a clear winner is hard to find. Hence, similar to previous works [2,7,14], we provide a dueling matrix for a comprehensive analysis of the methods' overall performance. The column-wise entries in the matrix in Fig. 7 show the amount of **losses**, and the row-wise entries indicate the amount of **wins** against each other method (in %). A win means that for a specific experimental setting, i.e., a specific dataset, acquisition round, query size, and model architecture, comparing the results of 5 runs, a method has statistically better accuracy than the other method (with p-value=0.05).

A loss is defined analogously. Losses and wins do not necessarily sum up to 100% as the two methods can perform comparably well with no statistical difference. When discussing the quality of an AL method, it is hence important to evaluate the wins *and* losses. The bottom row and the rightmost column denote the average losses and wins over all experiments compared to all other AL methods. FALCUN is consistently strong over a wide range of datasets, as the dueling matrix in Fig. 7 shows. FALCUN has the most wins (highest numbers in every column) compared to every other method and the most wins over random

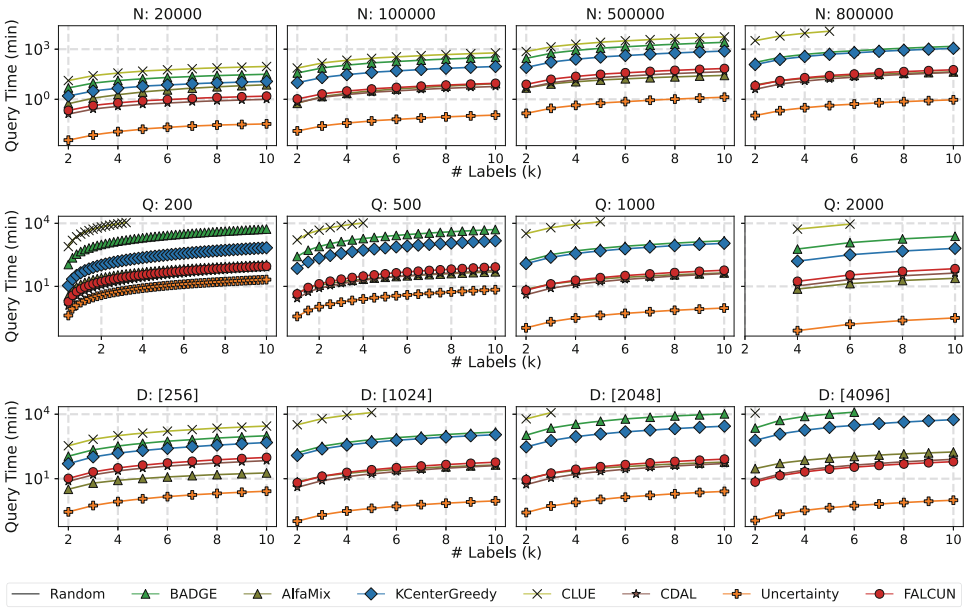


Fig. 9. Average cumulated acquisition times (y-axis) on a log-scale vs. annotated samples (x-axis) over varying unlabeled pool sizes N (first row), query sizes Q (second row), and dimensionality of the penultimate layer D (third row).

sampling. Simultaneously, it has the fewest losses. Only FALCUN is *never worse than random sampling*, one of the most important criteria for successful AL methods.

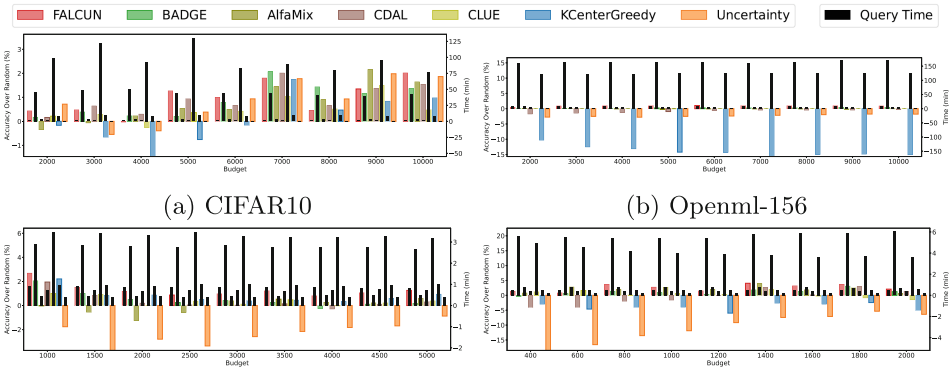
4.2 Query Time Efficiency

The training for the grayscale image datasets and tabular datasets is arguably fast (around 1 min for the last AL round). For the colored image data, training takes around 75 min in the last round. In such situations, the limiting factor for the overall runtime is the query time. We systematically analyzed the scalability of all tested methods by varying dataset size, query size, and hidden dimensionality of the multilayer perceptron evaluated for the largest of all datasets (i.e., Openml-156). We stopped each experiment after ten days (e.g. CLUE). The results are shown in Fig. 9. FALCUN denotes fast and robust runtimes over varying settings, being comparably fast as CDAL and particularly robust to varying hidden dimensinoality. We summarize these extensive experiments by giving the smallest and largest average query times among the scalability analysis in Table 4. Moreover we provide runtime complexities for all methods. Note that the runtime complexity of our acquisition is dependent on the size of the unlabeled pool, the query size, and the number of classes ($\mathcal{O}(Q \cdot N_u \cdot C)$) but not on the hidden dimensionality D. BADGE, one of the strongest competitors regarding label efficiency, has a worse runtime complexity with $\mathcal{O}(Q \cdot N_u \cdot C) \in \mathcal{O}(Q \cdot N_u \cdot C \cdot D)$.

Table 4. Time Complexity w.r.t. query size Q, Dimensionality of latent features D, unlabeled pool size N_u, number of classes C, labeled pool size N_l, number of cluster rounds i, and a method-specific candidate pool in AlfaMix N_{cp} with $N_{cp} \leq N_u$., final min. and max. average cumulated query time among the scalability analysis.

AL Strategy	Time Complexity	min	max
Entropy	$\mathcal{O}(N_u)$	1.8 sec	21 min
CDAL	$\mathcal{O}(N_l \cdot N_u \cdot C + Q \cdot N_u)$	1 min	80 min
FALCUN	$\mathcal{O}(Q \cdot N_u \cdot C)$	1.5 min	97 min
AlfaMix	$\mathcal{O}(Q \cdot N_{cp} \cdot i \cdot D)$	7.3 min	175 min
KCenterGreedy	$\mathcal{O}(N_l \cdot N_u \cdot D + Q \cdot N_u)$	11.8 min	25 h
BADGE	$\mathcal{O}(Q \cdot N_u \cdot C \cdot D)$	31.5 min	208 h
CLUE	$\mathcal{O}(Q \cdot N_u \cdot i \cdot D)$	92 min	>227 h

That leads to multiple times higher run times compared to FALCUN (208hrs in the worst case for BADGE vs 97minutes for FALCUN). CDAL, followed closely by FALCUN, is the fastest among all tested methods. In the fastest setting, when the unlabeled pool contains 20,000 objects, FALCUN is only half a minute slower than CDAL. In the most challenging setting with a latent dimension of 4096, FALCUN is only 17% slower.



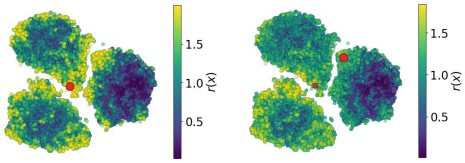
(a) CIFAR10 (b) Openml-156

(c) RMNIST (d) EMNIST

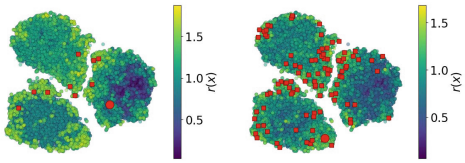
Fig. 10. Runtimes (black bars, smaller is better) and improvement over random sampling in average test accuracies (colored bars, larger is better) for all acquisition rounds for tabular data (Openml-155 and Openml-156) and grayscale data (RMNIST, EMNIST).

Considering quality and runtime together, Fig. 10 shows the improvement over random sampling in terms of average accuracy per method (colored bars) and the corresponding query time in minutes in a certain acquisition round (black thin bars) for all tested methods. *Large accuracy bars are better* whereas

smaller time bars are better. FALCUN (red bars) has strong performance on all datasets and never has worse average accuracy than random sampling (i.e., values smaller than zero). CLUE and especially BADGE perform on par in some settings, but their query times are much higher, in some cases up to > 200 hours. AlfaMix is fast and has good quality on Openml-155 and decent performance on EMNIST. However, AlfaMix is prone to duplicates: it performs even worse than random sampling on RMNIST in many acquisition rounds. CDAL is quite fast but performs worse than random sampling more often, especially for small budgets on EMNIST and Openml-156. Entropy is fast, but not label-efficient. KCenterGreedy is fast for smaller datasets (e.g., RMNIST and EMNIST) but does not scale well to larger datasets (see Openml-156) and is only comparably label-efficient for the redundant dataset RMNIST because it has the strongest emphasis on maximizing diversity. FALCUN has a robust performance across all datasets and low query times (never above 10 min).

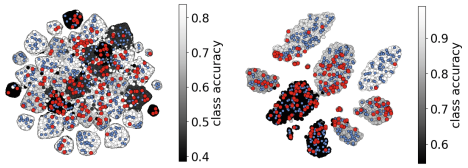


(a) 1^{st} Query. (b) 2^{nd} Query.



(c) 10^{th} Query. (d) 100^{th} Query.

Fig. 11. Exemplary course of relevance scores $r(x)$ and their dependency of selected queries (red) on 3-class MNIST, t-SNE visualization. (Color figure online)



(a) Openml-156. (b) BloodMNIST.

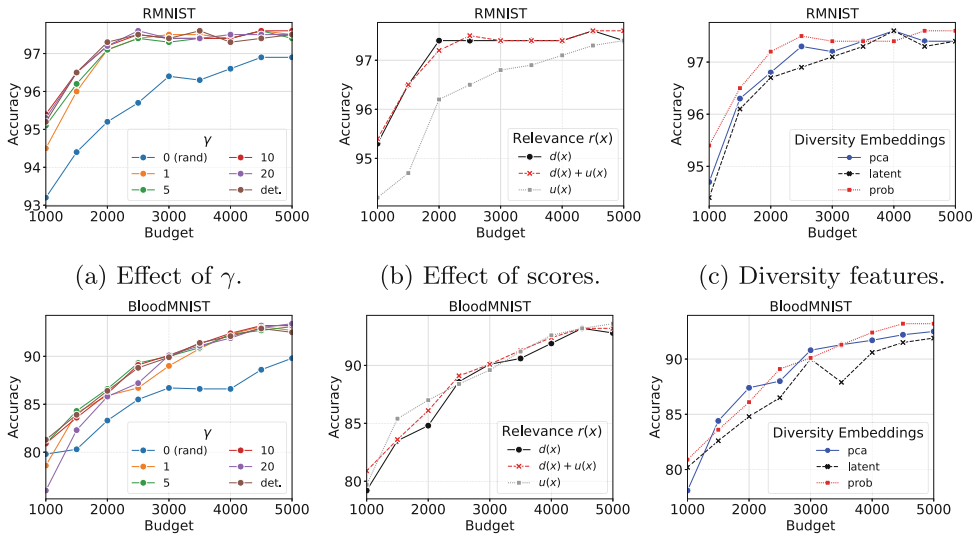
Fig. 12. Hue in the t-SNE visualizations indicates the predictive accuracy of the model on the respective class. Initially sampled objects are blue, samples chosen by FALCUN in the first acquisition round are red. FALCUN selects diverse instances favoring classes that are harder to distinguish by the current model: "darker" classes contain more red dots. (Color figure online)

4.3 Qualitative Evaluation

Figure 11 illustrates the selection of instances and the course of FALCUN's relevance scores $r(x)$ over one acquisition round on a 3-class MNIST task (also

used for the visualization in Fig. 2) for better interpretability. Yellow regions indicate a high relevance score promoting regions of high interest. Initially, all instances with high uncertainty, primarily located at the decision boundary, receive higher scores (see Fig. 11a). The score in the surrounding of the selected instance (red circle) gets darker as the objects located close to it receive a smaller diversity score (see Fig. 11b). In the first iterations, uncertain, but still diverse instances are preferred. In Fig. 11d we derive a diverse set located in all three clusters mainly consisting of objects from uncertain areas.

In Fig. 12, we analyze FALCUN's selection on Openml-156 (Fig. 12a) and BloodMNIST (Fig. 12b). It effectively finds instances majorly located in regions where the classifier has more confusion (darker areas) while still enhancing diversity and not oversampling certain regions. E.g., on the right, most instances are chosen from the two most uncertain classes ($\sim 55\%$ accuracy). In contrast, only two objects are selected from the most confident class where the model already achieves $\sim 99\%$ accuracy.



(a) Effect of γ. (b) Effect of scores. (c) Diversity features.

(d) Effect of γ. (e) Effect of scores. (f) Diversity features.

Fig. 13. Ablation Study on RMNIST (top row) and BloodMNIST (bottom row).

4.4 Ablation

Effect of γ. In Figs. 13a and 13d, we vary γ, where smaller values lean towards uniform selection and larger values lean towards deterministic selection, including a completely deterministic selection (det.). While a random selection ($\gamma = 0$, blue line) is always worst, we see that the exact choice of γ does not largely affect the performance. Having a value between 5 and 20 yields very robust

and consistent results. A deterministic selection seems similarly strong despite a few fluctuations. However, we argue that we should stick to our probabilistic selection so as not to end up in a failure mode due to highly biased selection.

Effect of Scores. Figures 13b and 13e show the results when switching off either the uncertainty or the diversity component to calculate the final relevance score. For RMNIST, considering uncertainty without diversity yields the worst results. Hence, powering similar to [3] without a dedicated diversity function is less effective for highly redundant datasets. BloodMNIST benefit more from uncertainty than from diversity. In general, our experiments show that sometimes uncertainty and sometimes diversity are more important. However, knowing which type is needed in a real-world scenario is notoriously hard when there is almost no information. In contrast, our combined score is always among the best, and due to the robustness across datasets, it is a highly attractive choice.

Effect of Diversity Features. Lastly, we investigate the performance when calculating diversity on the latent embeddings instead of the final output probabilities. As a simple baseline we also perform PCA on the latent features and use the result as input for the diversity component (see Figs. 13c and 13f). Interestingly, using latent features is worst in many situations. We assume this is due to curse-of-dimensionality issues. Furthermore, using the probability vector is almost always the best method. We hypothesize that using the probability space for uncertainty and diversity leads to a more harmonized selection. Our diversity in the probability space also indirectly covers uncertain regions, and the margin uncertainty function indirectly covers diverse concepts. Combining two isolated scores can be tricky since it could unintentionally set a too strong focus on one or the other component.

5 Conclusion

We introduced FALCUN, a novel deep AL method that employs a natural transition from emphasizing uncertain instances at the decision boundary towards enhancing more batch diversity. This natural balance ensures robust label efficiency on varying datasets, query sizes, and architectures, even on highly redundant datasets. As FALCUN only operates on the output probability vectors, it achieves faster acquisition times than many established methods performing a search through the high-dimensional embedding space of a neural network.

References

1. Agarwal, S., Arora, H., Anand, S., Arora, C.: Contextual diversity for active learning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12361, pp. 137–153. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58517-4_9
2. Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds. In: ICLR (2020)

3. Bahri, D., Jiang, H., Schuster, T., Rostamizadeh, A.: Is margin all you need? an extensive empirical study of active learning on tabular data. arXiv preprint arXiv:2210.03822 (2022)
4. Dagan, I., Engelson, S.P.: Committee-based sampling for training probabilistic classifiers. In: Machine Learning Proceedings 1995, pp. 150–157 (1995)
5. Gal, Y., Islam, R., Ghahramani, Z.: Deep bayesian active learning with image data. In: ICML, pp. 1183–1192 (2017)
6. Gilhuber, S., Berrendorf, M., Ma, Y., Seidl, T.: Accelerating diversity sampling for deep active learning by low-dimensional representations. In: Kottke, D., Krempl, G., Holzinger, A., Hammer, B. (eds.) Proceedings of the Workshop on Interactive Adaptive Learning co-located with European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2022), Grenoble, France, September 23, 2022. CEUR Workshop Proceedings, vol. 3259, pp. 43–48 (2022)
7. Gilhuber, S., Busch, J., Rotthues, D., Frey, C.M., Seidl, T.: Diffusal: Coupling active learning with graph diffusion for label-efficient node classification. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 75–91. Springer (2023)
8. Jiang, H., Gupta, M.R.: Bootstrapping for batch active sampling. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining pp. 3086–3096 (2021)
9. Kirsch, A., Farquhar, S., Atighehchian, P., Jesson, A., Branchaud-Charron, F., Gal, Y.: Stochastic batch acquisition for deep active learning. arXiv preprint arXiv:2106.12059 (2021)
10. Kirsch, A., Van Amersfoort, J., Gal, Y.: Batchbald: efficient and diverse batch acquisition for deep bayesian active learning. NeuRIPS, pp. 7026–7037 (2019)
11. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
12. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
13. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. Neural Information Processing Systems (NIPS) (2011)
14. Parvaneh, A., Abbasnejad, E., Teney, D., Haffari, G.R., Van Den Hengel, A., Shi, J.Q.: Active learning by feature mixing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12237–12246 (2022)
15. Prabhu, V., Chandrasekaran, A., Saenko, K., Hoffman, J.: Active domain adaptation via clustering uncertainty-weighted embeddings. In: ICCV, pp. 8505–8514 (2021)
16. Roth, D., Small, K.: Margin-based active learning for structured output spaces. In: European Conference on Machine Learning, pp. 413–424 (2006)
17. Rubashevskii, A., Kotova, D., Panov, M.: Scalable batch acquisition for deep bayesian active learning. In: Proceedings of the 2023 SIAM International Conference on Data Mining (SDM), pp. 739–747. SIAM (2023)
18. Sener, O., Savarese, S.: Active learning for convolutional neural networks: a core-set approach. In: ICLR (2018)
19. Settles, B.: Active learning literature survey (2009)
20. Shannon, C.E.: A mathematical theory of communication. ACM SIGMOBILE mobile computing and communications review **5**(1), 3–55 (2001)
21. Sinha, S., Ebrahimi, S., Darrell, T.: Variational adversarial active learning. In: ICCV, pp. 5972–5981 (2019)

22. Wang, D., Shang, Y.: A new active labeling method for deep learning. In: 2014 International Joint Conference on Neural Networks (IJCNN), pp. 112–119. IEEE (2014)
23. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747 (2017)
24. Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., Ni, B.: Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. Sci. Data **10**(1), 41 (2023)
25. Zhdanov, F.: Diverse mini-batch active learning. arXiv:1901.05954 (2019)

Supplementary Materials to "FALCUN: A Simple and Efficient Deep Active Learning Strategy"

Sandra Gilhuber[1,2], Anna Beer[3], Yunpu Ma[1], and Thomas Seidl[1,2]

[1] LMU Munich, Germany {gilhuber,seidl}@dbs.ifi.lmu.de
[2] Munich Center for Machine Learning (MCML), Germany
[3] University of Vienna, Austria anna.beer@univie.ac.at

A Connection of Uncertainty and Initial Diversity

The margin uncertainty estimates the uncertainty of a model about its prediction $\mathbf{p}(x)$ of an instance x and is defined as

$$u(x) := 1 - (\mathbf{p}(x)[c_1] - \mathbf{p}(x)[c_2]) \in [0, 1]. \tag{1}$$

It has the smallest values, i.e., the highest confidence, when the prediction for one class is 1, and the highest scores, i.e., the highest uncertainty, at the decision boundary. Despite assessing the model's uncertainty, we can interpret margin *uncertainty* as an estimate of an instance's *diversity* relative to the concepts already learned by the model regarding the predicted class. In other words, margin uncertainty can be seen as the distance of an instance's prediction probability from the one-hot encodings of the two most probable classes, as shown below.

Let $\hat{\mathbf{p}}$ be a one-hot encoding $\hat{\mathbf{p}}_i = \delta_i k$ of $\mathbf{p}(x)$ where $\delta_i k$ is the Kronecker Delta function and $k = \arg\max_c(\mathbf{p}(x))$ is the index of the most probable class. Further, let $\hat{\mathbf{p}}_2$ be a one-hot encoding $\mathbf{p}(x)$ with $\hat{\mathbf{p}}_{2_i} = \delta_i k_2$ where $k_2 = \arg\max_{2_c}(\mathbf{p}(x))$ is the index of the second most probable class.

$$\begin{aligned} d_{init}(x) =& 1 - 1/2 \cdot |\underbrace{dist(\hat{\mathbf{p}}, \mathbf{p}(x))}_{\substack{\text{Distance to most} \\ \text{probable class}}} - \underbrace{dist(\hat{\mathbf{p}}_2, \mathbf{p}(x)))}_{\substack{\text{Distance to 2nd most} \\ \text{probable class}}}| \\ =& 1 + 1/2 \cdot \left(\sum_{i=1}^{C} |\hat{p}_i - p_i(x)| - \sum_{i=1}^{C} |\hat{p}_{2_i} - p_i(x)| \right) \\ =& 1 + 1/2 \cdot \left(1 - \mathbf{p}(x)[c_1] + \sum_{i=1}^{C} (i \neq k) p_i(x) - \left(1 - \mathbf{p}(x)[c_2] + \sum_{i=1}^{C} (i \neq k_2) p_i(x) \right) \right) \\ =& 1 + 1/2 \cdot (2 \cdot (1 - \mathbf{p}(x)[c_1]) - 2 \cdot (1 - \mathbf{p}(x)[c_2])) = 1 - (\mathbf{p}(x)[c_1] - \mathbf{p}(x)[c_2]) = u(x) \end{aligned} \tag{2}$$

Therefore, we initialize the diversity score with the uncertainty estimate: $d_{init}(x) := u(x)$.

B Experimental Details

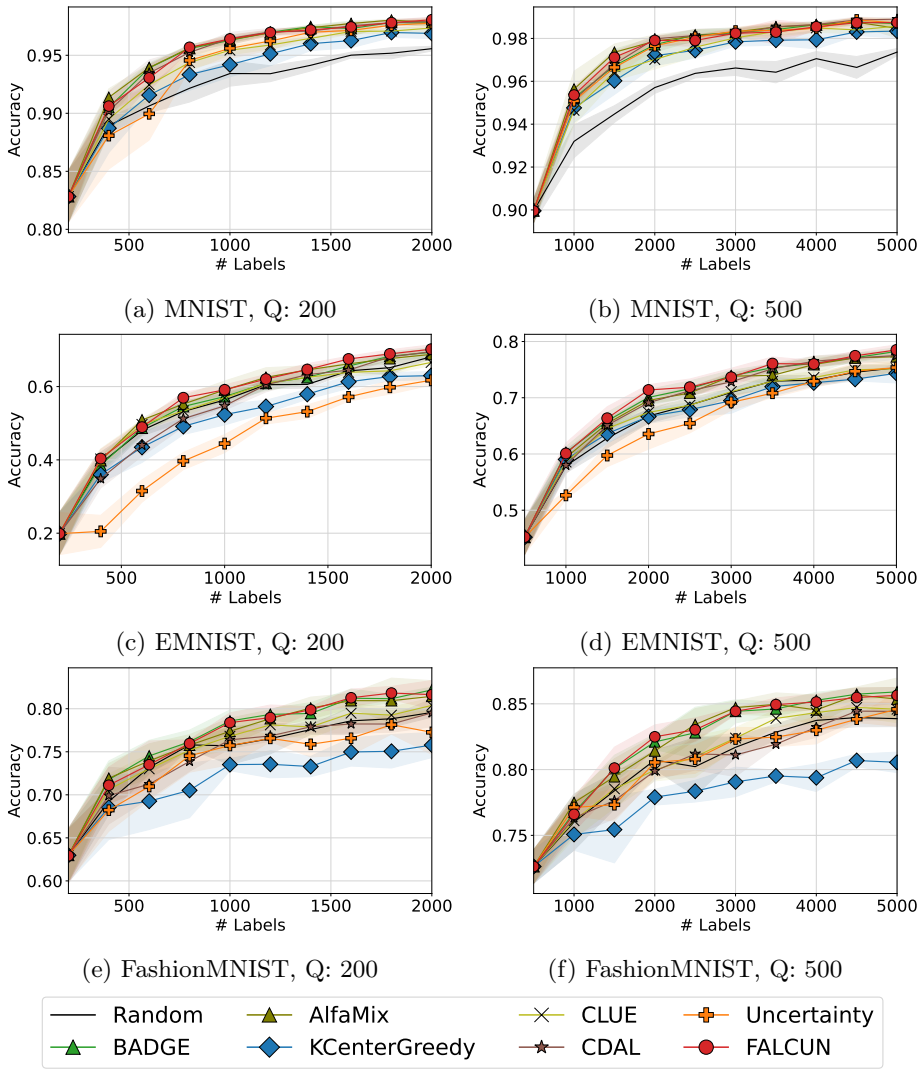
The implementation is in Python and uses PyTorch [1], NumPy, and scikit-learn [2]. Our experiments have been evaluated on GPUs (NVIDIA GeForce RTX 2080 Ti) in an Ubuntu 20.04.2 LTS environment. For more details, we refer to our publicly available code base. BloodMNIST contains images from different normal cells belonging to eight classes, and DermaMNIST consists of dermatoscopic images categorizing seven different diseases [3]. We rescale images from the medical datasets from 28x28 to 32x32 with nearest-neighbor interpolation. More details can be found in the publicly available code base.

C All Learning Curves

In the following, we report all learning curves for all tested datasets and settings, including grayscale data with varying query sizes (see Figure 2), RGB data with varying model architecture (see Figure 3), redundant data (see Figure 5), and tabular data (see Figure 6).

References

1. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
2. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. JMLR **12**, 2825–2830 (2011)
3. Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., Ni, B.: Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. Scientific Data **10**(1), 41 (2023)



(a) MNIST, Q: 200

(b) MNIST, Q: 500

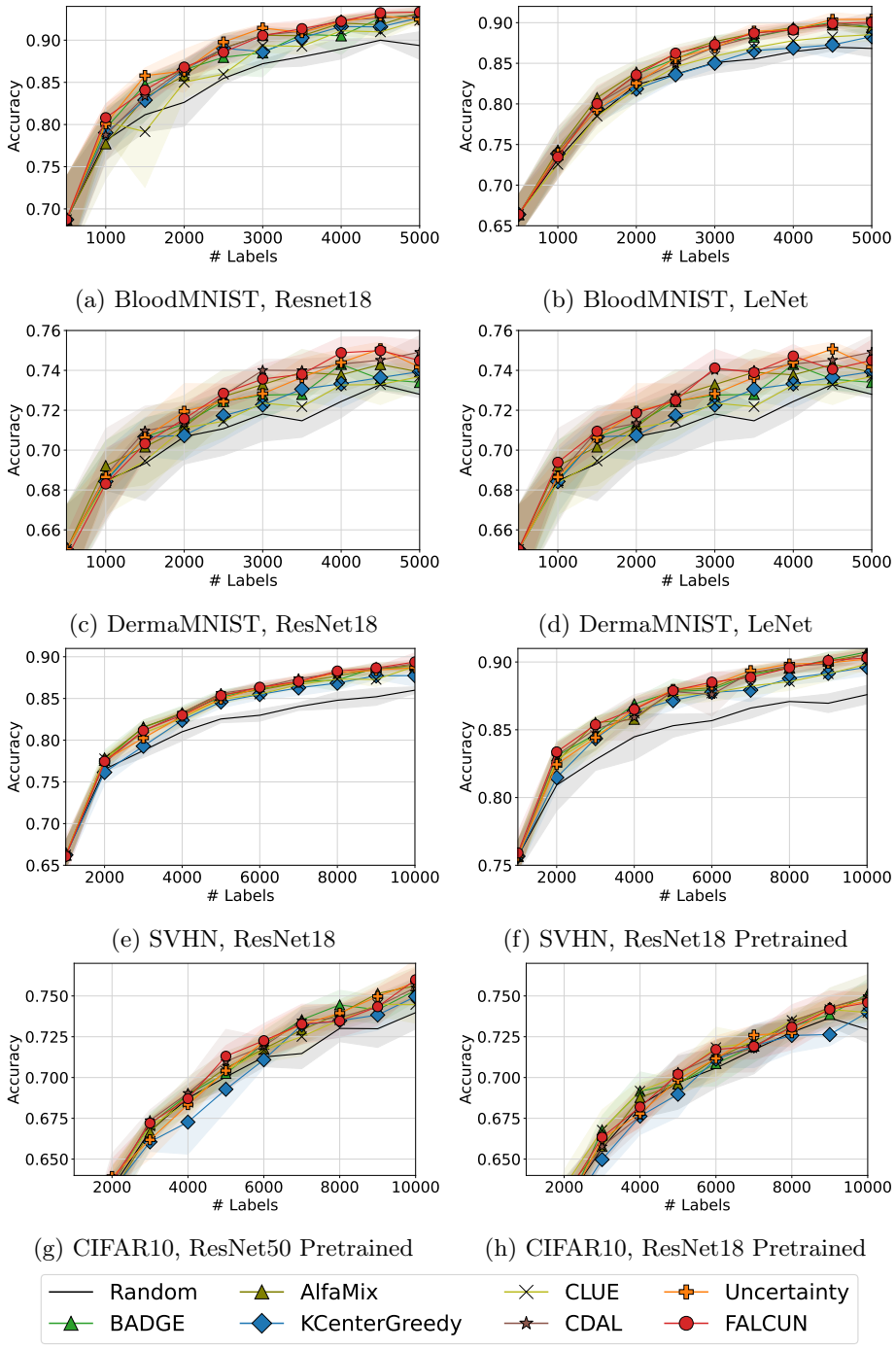
(c) EMNIST, Q: 200

(d) EMNIST, Q: 500

(e) FashionMNIST, Q: 200

(f) FashionMNIST, Q: 500

Fig. 2: AL Curves grayscale images.



(a) BloodMNIST, Resnet18

(b) BloodMNIST, LeNet

(c) DermaMNIST, ResNet18

(d) DermaMNIST, LeNet

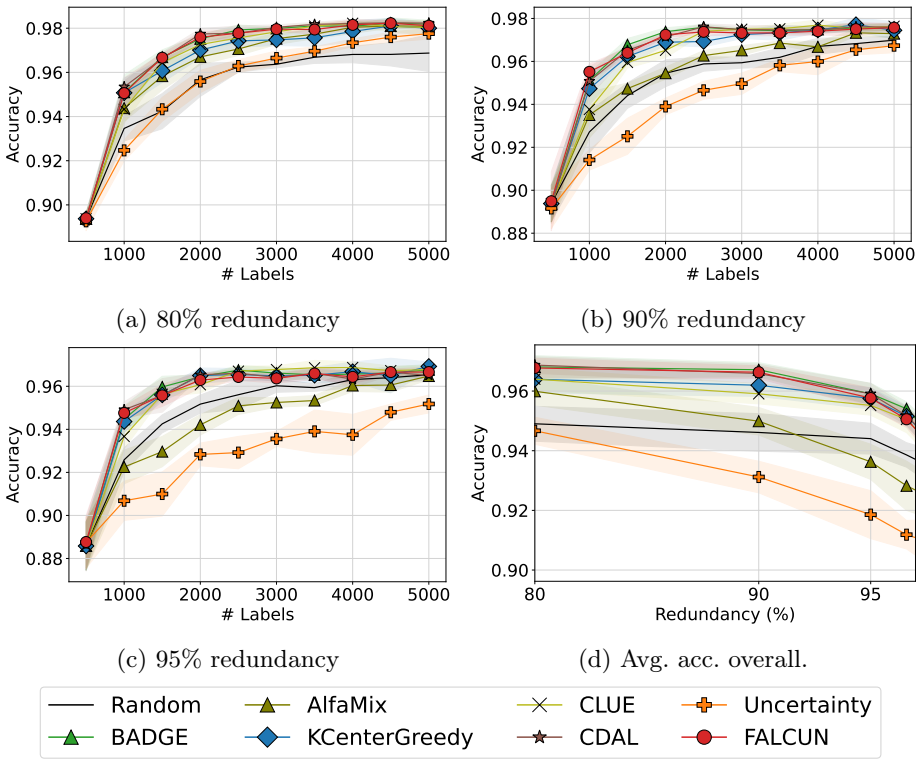
(e) SVHN, ResNet18

(f) SVHN, ResNet18 Pretrained

(g) CIFAR10, ResNet50 Pretrained

(h) CIFAR10, ResNet18 Pretrained

Fig. 3: AL Curves colored image datasets.



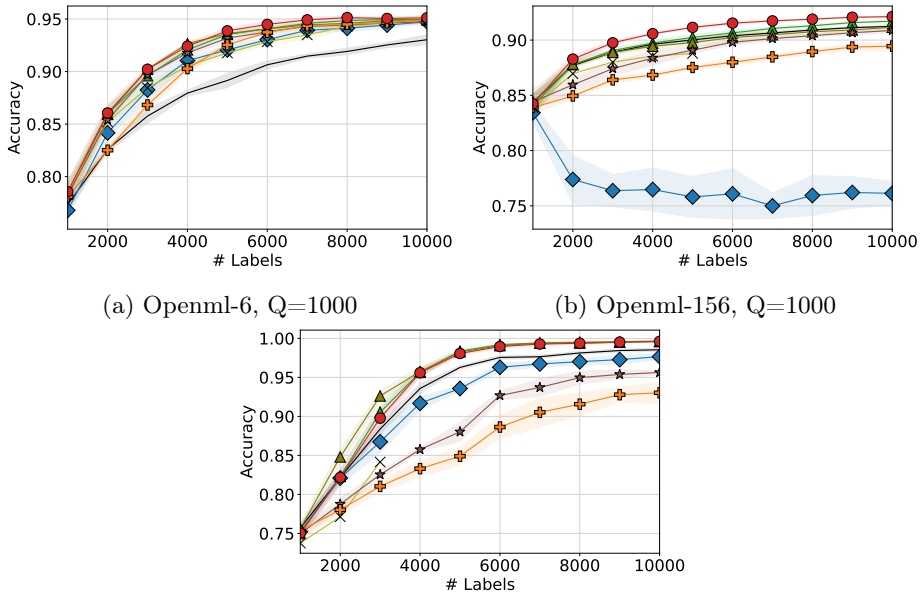
(a) 80% redundancy

(b) 90% redundancy

(c) 95% redundancy

(d) Avg. acc. overall.

Fig. 5: AL Curves on RMNIST.



(a) Openml-6, Q=1000

(b) Openml-156, Q=1000

(c) Openml-155, Q=1000

Fig. 6: AL Curves for tabular data.

C DiffusAL: Coupling Active Learning with Graph Diffusion for Label-Efficient Node Classification

Authors

Sandra Gilhuber*, Julian Busch*, Daniel Rotthues, Christian MM Frey, and Thomas Seidl
* These authors contributed equally to this work.

Venue

Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD), pages 75–91. Springer, 2023.

DOI

https://doi.org/10.1007/978-3-031-43412-9_5

Code

https://github.com/lmu-dbs/diffusal

Declaration of Authorships

Sandra Gilhuber and Julian Busch proposed the research idea and developed and conceptualized it with Daniel Rotthues. Daniel Rotthues implemented the first approach for his master's thesis under the supervision of Sandra Gilhuber and Julian Busch. Sandra Gilhuber improved the approach and implementation for the final publication and conducted the experiments. Sandra Gilhuber and Julian Busch analyzed their results. Sandra Gilhuber, Julian Busch, and Christian Frey wrote and revised the manuscript and Sandra Gilhuber discussed it with Thomas Seidl.

Copyright Notice

DiffusAL: Coupling Active Learning with Graph Diffusion for Label-Efficient Node Classification

Sandra Gilhuber[1,2( )], Julian Busch[1,3], Daniel Rotthues[1], Christian M. M. Frey[4], and Thomas Seidl[1,2,4]

[1] LMU Munich, Munich, Germany
{gilhuber,seidl}@dbs.ifi.lmu.de
[2] Munich Center for Machine Learning (MCML), Munich, Germany
[3] Siemens Technology, Princeton, NJ, USA
busch.julian@siemens.com
[4] Fraunhofer IIS, Erlangen, Germany
christian.maximilian.michael.frey@iis.fraunhofer.de

Abstract. Node classification is one of the core tasks on attributed graphs, but successful graph learning solutions require sufficiently labeled data. To keep annotation costs low, active graph learning focuses on selecting the most qualitative subset of nodes that maximizes label efficiency. However, deciding which heuristic is best suited for an unlabeled graph to increase label efficiency is a persistent challenge. Existing solutions either neglect aligning the learned model and the sampling method or focus only on limited selection aspects. They are thus sometimes worse or only equally good as random sampling. In this work, we introduce a novel active graph learning approach called *DiffusAL*, showing significant robustness in diverse settings. Toward better transferability between different graph structures, we combine three independent scoring functions to identify the most informative node samples for labeling in a parameter-free way: i) *Model Uncertainty*, ii) *Diversity Component*, and iii) *Node Importance* computed via graph diffusion heuristics. Most of our calculations for acquisition and training can be pre-processed, making DiffusAL more efficient compared to approaches combining diverse selection criteria and similarly fast as simpler heuristics. Our experiments on various benchmark datasets show that, unlike previous methods, our approach significantly outperforms random selection in 100% of all datasets and labeling budgets tested.

Keywords: active learning · node classification · graph neural networks

1 Introduction

Graph representation learning [17] and, especially, *Graph Neural Networks* (GNNs) [2,5,16] have been adopted as a primary approach for solving machine

S. Gilhuber and J. Busch—Equal contribution

D. Koutra et al. (Eds.): ECML PKDD 2023, LNAI 14169, pp. 75–91, 2023.
https://doi.org/10.1007/978-3-031-43412-9_5

learning tasks on graph-structured data, including node classification [18], graph classification [21], and link prediction [41]. Applications range from quantum chemistry [16] over traffic forecasting [44] to cyber-security [6].

However, supervised GNN models require sufficient training labels and usually assume that such labels are freely available. But, in reality, while unlabeled data is usually abundant, it is laborious and costly to provide annotations. Graph active learning has emerged as a promising direction to reduce labeling costs by carefully deciding which data should be labeled to increase label efficiency. Under a limited budget, e.g., a fixed number of data samples to be labeled or time spent labeling by a domain expert, active learning aims to annotate an optimized set of training data iteratively. Hence, a key aspect of graph active learning is identifying the most informative instances in the abundance of unlabeled data for labeling. In particular, the goal is to be consistently more label-efficient than random labeling. Since random sampling is arguably the fastest and least complex method, active learning methods that are not significantly better than random sampling are not worthwhile.

However, since graphs can vary widely, it is very difficult to design an approach significantly better than random sampling across different labeling budgets and graph structures. Existing graph-active learning approaches reach their limits for various reasons: Some approaches focus only on limited selection aspects [23,28] and outperform random selection only on certain graphs. Others focus on one-shot selection without iterative re-training and active selection and can therefore not exploit model-related uncertainty scores [37,43]. Other methods try to include various criteria in the selection but are sensitive to user-defined hyper-parameters or are not deliberately aligned with the used model architecture [8,15]. Moreover, many methods use a GCN [18] for training and acquisition. However, GCNs learn latent node features and perform neighborhood aggregation in a coupled fashion, which can negatively influence the time needed for the active learning procedure. In contrast, *Graph diffusion* is a promising direction tackling limitations such as restriction to k-hop neighborhoods [7] or over-smoothing, where neighborhood aggregation and learning are decoupled.

In this work, we use diffusion-based heuristics to combine graph learning with active learning. In particular, we propose *DiffusAL*, a novel graph active learning method that leverages graph diffusion for highly accurate node classification and efficiently re-uses the computed diffusion matrix and diffused node feature vectors in the learning procedure.

We introduce a new scoring function for identifying a node's utility which consists of three factors: i) *Model Uncertainty*, ii) *Diversity Component*, and iii) *Node Importance*. DiffusAL combines these scores in a parameter-free scoring function that naturally adapts to consecutively learning iterations.

Specifically, for i) *Model Uncertainty*, we exploit a state-of-the-art scoring that has shown an improving impact on the selection of nodes [32]. Next, the ii) *Diversity Component* refers to the variability of node features and, therefore, their respective labels. For that, we apply a clustering method on the precomputed diffusion matrix where diversity is reached by picking samples from underrepresented communities. Finally, for computing iii) *Node Importance*, we exploit the information given by diffusion matrix based on the Personalized

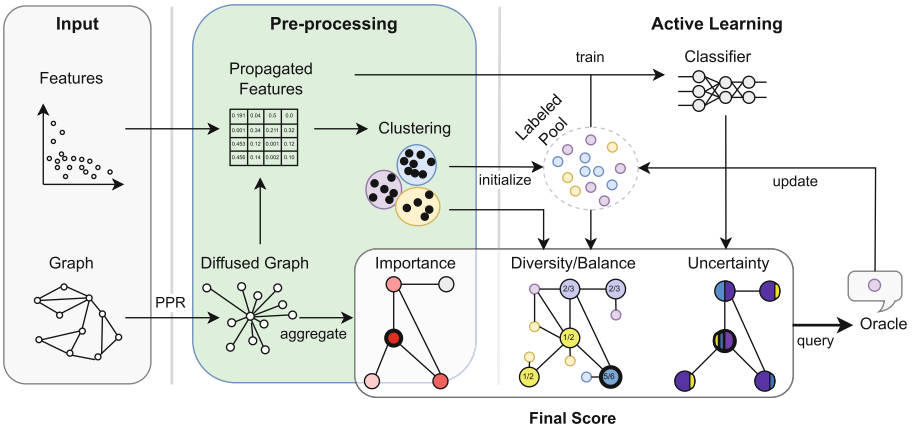


Fig. 1. DiffusAL pipeline consisting of the original input graph and corresponding node features (grey box), pre-computed static model-independent scores, such as the propagated feature matrix and derived node importance (green box), a dynamic, model-independent score based on the composition of the labeled pool (Diversity/Balance), as well as a dynamic, model-dependent informativeness score (Uncertainty). These scores are combined into a final node rating (white box) to select the most useful instances for annotation. (Color figure online)

PageRank (PPR), which provides information about the relative importance of nodes in a graph w.r.t. a particular seed node. The high-level key concepts of DiffusAL are illustrated in Fig. 1.

We evaluate DiffusAL on five real-world benchmark datasets, demonstrating its superiority over a variety of competitors. Notably, DiffusAL is the only competitor to outperform random selection with statistical significance in 100% of the evaluated datasets and labeling budgets. In a series of ablation studies, we show that DiffusAL works consistently well on all benchmark datasets, analyze which components contribute to its performance, and investigate its efficiency.

In summary, our contributions are as follows:

- Enhancing the selection of influential nodes by using *diffusion-based node importance* and utilizing pre-computed *clustering on diffused features* to prevent oversampling a particular region.
- Combining three complementary node scoring components in a parameter-free way.
- Achieving high efficiency by propagating statically pre-computed features stored in a diffusion matrix.

2 Related Work

Early works on graph active learning [3,24] exploit the graph structure for selecting nodes for querying without graph representation learning. More recent

approaches [8,15,23,28,38] use GCNs to exploit the graph structure as well as learned features. FeatProp [38] leverages node feature propagation followed by K-Medoids clustering for the selection of instances. By defining the pairwise node distances between the corresponding propagated node features, the model selects nodes being closest to the cluster representatives yielding a diverse set over the input space. However, the diversity scoring function in our model puts more weight on underrepresented clusters yielding a more balanced view of the available data space and, therefore, is more suitable for imbalanced data. In [43], the authors proposed *GRAIN*, a model inspecting social influence maximization for data selection. Their objective is a diversified influence maximization by exploiting novel influence and diversity functions. In contrast to their work, we focus on an iterative active learning setting [10] since it directly enables exploiting the uncertainty scores entangled to a model which is known to be valuable for query selection. The most related work to our approach is presented in [8] where the authors propose *Active Graph Embedding* (AGE) using as selection heuristic a weighted sum of information entropy, information density, and graph centrality defined on direct neighborhoods. For the latter, they propose to use PageRank centrality. The time-sensitive coefficients of the weighted sum are chosen from a beta distribution using the number of training iterations as input. We overcome these limitations related the restriction on direct neighborhoods aggregations used in standard GNNs [2,5,16] by leveraging continuous relationships via graph diffusion [7,20]. In [15], *ANRMAB* is proposed. It uses a multi-armed bandit mechanism for adaptive decision-making by assigning different weights to different criteria when constructing the score to select the most informative nodes for labeling. The model *LSCALE* [23] exploits clustering-based (K-Medoids) active learning on a designed latent space leveraging two properties: low label requirements and informative distances. For the latter, the authors integrate *Deep Graph Infomax* [36] as an unsupervised model. Therefore, in contrast to our approach, the model utilizes a purely distance-based selection heuristic. The method *GEEM* [29] maximizes the expected error reduction to select informative nodes to label.

To the best of our knowledge, we are the first to leverage the power of diffusion-based heuristics for the computation of node importance, being an integral part of our scoring function, combining three complementary components to compute the nodes yielding the highest utility scores. Moreover, our novel scoring function uncouples from any parameter presets, being a critical choice without any a priori knowledge about the input data.

3 DiffusAL

3.1 Preliminaries

Notation. We consider the problem of active learning for node classification. We are given a graph $G = (V, E)$ represented by an adjacency matrix $A \in \{0,1\}^{n \times n}$ along with a node feature matrix $X \in \mathbb{R}^{n \times d}$. Each node $v \in V$ belongs to exactly one class $c_v \in \{1, \ldots, C\}$, where C is the number of classes present

in the dataset. A budget constraint B denotes the maximum number of nodes for which the active learning algorithm may request the correct labels from the oracle. The main goal is to select a subset of nodes $S \subset V$ such that $|S| = B$ and the accuracy of a classification model trained on these nodes is maximized. In a batch setting, b denotes the number of nodes selected within each acquisition round.

Recap: Feature Diffusion. In contrast to conventional GNN architectures [18,35,39] that learn latent node features and perform neighborhood aggregation in a coupled fashion, *graph diffusion* effectively decouples the two steps to address certain shortcomings of conventional GNN architectures, including the restriction to k-hop neighborhoods [7] and issues related to over-smoothing[14,22,26,40]. The general effectiveness of diffusion, when paired with conventional GNN architectures, was shown in [20]. In general, a parametric diffusion matrix can be defined as

$$P = \sum_{k=0}^{\infty} \theta_k T^k, \tag{1}$$

where T is a transition matrix and θ are weighting parameters. A popular choice is *Personalized PageRank (PPR)* [4,7,11,12,19], where $T = AD^{-1}$ is the random walk matrix, D is the diagonal degree matrix, and $\theta_k = \alpha(1-\alpha)^k$. Intuitively, P_{ij} corresponds to the probability that a random walk starting at node i will stop at node j and can be interpreted as the importance of node j for node i. The restart probability $\alpha \in [0, 1]$ controls the effective size of a node's PPR-neighborhood. An approximation of the PPR matrix can be pre-computed in time $O(n)$ using push-based algorithms [7]. This approximation requires a second hyper-parameter $\varepsilon > 0$ that thresholds small entries and, thus, has a sparsification and noise reduction effect. Once computed, the PPR matrix can replace the adjacency matrix used by conventional message-passing networks for neighborhood aggregation [7,19].

3.2 Model Architecture

For DiffusAL, we propagate the original node features such that the propagated node features don't depend on any learned transformations and can be pre-computed as well. We propose a query-by-committee (QBC) approach [33], where the propagated node features are connected to an ensemble of MLP classifiers to robustify uncertainty estimation during the sample selection process compared to a commonly used single MLP. Additionally, features are diffused over multiple scales by varying the hyper-parameter α controlling the effective neighborhood size over which features are aggregated. In particular, the model predictions are given as

$$Y = predict\left(\sum_{j\in\{1,\ldots,M\}} transform_j\left(\sum_{i\in\{1,\ldots,K\}} P^{(\alpha_i)}X\right)\right), \tag{2}$$

where K denotes the number of scales, and M denotes the number of MLPs in the classification ensemble. The pre-computed diffused features are aggregated over multiple scales using the *sum* function and fed to the hidden layer of each MLP. The learned representations are then aggregated using the *sum* function and passed to the shared prediction layer. All ensemble members share the same architecture and only differ in the random initialization of their weights and biases. The QBC can be trained very efficiently with gradient descent, and, in particular, the expensive diffusion step needs to be performed only once as a pre-processing step.

3.3 Node Ranking and Selection

In addition to facilitating highly effective and efficient prediction, the previously computed diffusion matrix $P = \sum_{i \in \{1,\dots,K\}} P^{(\alpha_i)}$ and diffused features PX are reused to calculate expressive ranking scores for active node selection.

Model Uncertainty. For measuring model uncertainty, we utilize the QBC defined above. In particular, we compute the Shannon entropy over the softmaxed output distribution to determine the **uncertainty score** for node i:

$$s_{\text{unc}}(i) = - \sum_{j \in \{1,\dots,C\}} y_{ij} \log y_{ij}. \tag{3}$$

The scores are L1-normalized over all unlabeled nodes to $[0, 1]$, so all scoring functions share the same scale and can be sensibly combined.

While this score is inspired by the classical query-by-committee [33] approach, it differs in the sense that it doesn't average the softmax outputs of the individual committee members but rather considers the softmax output of a single shared prediction layer applied to aggregated latent representations. Thereby, differing predictions become more distinct in the softmax output.

Diversity Component. For the diversity component, we perform k-Means clustering on the *diffused features* with $k = b$ and assign each node a pseudo-label based on the clustering result. Note that we pre-compute these cluster assignments such that no re-computations are necessary at query time, in contrast to other approaches (e.g., based on GCNs), where updated node features would change the clustering.

The cluster-based pseudo labels are used to ensure decent coverage of the feature space. At each iteration, each node i receives a **diversity score**

$$s_{\text{div}}(i) = 1 - \frac{|c_{train}|}{|V_{train}|}, \tag{4}$$

where $c \in C$ denotes the cluster node i was assigned to, $|c_{train}|$ denotes the number of nodes in the currently labeled training set belonging to cluster c, and $|V_{train}|$ is the number of currently labeled training nodes. In short, each node in

the unlabeled pool is weighted by the relative size of its cluster in the training set, such that nodes from currently underrepresented clusters receive a higher score. In contrast to only focusing on avoiding redundancy in the current batch [1], our diversity score can also be interpreted as a balancing score ensuring that no region is over-sampled within the labeled pool.

Some existing works on graph active learning [8,15] ignore the limitations of a randomly initialized labeled pool and ensure class balance. However, this simplification is rather unrealistic in a real-world active learning setting. To overcome this limitation, we again exploit the k-Means clustering used for the diversity score and select nodes closest to centroids for the initial pool, inspired by clustering-based sampling approaches [23,37] and existing work on initial pool selection [9].

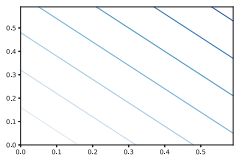
Node Importance. Graph diffusion allows for a natural way to quantify node importance. Since the weights P_{ij} used for neighborhood aggregation can be interpreted as importance scores, summing up the importance of a node i for all other nodes j yields a measure of the general importance of node i, measuring its total influence on the predictions for other nodes. Since the columns of S are stochastic, this procedure yields consistently scaled overall importance scores. In particular, the **importance score** of node i is given by the row-wise sum

$$s_{\text{imp}}(i) = \sum_{j \in V} P_{ij}. \tag{5}$$

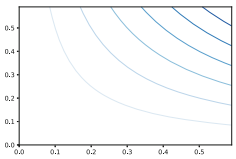
Since the importance scores for all nodes can be computed directly from the PPR matrix, they can be pre-computed before the active learning cycle starts. Our node importance score is a proxy for how much influence a node has on other nodes, where nodes with higher scores are assumed to carry more valuable information about many other nodes as well. Node importance could be interpreted as a novel representativeness measure, which has been quantified via density- or center-based selection within previous (graph) active learning approaches [8,15,42]. However, we do not need to recompute a clustering on learned representations after each selected sample, nor do we require good representations since we can extract the information directly from the graph topology. Further, our importance score of a node directly reflects the influence of that node on the model's predictions, since the weights from which we compute the scores are directly used for neighborhood aggregation. This is not the case for alternative existing measures.

Score Combination and Node Selection. In summary, the uncertainty score assigns higher weights to nodes about which the committee is most uncertain, the diversity score assigns higher weights to nodes belonging to underrepresented clusters, and the node importance score assigns higher weights to nodes with a higher influence on the predictions for other nodes. The individual scores for a node are combined in a multiplicative fashion to determine the node's utility:

$$s(i) = s_{\text{unc}}(i) \cdot s_{\text{div}}(i) \cdot s_{\text{imp}}(i). \tag{6}$$



(a) **Sum aggregation**: Isolines are straight due to fixed weighting.



(b) **Multiplicative aggregation**: Isolines are curved, favoring similar values over diverging ones.

Fig. 2. Score aggregation: for two arbitrary scores on the x and y axes (e.g. uncertainty and representativeness), the corresponding aggregated score is depicted as an isoline, i.e., each point on the line corresponds to the same final value.

As illustrated in Fig. 2, the intuition behind the multiplicative combination is to slightly favor nodes displaying a well-rounded distribution of scores over those with a strong imbalance when the sum of the scores is identical while still allowing extraordinarily important or uncertain nodes to be selected. Existing works use slightly different variations of time-sensitive weighted sums, thereby gradually shifting the focus from representativeness to uncertainty [8,42]. A disadvantage of time-sensitive weighting is that the performance of the selection algorithm depends on the choice of good hyper-parameters, which is difficult in a real-world active learning setting. In contrast, our multiplicative approach is parameter-free and naturally time-sensitive. In the early stages of training, the classifiers essentially guess predictions more or less uniformly, leading to roughly similar uncertainty scores for most nodes. Consequently, the uncertainty score is close to a constant factor applied equally to all nodes, thus naturally making the model-free scores the deciding ones in the final score. However, uncertainty scores become increasingly important once the classifiers become more confident in their predictions. The combined utility score is determined for each unlabeled node in each active learning cycle. Afterward, the unlabeled nodes are ranked according to their utility, and the nodes with the highest utility scores are labeled.

4 Experiments

To demonstrate the effectiveness and efficiency of DiffusAL, we conduct a series of experiments. In particular, we investigate three research questions:

R1 - How does DiffusAL perform compared to state-of-the-art methods?
R2 - How does each of DiffusAL's components contribute?
R3 - How is the training and acquisition efficiency?

4.1 Experimental Setup

Datasets. We evaluate DiffusAL on several well-established benchmark datasets for node classification, namely the citation networks Citeseer [30], Cora [30] and

Table 1. Dataset statistics (only considering the largest connected component).

Dataset	#Nodes	#Edges	#Features	#Classes
Citeseer	2120	3679	3703	6
Cora	2485	5069	1433	5
Pubmed	19717	44324	500	3
Co-author CS	18333	81894	6805	15
Co-author Physics	34493	247962	8415	5

Pubmed [25], as well as the co-author networks Computer Science (CS) [34] and Physics [34], summarized in Table 1. For each dataset, only the largest connected component is used, and features are L1-normalized.

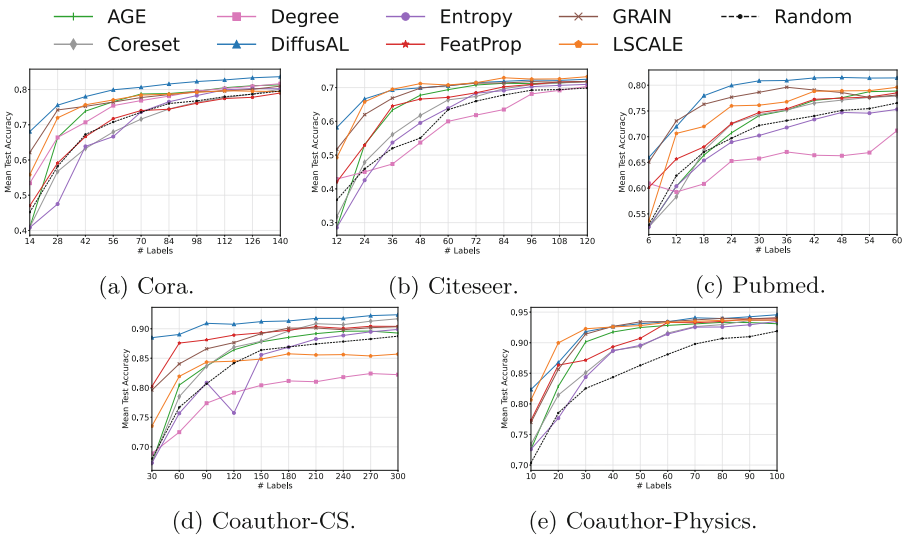
Implementation Details. All experiments were implemented using PyTorch [27] and PyTorch Geometric [13] and run on a single Nvidia Quadro RTX 8000 GPU. For more details, we refer to our publicly available codebase[1].

Competitors. We compare DiffusAL with *random* sampling, *entropy* sampling [32], and *coreset* [31] as graph-independent uncertainty-aware and diversity-aware active learning strategies, respectively. Furthermore, we include *degree* sampling as a graph-based representativeness-based baseline, selecting the highest degree nodes, as well as the state-of-the-art graph-specific active learning methods *AGE* [8], *FeatProp* [37], *LSCALE* [23] and *GRAIN* [43].

As proposed in [8,23,37,43], all baselines use GCNs as classifiers, except LSCALE, which uses the proposed distance-based classifier. Our proposed method DiffusAL uses the introduced QBC as a classifier, and we provide comprehensive experiments showing the influence of the prediction model.

Hyperparameters. We use the same hyper-parameters having a hidden layer size of 16, a dropout rate of 0.5, a learning rate of 0.01, and L2-regularization of 5×10^{-4} as proposed in [37]. For DiffusAL, we select α and ϵ as suggested in [7]. We follow a batch selection and retrain from scratch after each acquisition round. However, to ensure more diverse uncertainties (and because the other two scores are static), we follow the setting of [8] and also incrementally train the model for one epoch between instance selection within one acquisition round. The evaluation in Sect. 4.4 shows that this does not impair our efficiency. To provide a meaningful evaluation without the effects of an under-trained model or randomness factors, we report test accuracy for all approaches using a validation set of size 500 and early stopping. However, the validation set is only part of the evaluation, not the procedure itself. We set the size of the initial pool to 2C (cf. 3.1) and report results up to a budget of 20C with step sizes also twice the number of classes. To simulate a fairly realistic active learning scenario, the initial pool is sampled randomly without guaranteeing class balance for the baseline approaches without a specific initialization method. All experiments report an average of ten random seeds.

[1] https://github.com/lmu-dbs/diffusal.



(a) Cora. (b) Citeseer. (c) Pubmed.

(d) Coauthor-CS. (e) Coauthor-Physics.

Fig. 3. Active learning curves with the number of labeled nodes on the x-axis and average accuracy (over 10 random seeds) on the y-axis.

4.2 R1 - Performance Comparison

Figure 3 depicts the active learning curves for all budgets and datasets. DiffusAL (blue) is among the best-performing methods on all datasets. Especially on Cora and Coauthor-CS, we reach the highest mean accuracy for all labeling budgets and are the only competitor to reach a final accuracy of 83.6% and 92.4%, respectively. On Pubmed, GRAIN is similarly strong for the first two iterations. However, afterward, DiffusAL outperforms all methods for the remaining budgets and reaches a final average accuracy of 81.4%. In comparison, LSCALE, the second-best performing method with respect to the final budget, only reaches 79.9%.

On Citeseer and Physics[2], GRAIN and LSCALE are similarly strong as DiffusAL. For both datasets, the learning curves converge to similar accuracies above a certain labeling budget for some methods such that a clear winner can no longer be pronounced. Therefore, Fig. 4 provides a comprehensive dueling matrix indicating how often each strategy has won and lost against the other strategy in a similar fashion as was proposed in [1]. We apply a two-sided t-test with a p-value of 0.05 to the classification accuracies over 10 random seeds to count whether one method outperformed another with statistical significance.

[2] On Physics, Degree underperformed considerably and is therefore omitted for better presentation.

In total, we evaluated 50 experimental settings for each strategy (5 different datasets, 10 different labeling budgets from 2C to 20C). The values in a column and row of a method denote the percentage of losses and wins against another method, respectively. The bottom row indicates the average losses of each strategy over all experiments, and the right-most column indicates the average wins of a strategy over all experiments. The losses and wins in the cells c_{ij} and c_{ji} do not necessarily add up to 100%. The margin between the wins in cell ij and the losses in cell ji indicates how often the strategy i has performed equally well as competitor j. Both numbers, the average losses *and* the average wins, are particularly interesting when evaluating the success of an active learning method.

	DiffusAL	GRAIN	LSCALE	AGE	FeatProp	Coreset	Entropy	Random	Degree	Average Wins (%)
DiffusAL	**0**	**62**	**62**	**72**	**72**	**86**	**92**	**100**	**94**	71
GRAIN	**0**	0	26	36	42	58	72	90	80	45
LSCALE	**2**	4	0	28	34	54	66	72	78	38
AGE	**0**	0	14	0	20	34	48	58	68	27
FeatProp	**0**	4	22	14	0	16	42	44	62	23
Coreset	**0**	4	14	8	2	0	28	28	58	16
Entropy	**0**	0	8	0	6	4	0	18	46	9.1
Random	**0**	0	8	0	0	4	0	0	52	7.1
Degree	**0**	0	4	6	12	22	16	18	0	8.7
Average Losses (%)	0.2	8.2	18	18	21	31	40	48	60	

Fig. 4. Pairwise dueling matrix. Cell ij indicates how often competitor i won against competitor j **with statistical significance** over all datasets and labeling budgets (in %). The bottom-most row and right-most column denote each method's average losses and wins, respectively (in %).

In summary, the dueling matrix reveals the following insights:

- DiffusAL has the **fewest losses (0.2%, see first column)** and the **most wins (71%, see first row)**.
- DiffusAL **wins over random sampling most often (100%)**.
- Concerning wins over random sampling, GRAIN is the second-best method (90%). However, DiffusAL statistically never loses against GRAIN.
- The only strategy that can outperform DiffusAL is LSCALE. However, we beat LSCALE in 62% of experiments and lost only 2% of experiments.

4.3 R2 - Analysis of Contributing Factors

The selected datasets vary widely in terms of the number of nodes, edges, features, classes, and class distribution, making it difficult to develop an approach that can perform well across the spectrum. In the following, we analyze which components contribute most to DiffusAL's success and why it is so strong over a broad range of datasets. Table 2 shows the performance of DiffusAL (bottom row) and DiffusAL when switching off individual parts of the acquisition function, i.e., the diversity component (D), the uncertainty score (U) and the importance score (I) and exchanging the model architecture (middle rows) for 2C, 6C, and 12C labeling budgets on all datasets where C is the number of classes. Red, bold numbers indicate the smallest accuracy, indicating the largest influence of a switched-off component, and blue, bold numbers indicate the highest accuracy.

We exchange the classifier with a single network variant (MLP) and with a GCN taking the raw features as input instead of diffused features (GCN). Furthermore, we report results when using an additive score instead of a multiplicative score.

Table 2. Comparison of DiffusAL with ablated variants. **Blue, bold** numbers indicate the **highest, i.e. best,** accuracy. **Red, bold** numbers indicate the **lowest, i.e. worst,** accuracy and hence the component with largest influence.

			Cora			Citeseer			Pubmed			CS			Physics		
D	U	I	2C	6C	12C	2C	6C	12C	2C	6C	12C	2C	6C	12C	2C	6C	12C
		✓	45.5	77.8	80.6	**43.3**	63.8	69.7	56.3	68.0	69.8	71.9	81.7	82.2	71.9	89.8	93.8
	✓		45.5	76.1	80.1	43.3	65.5	70.0	56.3	70.6	75.4	71.9	83.3	90.8	71.9	89.6	93.1
	✓	✓	45.5	78.5	81.7	43.3	69.8	71.3	56.3	75.3	80.0	71.9	89.3	91.4	71.9	92.4	93.9
✓			-	74.5	76.0	-	67.6	71.1	-	64.6	76.5	-	89.4	90.4	-	86.4	87.1
✓		✓	-	76.4	80.5	-	67.7	71.0	-	77.2	79.9	-	87.5	87.3	-	91.5	92.4
✓	✓		-	78.6	81.9	-	69.1	71.0	-	74.9	77.1	-	90.5	91.6	-	88.3	90.9
Additive			-	78.8	81.3	-	**70.8**	71.3	-	79.1	80.2	-	**91.0**	**92.1**	-	91.7	92.7
MLP			62.0	78.8	81.8	52.7	70.6	**71.8**	64.1	78.8	79.9	87.8	90.4	91.2	80.4	91.4	93.6
GCN			61.8	77.5	80.7	49.8	69.3	71.3	64.5	76.5	78.5	83.3	89.6	91.2	**82.7**	91.6	93.1
DiffusAL			**68.0**	**79.9**	**82.3**	**58.2**	69.9	**71.8**	**65.9**	**80.0**	**81.4**	**88.5**	90.8	91.8	82.3	**92.6**	**94.1**

The importance, uncertainty, and additive scoring have no influence on the initial pool selection, so we leave out numbers there. Our QBC robustifies the accuracy, especially in the first iteration, compared to the other two variants (MLP, GCN). The performance difference between the models gets smaller with increasing label information. In particular, when label information is sparse, the committee stabilizes the prediction. However, the diversity component has the largest impact on the initial set for all datasets. When switching off diversity (first three rows), the accuracy drops between 9.6% (Pubmed) and 22.5% (Cora). Other approaches, such as FeatProp or LSCALE, also use clustering in the first iteration. However, our sampling directly operates on the diffused features, which subsequently directly influence the training and thus results in a very strong initial performance.

In general, switching off two scores yields worse results than only switching off one score, which indicates that the other two scores stabilize the results. But there is not one most important score over all datasets, supporting our claim that a robust selection benefits from diverse criteria. For instance, the accuracy drops the most when switching off *uncertainty* and *diversity* on Citeseer (by 2.1%) and especially on Coauthor-CS (by 9.6%). However, the performance on Cora and Physics primarily needs *uncertainty* and *importance*. In contrast, Pubmed benefits most from *diversity* and *importance*. Interestingly, some of our findings might give an indication of the performance of other methods. For instance, we found that importance, i.e., representativeness, is not beneficial on Coauthor-CS. LSCALE, which mainly focuses on representativeness sampling, yields the worst performance on this dataset. On Pubmed, however, uncertainty seems not to work well. Entropy and AGE both include uncertainty sampling and

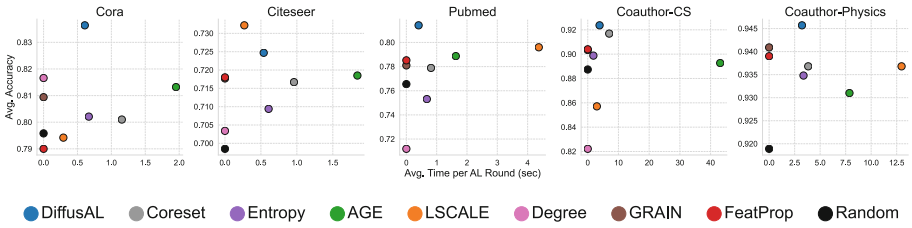


Fig. 5. Average time in seconds (x-axis) required for one active learning round compared to the average final accuracy (y-axis) for all methods (color). (Color figure online)

Table 3. Average time in seconds required for acquisition (acq), training (train), and in total ($\sum$) within one active learning iteration. Bold and underlined numbers indicate the fastest and second fastest methods, respectively. In total, DiffusAL is the fastest method on Physics and Pubmed, and the second fastest method on Cora and Citeseer.

	CS			Citeseer			Cora			Physics			Pubmed		
	acq	train	$\sum$	acq	train	$\sum$	acq	train	$\sum$	acq	train	$\sum$	acq	train	$\sum$
AGE	41.271	1.849	43.120	1.177	0.665	1.842	1.409	0.544	1.953	4.506	3.366	7.873	0.952	0.679	1.631
Coreset	5.191	1.797	6.988	0.344	0.615	0.960	0.537	0.616	1.154	0.572	3.258	3.830	0.138	0.675	0.813
Entropy	**0.005**	1.831	**1.836**	**0.002**	0.605	0.607	**0.002**	0.665	0.667	**0.011**	3.358	3.369	**0.002**	0.674	0.676
LSCALE	2.649	**0.317**	2.966	0.019	**0.249**	**0.269**	**0.042**	**0.250**	**0.292**	12.722	**0.258**	12.980	4.121	**0.247**	4.368
DiffusAL	2.567	1.282	3.849	0.183	0.356	0.539	0.268	0.339	0.608	0.357	2.863	**3.220**	0.043	0.361	**0.404**

yield worse results. On Cora, where uncertainty and representativeness seem effective, Coreset and FeatProp, which mainly focus on diversity, are among the worst-performing methods.

Using an *additive* score instead of a multiplicative score yields slightly worse results in general. From 10 comparisons, summing up the scores only yields three times slightly better results. However, the maximum difference is 0.9% (Citeseer 6C), whereas using the multiplicative in DiffusAL, the additive score is up to 1.4% (Physics 12C) better.

4.4 R3-Acquisition and Training Efficiency

Figure 5 shows the total average time (in seconds) for one active learning step on the x-axis (smaller is better) and the final accuracy after all 20C labels are selected on the y-axis (larger is better) for all methods (color).

We focus on an iterative AL selection where re-training between acquisition steps is necessary to get new uncertainty scores. In contrast, GRAIN, FeatProp, degree sampling, and random sampling select all instances for labeling at once and do not require re-training. Therefore, their average time is set to zero, and their accuracy is plotted for comparison. However, these methods are generally less label-efficient since they are not directly coupled to the current learning model. Except for Citeseer, DiffusAL is always on the Pareto-front, yielding the best final average accuracy while still being fairly time-efficient. In Table 3, we

split the total time into the acquisition and the training time for the iterative methods. All GCN-based methods (Coreset, AGE, Entropy) denote fairly similar training times. Despite using an ensemble, DiffusAL is slightly faster than the GCN-based methods since the features are pre-computed. AGE and Coreset both require a longer time for acquisition. AGE can exploit pre-calculated centrality scores. However, the uncertainty score and especially the density score must be freshly calculated in each round. Especially for the very large graph data CS, AGE requires over 40 s for one active learning iteration. Coreset extracts the latent representations from the current model and requires the computation of a pairwise distance matrix. Compared to that, DiffusAL only needs to calculate the uncertainty scores derived from the QBC model since the other scores are pre-computed. Only the entropy-based selection scheme has a faster acquisition time since it only needs one forward pass through the network.

LSCALE, which also defined a dedicated network towards a unified learning and selection framework, has the fastest training times out of all methods. However, depending on the dataset, the acquisition time is much larger than DiffusAL's acquisition time. As such, the overall time needed for one active learning round varies considerably between datasets. For instance, on Citeseer and Cora, LSCALE is the fastest method out of all iterative methods. Still, on the much larger graphs Pubmed and Physics, it is the slowest method due to larger acquisition times (4.4 s and 12.7 s, respectively). Overall, even though we use an ensemble method, our training and acquisition times are fairly stable across datasets and, in total, comparably good as plain uncertainty sampling with a GCN.

5 Conclusion

The annotation of unlabeled nodes in graphs is a time-consuming and costly task and, accordingly, it is of great interest to advance label-efficient methods. Motivated by the success of diffusion-based graph learning approaches, we propose DiffusAL, a novel active learning strategy for node classification. DiffusAL uses diffusion to predict node labels accurately and compute meaningful utility scores consisting of *model uncertainty*, *diffused feature diversity*, and *node importance* for active node selection, such that training and data selection cooperate toward label-efficient node classification. DiffusAL is significantly better generalizable over a wide range of datasets and is, in terms of statistical significance, not beaten by any other method in 99.8% of all experiments. Moreover, it is the only method that significantly outperforms random selection in 100% of the evaluated settings. Due to pre-computed features stored in a diffusion matrix, our model can efficiently compute a node's utility for training and acquisition. Our extensive ablation study shows that each component of DiffusAL contributes to different datasets and active learning stages, making it robust in diverse graph settings.

References

1. Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds. In: ICLR (2020)
2. Battaglia, P.W., et al.: Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261 (2018)
3. Bilgic, M., Mihalkova, L., Getoor, L.: Active learning for networked data. In: Proceedings of the 27th International Conference on Machine Learning (ICML 2010), pp. 79–86 (2010)
4. Borutta, F., Busch, J., Faerman, E., Klink, A., Schubert, M.: Structural graph representations based on multiscale local network topologies. In: WI-IAT (2019)
5. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. IEEE SPM **34**(4), 18–42 (2017)
6. Busch, J., Kocheturov, A., Tresp, V., Seidl, T.: Nf-gnn: network flow graph neural networks for malware detection and classification. In: SSDBM (2021)
7. Busch, J., Pi, J., Seidl, T.: Pushnet: efficient and adaptive neural message passing. In: ECAI (2020)
8. Cai, H., Zheng, V.W., Chang, K.C.C.: Active learning for graph embedding. arXiv preprint arXiv:1705.05085 (2017)
9. Chandra, A.L., Desai, S.V., Devaguptapu, C., Balasubramanian, V.N.: On initial pools for deep active learning. In: NeurIPS 2020 Workshop on Pre-registration in Machine Learning, pp. 14–32. PMLR (2021)
10. Contardo, G., Denoyer, L., Artières, T.: A meta-learning approach to one-step active-learning. In: AutoML@PKDD/ECML (2017)
11. Faerman, E., Borutta, F., Busch, J., Schubert, M.: Semi-supervised learning on graphs based on local label distributions. In: MLG (2018)
12. Faerman, E., Borutta, F., Busch, J., Schubert, M.: Ada-lld: adaptive node similarity using multi-scale local label distributions. In: WI-IAT (2020)
13. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch Geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds (2019)
14. Frey, C.M.M., Ma, Y., Schubert, M.: Sea: graph shell attention in graph neural networks. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (2022)
15. Gao, L., Yang, H., Zhou, C., Wu, J., Pan, S., Hu, Y.: Active discriminative network representation learning. In: IJCAI (2018)
16. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: ICML, pp. 1263–1272. PMLR (2017)
17. Hamilton, W.L.: Graph representation learning. Synth. Lect. Artifi. Intell. Mach. Learn. **14**(3), 1–159 (2020)
18. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
19. Klicpera, J., Bojchevski, A., Günnemann, S.: Predict then propagate: graph neural networks meet personalized pagerank. In: ICLR (2019)
20. Klicpera, J., Weißenberger, S., Günnemann, S.: Diffusion improves graph learning. Adv. Neural. Inf. Process. Syst. **32**, 13354–13366 (2019)
21. Lee, J.B., Rossi, R., Kong, X.: Graph classification using structural attention. In: KDD, pp. 1666–1674 (2018)
22. Li, Q., Han, Z., Wu, X.M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: AAAI (2018)

23. Liu, J., Wang, Y., Hooi, B., Yang, R., Xiao, X.: Lscale: latent space clustering-based active learning for node classification. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, 19–23 September 2022, Proceedings, Part I, pp. 55–70. Springer (2023). https://doi.org/10.1007/978-3-031-26387-3_4
24. Moore, C., Yan, X., Zhu, Y., Rouquier, J.B., Lane, T.: Active learning for node classification in assortative and disassortative networks. In: Proceedings of the 17th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining, pp. 841–849 (2011)
25. Namata, G.M., London, B., Getoor, L., Huang, B.: Query-driven active surveying for collective classification. In: MLG (2012)
26. Ogawa, Y., Maekawa, S., Sasaki, Y., Fujiwara, Y., Onizuka, M.: Adaptive node embedding propagation for semi-supervised classification. In: Oliver, N., Pérez-Cruz, F., Kramer, S., Read, J., Lozano, J.A. (eds.) ECML PKDD 2021. LNCS (LNAI), vol. 12976, pp. 417–433. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86520-7_26
27. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019)
28. Regol, F., Pal, S., Zhang, Y., Coates, M.: Active learning on attributed graphs via graph cognizant logistic regression and preemptive query generation. In: ICML, pp. 8041–8050. PMLR (2020)
29. Regol, F., Pal, S., Zhang, Y., Coates, M.: Active learning on attributed graphs via graph cognizant logistic regression and preemptive query generation. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, JMLR.org (2020)
30. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. AI Mag. **29**(3), 93 (2008)
31. Sener, O., Savarese, S.: Active learning for convolutional neural networks: a core-set approach. In: ICLR (2018)
32. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin-Madison (2009)
33. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT 1992, pp. 287–294. Association for Computing Machinery, New York (1992)
34. Shchur, O., Mumme, M., Bojchevski, A., Günnemann, S.: Pitfalls of graph neural network evaluation. In: NeurIPS Relational Representation Learning Workshop (2018)
35. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: ICLR (2018)
36. Veličković, P., Fedus, W., Hamilton, W.L., Lió, P., Bengio, Y., Hjelm, R.D.: Deep graph infomax. In: ICLR (2018)
37. Wu, Y., Xu, Y., Singh, A., Yang, Y., Dubrawski, A.: Active learning for graph neural networks via node feature propagation. arXiv preprint arXiv:1910.07567 (2019)
38. Wu, Y., Xu, Y., Singh, A., Yang, Y., Dubrawski, A.: Active learning for graph neural networks via node feature propagation. CoRR abs/ arXiv: 1910.07567 (2019)
39. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks?. In: ICLR (2019)

40. Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.i., Jegelka, S.: Representation learning on graphs with jumping knowledge networks. In: ICML, pp. 5453–5462. PMLR (2018)
41. Zhang, M., Chen, Y.: Link prediction based on graph neural networks. In: Advances in Neural Information Processing Systems 31 (2018)
42. Zhang, W., Shen, Y., Li, Y., Chen, L., Yang, Z., Cui, B.: Alg: fast and accurate active learning framework for graph convolutional networks. In: SIGMOD, pp. 2366–2374 (2021)
43. Zhang, W., et al.: Grain: Improving data efficiency of graph neural networks via diversified influence maximization. Proc. VLDB Endow. **14**(11), 2473–2482 (2021)
44. Zhao, L., et al.: T-gcn: a temporal graph convolutional network for traffic prediction. IEEE Trans. Intell. Transp. Syst. **21**(9), 3848–3858 (2019)

Supplementary Materials to "DiffusAL: Coupling Active Learning with Graph Diffusion for Label-Efficient Node Classification"

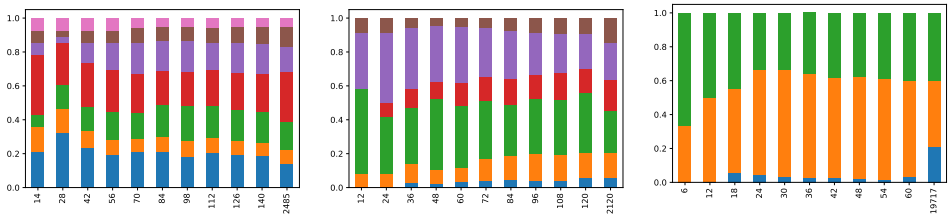
Sandra Gilhuber*[1,2], Julian Busch*[1,3], Daniel Rotthues[1], Christian M.M. Frey[4], and Thomas Seidl[1,2,4]

[1] LMU Munich, Germany {gilhuber,seidl}@dbs.ifi.lmu.de
[2] Munich Center for Machine Learning (MCML), Germany
[3] Siemens Technology, Princeton, NJ, USA busch.julian@siemens.com
[4] Fraunhofer IIS, Germany
christian.maximilian.michael.frey@iis.fraunhofer.de

A Diffusion and Node Importance

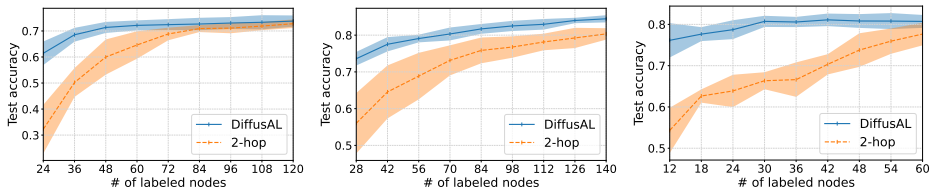
We provide further analysis on the node importance scores since they are a key property of DiffusAL.

Class Distribution of Important Nodes Figure 1 displays the class distribution of the top k most important nodes for the citation networks. The last bar indicates the original class distribution comprising all nodes. On Pubmed, the minority class is heavily underrepresented in the top 60 most important nodes, leading the node importance score in our active selection to favor samples from the other two majority classes. On Citeseer, a similar but weaker trend can be observed. In contrast, on Cora, the distribution of the top k most important nodes rapidly approximates the actual class distribution.



(a) Cora. (b) Citeseer. (c) Pubmed.

Fig. 1: Class distribution of the most important nodes: the x-axis represents various budgets of nodes, the y-axis measures the fraction that each class makes up in the given labeled set. Colors indicate the respective classes.



(a) Citeseer. (b) Cora. (c) Pubmed.

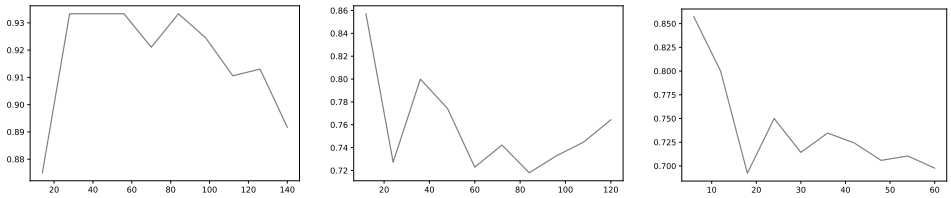
Fig. 2: DiffusAL (blue) compared to a purely *2*-hop-based alternative (orange).

Node Importance vs. Degree As already discussed, the intuitive interpretation of the importance score for a given node i is the probability of a random walk that starts at a random node j to end at node i. Consequently, a valid assumption is that important nodes tend to have an above-average degree since high-degree nodes are more likely to be visited during a random walk. Here, we analyze how the degree differs from node importance since common centrality measures, such as degree centrality, have already been established as active learning criteria in related work. Figure 3 displays the overlap of the most important nodes compared to the highest degree nodes for different budgets up to $20C$ for each dataset. Cora has the highest average overlap for the considered budgets with over 90%. However, for the other datasets, this overlap is a lot smaller at only around 75% on average, indicating that node degree, while still seemingly a large one, is not the only influencing factor determining the importance of a node. Furthermore, a general insight is that the overlap for the topmost important nodes is the largest and decreases afterward.

Node Importance and Diffusion vs. 2-hop We further analyze the influence of diffusion on sampling *and* graph learning and drop all diffusion-related content to demonstrate the advantages. We replace every aspect concerned with diffusion with comparable components entirely based on k-hop (2-hop) neighborhoods. Instead of using the PPR matrix to compute propagated features, the original adjacency matrix (squared and symmetrically normalized) is used. Furthermore, these features serve as the basis for the labeled pool initialization, cluster affiliation, and classifier training. Additionally, instead of the PPR matrix, we consider the column-wise sum of the adjacency matrix for the importance score. The results, depicted in Figure 2, reveal the advantages of diffusion. Test accuracy decreases for all three citation networks when using the 2-hop-based replacement scores while variability increases, as can be seen from the wider error bands.

B Further Explanation on DiffusAL vs. AGE

Among all competitors, AGE is arguably the most related one. However, there are some important differences regarding the concrete realization of the acquisition function. AGE uses a density score as well as a centrality score. Both



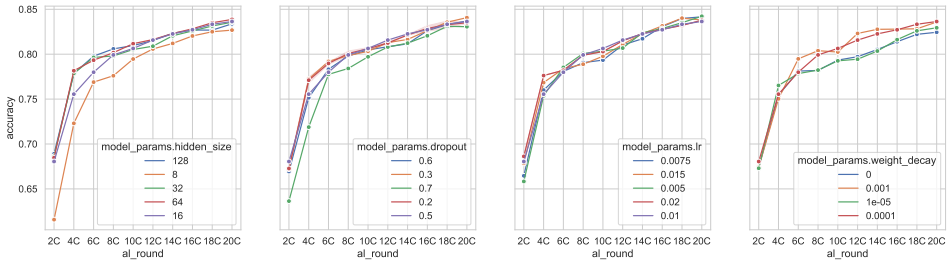
(a) Cora. (b) Citeseer. (c) Pubmed.

Fig. 3: Overlap of most important and highest degree nodes for a given budget - x: various sampling budgets, y: the fraction of nodes appearing in both respective sets of nodes.

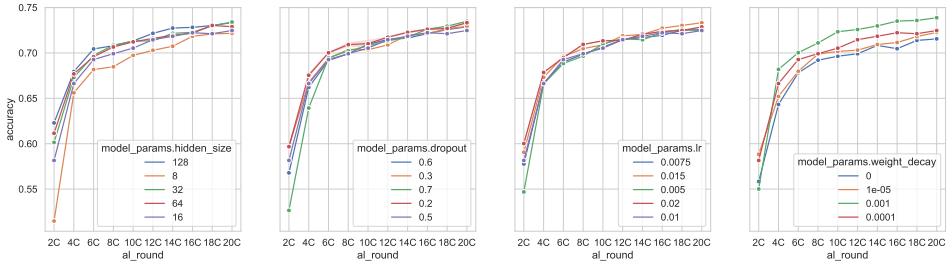
are common choices for representativeness. As such, there might be a strong tendency to favor very representative instances from the graph. In contrast, we use a diversity score and node importance as representativeness estimate. The only purpose of the diversity score is to ensure that no region is oversampled. However, the node importance ensures that influential points are selected. A key difference to the centrality score used in AGE is that node importance not only considers the local neighborhood but takes the whole graph structure into account. Furthermore, our selection and training both exploit expressive features in a consistent fashion. That is, the training directly makes use of the precomputed features. These features are also used for clustering, directly ensuring the diversity that is known to the model. The node importance directly corresponds to nodes that are most influential. Lastly, DiffusAL does not use any time-sensitive weighting parameters.

C Hyperparameter Variations

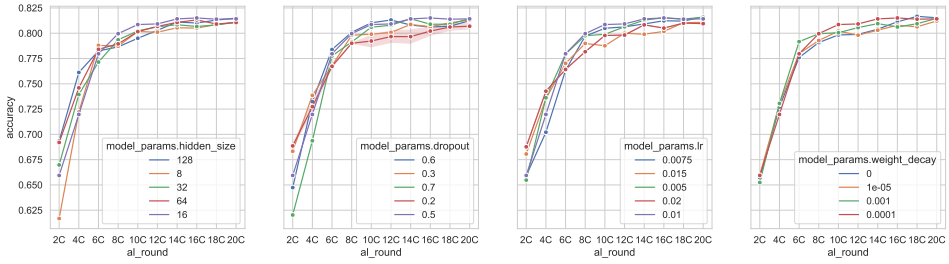
We further have conducted experiments to show the robustness of DiffusAL regarding the hyper-parameters for model training. For all datasets, DiffusAL denotes quite stable learning curves. Only on CS, hidden size, dropout, and weight decay seem to have a larger impact.



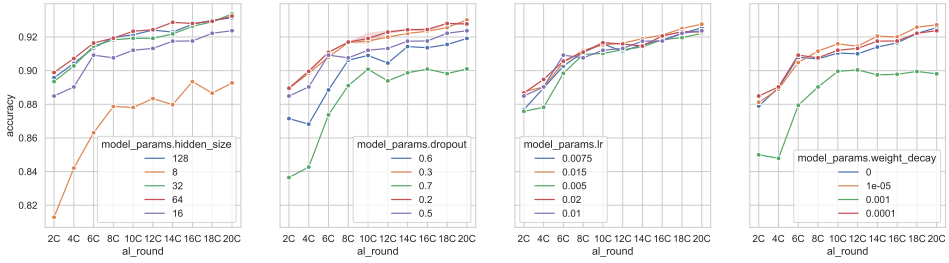
(a) Cora.



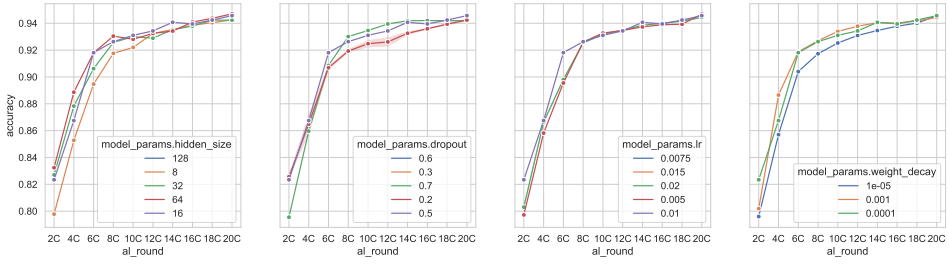
(b) Citeseer.



(c) Pubmed.



(d) CS.



(e) Physics.

Fig. 4: Hyperparameter variation. We show results when increasing and decreasing values for hidden size (column 1), dropout (column 2), learning rate (column 3), and weight decay (column 4) .

D VERIPS: Verified Pseudo-label Selection for Deep Active Learning

Authors

Sandra Gilhuber, Philipp Jahn, Yunpu Ma, and Thomas Seidl

Venue

2022 IEEE International Conference on Data Mining (ICDM), pages 951–956. IEEE, 2022.

DOI

https://doi.org/10.1109/icdm54844.2022.00113

Code

https://github.com/lmu-dbs/VERIPS

Declaration of Authorships

Sandra Gilhuber, Philipp Jahn, and Yunpu Ma developed and conceptualized the research idea. Philipp Jahn implemented and evaluated the approach with the support of Sandra Gilhuber. Sandra Gilhuber, Philipp Jahn, and Yunpu Ma analyzed the results and discussed the findings. Sandra Gilhuber wrote the manuscript and revised it together with Philipp Jahn and Thomas Seidl.

Author's Note

This work was accepted as a short paper. The supplementary material includes additional experiments that were omitted from the proceedings version due to page limitations.

Copyright Notice

VERIPS: Verified Pseudo-label Selection for Deep Active Learning

Sandra Gilhuber[1,2], Philipp Jahn[1,2], Yunpu Ma[1,2], Thomas Seidl[1,2]
[1]LMU Munich, Germany
[2]Munich Center for Machine Learning, Germany
{gilhuber, jahn, ma, seidl}@dbs.ifi.lmu.de

***Abstract*—Active learning has the power to significantly reduce the amount of labeled data needed to build strong classifiers. Existing active pseudo-labeling methods show high potential in integrating pseudo-labels within the active learning loop but heavily depend on the prediction accuracy of the model. In this work, we propose VERIPS, an algorithm that significantly outperforms existing pseudo-labeling techniques for active learning. At its core, VERIPS uses a pseudo-label verification mechanism that consists of a second network only trained on data approved by the oracle and helps to discard questionable pseudo-labels. In particular, the verifier model eliminates all pseudo-labels for which it disagrees with the actual task model. VERIPS overcomes the problems of poorly performing initial models, e.g., due to imbalanced or too small initial pools, where previous methods select too many incorrect pseudo-labels and recovering takes long or is not possible. Moreover, VERIPS is particularly insensitive to parameter choices that existing approaches suffer from. Our code is available at https://github.com/lmu-dbs/VERIPS.**

***Index Terms*—active learning, pseudo-labeling, image classification**

I. INTRODUCTION

High labeling costs are a major challenge for many real-world applications, while vast amounts of unlabeled data are available almost for free. One approach to reduce annotation costs is active learning, which alternates between model training, sample acquisition, and manual annotation to find the most informative and meaningful set of data possible.

Active learning algorithms for image classification tasks vary between searching for the most informative samples [1], a diverse subset [2] or both [3].

However, most of them miss the opportunity to exploit the rich information hidden in the unlabeled data. In contrast, exploiting unlabeled data is the key idea of semi-supervised learning. A lot of semi-supervised research focuses on consistency regulation [4]–[7] and yield great successes. Key concepts from this direction have already been considered in semi-supervised active learning [8]–[10]. However, these approaches depend on domain-specific data augmentations and are computationally intensive. On the other hand, pseudo-labeling is a universally applicable concept with comparatively low computational complexity. The idea of pseudo-labeling is simple: the model provides artificial labels for data samples that yield high confidence scores and incorporates those samples as additional data into the classifier's training. A natural observation is that confidence scores also play a crucial role in active learning, except that the opposite side of the scale is more relevant. In particular, samples with high predictive uncertainty are considered very informative and are subsequently sent to the oracle for an accurate label. Since active learning and pseudo-labeling have a similar ambition to determine the uncertainty of a model but look at different ends of the same scale, their combination is very elegant and possible without much additional effort.

A popular approach to combining pseudo-labels with active learning is CEAL [11]. The method successfully employs entropy-based pseudo-labels with threshold decay and shows promising results. However, CEAL needs two hand-crafted parameters for selecting pseudo-labels: a confidence threshold and a decay rate that adapts this threshold over the active learning loop. Since the selection of pseudo-labels is directly dependent on the threshold parameter, a sub-optimal choice can lead to the selection of many wrong artificial labels, especially in the early stages of active learning.

A sub-optimal initial pool, i.e., there are not enough samples of one category present or the pool is too small in general, can also lead to poorly calibrated initial models misguiding the pseudo-label selection [9], [12]. However, it is not trivial to determine an optimal size or composition of the initial pool [13], so it is often selected randomly in practice. Too large pools depict an under-utilization of active learning, which harms the sample efficiency. Given this challenge, pseudo-labeling is either a powerful tool to advance knowledge or decrease the generalization performance if the model is continuously built on incorrect predictions [14].

In this work, we discuss these open challenges of combining active learning with pseudo-labeling and propose a novel method that uses a verification mechanism aiming at automatic identification and subsequent elimination of questionable pseudo-labels to exploit the power of pseudo-labeling safely.

Figure 1 illustrates the overall concept of our novel method **VERIPS**: **VERI**fied **P**seudo-label **S**election for deep active learning. The key idea is to employ a verification mechanism using a second so-called verification model, which only learns from oracle-approved data and thus does not receive possibly misleading, artificially labeled training samples. The verification model refines the pseudo-labeled pool in case of disagreement with the target model. However, an omniscient verifier capable of discerning true pseudo-labels is *not* required here. For our mechanism, it is not crucial whether the target

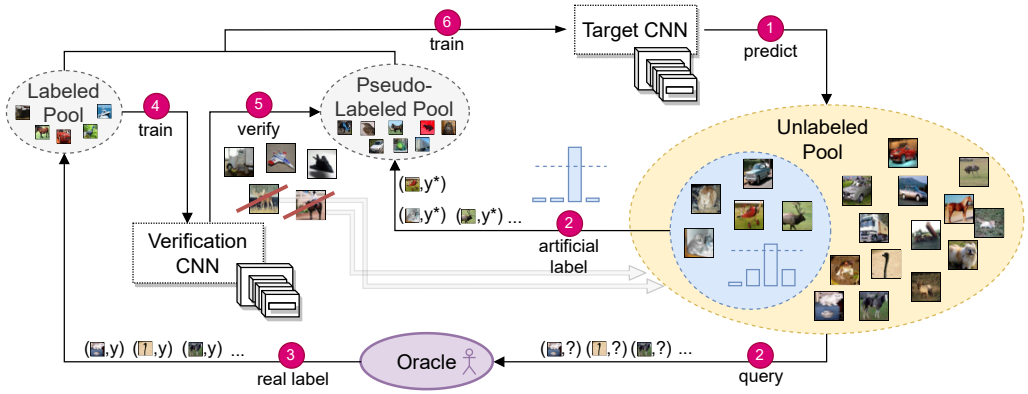


Fig. 1: Conceptual illustration of our method VERIPS. ① The target CNN, initially trained on a small dataset, makes predictions on the unlabeled pool. ② High confidence samples get pseudo-labels, informative samples are sent to the oracle which provides annotations ③. ④ A verification network is trained on the oracle-approved labeled data. ⑤ If the verification model does not agree on the artificially chosen label provided by the target model, the corresponding sample is sent back to the unlabeled pool. ⑥ The target model retrains on both the labeled and pseudo-labeled data.

or verification model is correct or not. Rather, it is a matter of preferring to play it safe and discarding more pseudo-labels than keeping potentially false ones. This mechanism is intended to provide support for the target model, erasing many pseudo-labels while the labeled pool is still too small to actually trust the model predictions and bypass this critical early phase safely. A valuable benefit of our approach is that the influence of a handcrafted confidence threshold parameter is reduced, and we do not need other parameters. VERIPS is elegant and simple and compatible with any active learning strategy in its base form. In extensive experiments evaluated on image classification tasks, we show that VERIPS ensures robust learning behavior along all stages of the active learning cycle, outperforming existing active learning strategies with and without pseudo-labels.

In summary, our main contributions are the following:

- We propose VERIPS, a novel active learning algorithm that uses pseudo-labeling. In particular, we introduce a verification component, which significantly reduces the selection of false pseudo-labels and thus increases label efficiency.
- In particular, we propose two manifestations of VERIPS and demonstrate their superiority over common active strategies with and without pseudo-labeling in extensive experiments on benchmark image classification tasks. Our method is insensitive to parameter choices, which existing active pseudo-labeling methods suffer from.

II. RELATED WORK

Active learning aims at reducing labeling costs by intelligently choosing the data that should be labeled. Active learning has been explored on various tasks, such as applications including graph data [15]–[20] or text [21]–[24]. However, in this work, we mainly consider approaches evaluated on image classification. For a detailed and thorough summary of applications and approaches, we point to [25] or one of [26], [27] for a deep learning oriented portrayal.

Uncertainty-based methods rank the unlabeled instances based on the uncertainty of the model regarding its predictions to estimate the informativeness of instances. Popular estimates are least confidence, margin [25], and entropy [28]. Query-by-committee [25], [29] describes a group of methods favoring instances where several models disagree about the class. In line with the idea of committee approaches, BALD [1] attempts to exploit disagreement over multiple different predictions using only one neural network by applying dropout at inference time and thus getting a distribution over the weights. The batch-setting, where multiple instances are sent to the oracle instead of one at a time, led to the emergence of diversity-based methods [2], [30]–[32] that seek to minimize redundancies within an acquisition round. A prominent hybrid approach with a novel notion of uncertainty is BADGE [3]. Uncertainty is based on so-called gradient embeddings, i.e., the magnitude of the gradient of a sample with respect to the final layer. However, the so far mentioned approaches in this section do not exploit the unlabeled pool.

In contrast, VERIPS is a novel active learning method with the main focus on safely integrating pseudo-labels and is therefore also related to semi-supervised learning, in particular to pseudo-labeling [33]. Pseudo-labeling can be successfully used in combination with consistency regulation, which aims at training models that are invariant to different data augmentations [6], [7], [34]. Such techniques have already been considered in some active semi-supervised works. One idea is to use data augmentations to determine uncertainty [8], [9], [35]. Others directly use advanced semi-supervised models [10] in the active learning loop. However, these methods rely on domain-specific data augmentations. Such adequate

augmentations might not be known or applicable. In contrast, our method is universally applicable and does not require domain-specific augmentations.

CEAL [11] is a well-cited active pseudo-labeling approach that uses entropy to estimate the confidence ranking and selects samples above a time-weighted threshold as pseudo-labeled training instances. Though the approach shows promising results, there are open challenges. Conventional pseudo-labeling often underperforms due to uncalibrated models providing noisy label suggestions [12]. In particular, neural networks often tend to be overconfident [36], [37] leading to high confidence scores for wrong pseudo-labels [12]. However, in active learning, we start with a very limited budget enforcing the problem of noisy predictions. As a result, CEAL selects many false artificial labels, which in turn leads to slow model convergence [9] since the model overfits on incorrect pseudo-labels [14]. We hypothesize that there is a lot of potential in combining pseudo-labeling with active learning, but using an uncertainty-aware selection scheme is not sufficient to overcome the problem of poorly performing initial models.

This work combines active learning with pseudo-labeling and proposes a novel component that alleviates the problem of selecting wrong and, thus, misleading information from artificially chosen labels.

III. Proposed Method VERIPS

In this section, we describe the building blocks of our approach and outline how they fit into the active learning cycle. Suppose we are given a large amount of unlabeled images $\mathcal{U} = \{x\}_{i=1}^{N}$ and a small amount of labeled images $\mathcal{L} = \{(x_i, y_i)\}$ which is randomly drawn from $\mathcal{U}$. We want to solve multi-class classification with C categories. A classifier denoted by θ learns a mapping from images to labels. $P(y|x;\theta)$ denotes the probability given the model θ and a sample x that x belongs to class y. We introduce a pseudo-labeled pool $\mathcal{P}$ containing instances of the form $(x, y*)$ where $y*$ denotes an artificially chosen label.

A. Verification Mechanism

The core of our method is the verification mechanism to refine the pseudo-label pool. The main goal is to keep pseudo-labels where the target model has assigned the correct class and to detect and eliminate false pseudo-labels. The problem is that the true class is unknown, so the model confidence is usually used as an indicator. However, relying on the prediction of a single network, especially if it was only trained on a small amount of data, is risky since neural networks tend to be overconfident [36], [37].

We suggest incorporating another network that gives a second opinion on the pseudo-labels. We call this the second network verification model, and in contrast to the actual task model, the verification model does not see potentially misleading artificial labels during training. The verifier and target models use the same backbone and training parameters; the only difference is the data they see during the training process. Specifically, a pseudo-label is discarded if:

$$\arg\max_c P(y_c|x;\theta_{Tar}^{rd}) \neq \arg\max_c P(y_c|x;\theta_{Ver}^{rd}), \quad (1)$$

where θ_{Tar}^{rd} are the model weights derived from the target model and θ_{Ver}^{rd} from the verification model in a specific round rd respectively. In other words, no matter how uncertain one of the models is about its current prediction, we discard a pseudo-labeled sample if they disagree on the same label. This is a straightforward yet effective check to reveal discrepancies and clean the pseudo-label pool.

We emphasize that we do not expect the verification model to be more reliable than the target model. The verification network acts as a discussion partner for the target model. It does not matter that the verification model itself does not yield very accurate predictions in the beginning. Especially at early iterations, they are very likely to disagree on most decisions. This puts more focus on the actual labels released by the oracle, ensuring healthy training until model confidence rises and an agreement among the models is established. The influence of the pseudo-labeling is thus automatically adjusted to the training state of the models and does not rely solely on the perfect selection of the threshold parameter or the predictive performance of a single model.

B. Selection of Pseudo-Labels

Suppose we receive an uncertainty estimate $u(x)$ for an instance x where larger values denote higher uncertainty. We decide whether to derive a one-hot encoded pseudo-label $y*$ of a sample x and add it to the pseudo-labeled pool $\mathcal{P}$ by:

$$y* = \begin{cases} \arg\max_c P(y_c|x;\theta_{Tar}) & \text{if } u(x) \geq \lambda \\ 0 & \text{if } u(x) < \lambda \end{cases} \quad (2)$$

where λ is a threshold parameter that intuitively tells us at which point we allow to trust the prediction. VERIPS does not use a decay rate to update the threshold since the verification mechanism already is an automatism that controls the selection of pseudo-labels.

C. Algorithm

Our goal is to iteratively retrain the model θ_T to obtain a strong classifier in the end. To accomplish this, we start the active learning loop and repeat the following steps until the required number of rounds R have been executed: First, we perform an acquisition step $a(\mathcal{U}, \theta_{Tar})$ of a predefined active learning strategy using the current target model θ_{Tar} and the unlabeled pool U. The oracle annotates the returned samples, and we move them to the labeled pool $\mathcal{L}$. Then we train the verification model θ_{Ver} on the increased labeled pool $\mathcal{L}$. Additionally, we select samples exceeding the threshold λ^{rd} following Eq. (2) and move them together with their one-hot encoded most probable class label from $\mathcal{U}$ to $\mathcal{P}$. Next, we visit each sample in the pseudo-labeled pool $\mathcal{P}$ and move it back to the unlabeled pool $\mathcal{U}$ if the verification model disagrees with the target model on the artificially chosen class label. The

Algorithm 1 VERIPS

Input: Unlabeled data pool $\mathcal{U}$, initially labeled data pool $\mathcal{L}$, pseudo-labeled pool $\mathcal{P} = \emptyset$, number of acquisition rounds R, AL batch-size B, initialized model θ^0_{Tar}, acquisition strategy $a(\cdot, \cdot)$, threshold λ.

for $rd = 0, 1, 2, \ldots, R$ **do**
  Obtain label for B samples based on $a(\mathcal{U}, \theta^{rd}_{Tar})$ and move them from $\mathcal{U}$ to $\mathcal{L}$.
  Move confident samples exceeding λ^{rd} from $\mathcal{U}$ to $\mathcal{P}$ based on Eq. (2).
  Train verification model θ^{rd}_{Ver} using $\mathcal{L}$.
  for $x \in \mathcal{P}$ **do**
    Remove x from $\mathcal{P}$ if it does not hold Eq. (1).
  end for
  Retrain θ^{rd}_{Tar} using $\mathcal{L} \cup \mathcal{P}$.
end for
return Final model parameters θ^R_{Tar} obtained in round R

verification model only learns from oracle-approved samples and has already seen the new labeled instances. However, we do not compare uncertainty estimates in the verification step; we only care about discrepancies in the class label's vote. Both the pseudo-labeled pool $\mathcal{P}$ as well as the labeled pool $\mathcal{L}$ comprise the training data for the target model θ_{Tar}. The loop finishes after a fixed number of rounds R, and we finally return the target model θ^R_{Tar}. The complete algorithm is shown in Algorithm 1.

D. VERIPS$_{\text{M}}$ and VERIPS$_{\text{E}}$

We propose two explicit variants of VERIPS: (a) VERIPS$_{\text{M}}$ uses margin as uncertainty estimate for both active acquisition and pseudo-labeling, i.e., $u(x) := 1 - (P(y_1|x;\theta) - P(y_2|x;\theta))$, where y_c ranges over the classes, and (b) VERIPS$_{\text{E}}$ analogously uses entropy, i.e., $u(x) := -\sum_c P(y_c|x;\theta) \log P(y_c|x;\theta)$. Both uncertainty estimates are commonly known and popular choices in the active learning community. Towards a simple and intuitive design, we use the same heuristic for active selection and pseudo-labeling and do not mix them. We follow a rank-based selection scheme without diversity sampling. VERIPS could be used with any uncertainty heuristic and active learning strategy. However, we only focus on the proposed two variants due to space limitations and leave further combinations open for future investigations.

IV. EXPERIMENTS

We evaluate our approach on benchmark datasets for image classification, namely SVHN [38], CIFAR10 [39] and MNIST [40].

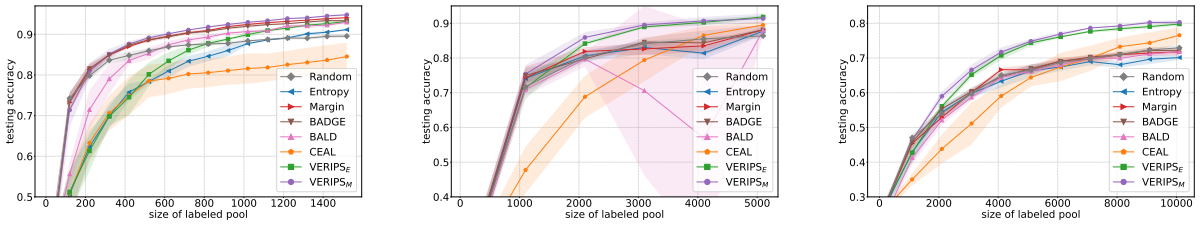
a) Training and Model: For SVHN and CIFAR10, we use a VGG16 [41]. For BALD [1] we use two dropout layers with dropout rate 0.5. We use early stopping on training accuracy when a value of 0.99 is reached with a maximum of 200 epochs and a learning rate of 0.001. For the rather simple dataset MNIST we only use a single-layered neural network with a hidden dimensionality of 256. All experiments can be reproduced using our publicly available code base.

b) Active Learning and Pseudo-Labeling: The initial pool contains 100 labeled images for CIFAR10 and SVHN and only 20 images for MNIST, and we add additional 1,000 images per active learning iteration for the former and 100 per iteration for MNIST. We repeat all experiments five times and report average test accuracy as well as the standard deviation. We consider the following active learning strategies: BADGE [3], BALD [1], MARGIN and ENTROPY [25]. Additionally, we include CEAL [11] as a direct competitor using pseudo-labels. We follow the original implementation of CEAL and set the decay rate $dr = 0.0033$, the starting threshold $\lambda_0 = 0.05$. For our proposed variants VERIPS$_{\text{E}}$ and VERIPS$_{\text{M}}$ we also use a threshold of $\lambda_0 = 0.05$.

A. Results of VERIPS

Fig. 2 depicts the learning curves of the selected active learning methods evaluated on MNIST, SVHN, and CIFAR10 with the average test accuracy on the y-axis and the number of labeled samples on the x-axis. Table I denotes exact results for different annotation budgets. Bold numbers indicate the best and underlined numbers indicate the second best performing methods. For all datasets, our method VERIPS$_{\text{M}}$ yields the strongest results for all labeling budgets. VERIPS$_{\text{E}}$ yields similar results for CIFAR10 and SVHN. However, for MNIST VERIPS$_{\text{E}}$ performs similar to the plain ENTROPY sampling baseline without pseudo-labels until roughly 500 labels are reached and cannot reach the strong performance of VERIPS$_{\text{M}}$ until the final budget. We assume this is because MNIST is a rather simple dataset where the power of pseudo-labeling is not necessarily needed. However, of the three entropy-based active learning methods, VERIPS$_{\text{E}}$ is nonetheless the strongest. However, VERIPS$_{\text{M}}$ is always superior, and on both CIFAR10 and SVHN, there is quite a gap between our methods and the others. Especially for the most challenging task on CIFAR10, we observe up to 10% more accurate predictions and superior performance for all budgets. The result of BALD on SVHN is quite interesting; for one particular run, the performance dropped drastically at some point, resulting in this large standard deviation. We hypothesize that the instability only occurs for BALD since it uses two additional dropout layers as described previously. However, probably most interestingly, CEAL does not perform very well across all datasets despite the additional use of pseudo-labels. CEAL often needs several iterations to outperform the active learning baseline (CIFAR10, SVHN) and, for MNIST, performs worst out of all strategies. For example, on CIFAR10 CEAL needs over 7,000 samples to get better accuracy than ENTROPY. Even the random sampling baseline outperforms CEAL, especially at the beginning. Moreover, both VERIPS variants have fewer fluctuations across multiple runs with different seeds, which implies that our method is generally more stable.

We assume that CEAL underperforms due to many wrong pseudo-labels in the early active learning stages. The model learns incorrect mappings, which leads to unjustified high

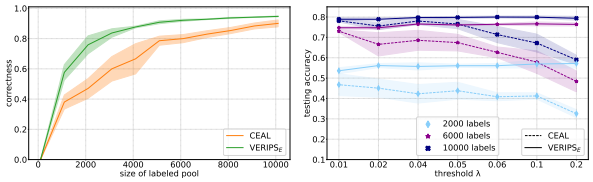


(a) MNIST (b) SVHN (c) CIFAR10

Fig. 2: Test accuracy (y-axis) for different active learning methods at each acquisition round (x-axis) evaluated on different datasets. Our method VERIPS$_M$ yields better test accuracy than CEAL and all active learning baselines on all datasets.

TABLE I: Average test accuracy and standard deviation of active learning methods for different labeling budgets and datasets. Bold numbers indicate best performing methods and underlined second best. Our methods (bottom) perform best.

Dataset	MNIST			SVHN			CIFAR10		
Actively chosen data	500	1,000	1,500	1,000	3,000	5,000	2,000	6,000	10,000
RANDOM	85.9 ± 0.4	88.4 ± 0.2	89.5 ± 0.3	71.7 ± 1.6	84.1 ± 0.8	86.3 ± 0.6	54.0 ± 1.3	68.7 ± 0.8	72.9 ± 0.8
ENTROPY	78.3 ± 0.5	87.7 ± 0.3	91.1 ± 0.2	74.1 ± 1.6	83.0 ± 1.9	87.7 ± 1.1	54.3 ± 1.7	67.4 ± 1.1	70.1 ± 0.6
MARGIN	88.6 ± 0.5	92.4 ± 0.2	94.0 ± 0.0	74.3 ± 3.1	82.5 ± 1.7	88.0 ± 0.4	52.8 ± 0.9	69.1 ± 0.9	71.7 ± 0.8
BADGE	88.6 ± 0.4	92.0 ± 0.3	93.3 ± 0.3	74.9 ± 2.1	84.5 ± 1.3	88.1 ± 1.2	55.3 ± 0.9	69.0 ± 0.6	72.0 ± 0.8
BALD	85.9 ± 0.4	88.4 ± 0.2	89.5 ± 0.3	70.9 ± 1.8	70.5 ± 24.4	87.9 ± 1.2	52.1 ± 1.4	68.0 ± 0.6	71.7 ± 1.2
CEAL	78.5 ± 4.0	81.5 ± 3.7	84.5 ± 3.3	47.7 ± 6.7	79.4 ± 0.6	89.5 ± 1.0	43.8 ± 4.1	67.4 ± 4.0	76.5 ± 2.2
VERIPS$_E$	81.0 ± 0.2	90.9 ± 0.2	93.5 ± 0.1	71.4 ± 3.3	88.9 ± 0.3	**91.8 ± 0.3**	55.7 ± 0.2	**76.9 ± 0.1**	79.9 ± 0.2
VERIPS$_M$	**89.1 ± 0.4**	**92.9 ± 0.1**	**94.8 ± 0.2**	**75.3 ± 2.5**	**89.5 ± 0.6**	91.3 ± 1.4	**58.9 ± 1.3**	**76.9 ± 0.6**	**80.2 ± 0.4**



(a) Correctness Ratio. (b) Varying thresholds.

Fig. 3: Left: Correctness ratio of pseudo-labels (y-axis) derived with CEAL and VERIPS at each acquisition round (x-axis). Right: Sensitivity of VERIPS$_E$ (solid) and CEAL (dashed) to threshold λ for different labeling budgets (colors).

confidence scores for other unlabeled samples, which are then chosen for the pseudo-labeled pool. This leads to a downward spiral from which the model has difficulty recovering. In the end, the model has seen many confusing and contradicting samples from the oracle-approved annotations and the pseudo-labeled pool and thus does not achieve the test accuracy reached by the other strategies. Our interim conclusion is that the use of pseudo-labeling in conjunction with active learning has high potential but can also be fragile. In the following subsections, we will provide more investigations to study the observed behavior of the active pseudo-label methods.

B. Pseudo-label Correctness

To better understand the reasons for the large performance gap between the pseudo-label methods CEAL and our method VERIPS, we take a closer look at the pseudo-label correctness ratio for the different acquisition steps in Fig. 3a. We focus on VERIPS$_E$ to ensure better comparability between the two methods. VERIPS$_E$ (green line) consistently depicts a higher correctness ratio, reaching almost 95% accuracy of the artificially chosen samples in the last iteration, with only minor variations over multiple runs. In contrast, CEAL (orange line) only reaches around 90% in the end and also has higher fluctuations across several seeds. As intended, VERIPS$_E$ successfully increases the correctness ratio resulting in strong results straight from the beginning.

C. Sensitivity Analysis

Fig. 3b shows how the average test accuracy (y-axis) of CEAL and VERIPS$_E$ changes with varying thresholds λ (x-axis) evaluated on CIFAR10. Different line colors indicate different labeling budgets, and the shaded area indicates the standard deviation. For better comparability, we fix the decay rate to the originally proposed setting of CEAL and again use our entropy-based variant. The test accuracy of VERIPS$_E$ is fairly stable across all parameters, clearly visible by the parallelism to the x-axis. In contrast, CEAL is susceptible to the threshold, especially when the threshold is set too large since there is no mechanism filtering wrong pseudo-labels. This dependency can not be solved trivially since the optimal threshold can vary between different training settings and datasets. However, VERIPS$_E$ yields robust and superior test accuracies mostly insensitive to the selected threshold.

V. CONCLUSION AND FUTURE WORK

To summarize, we have discussed the challenges of combining active learning with pseudo-labeling. When there is hardly

any labeled data available, pseudo-labels are often incorrect due to unreliable model predictions. On the other hand, it is hard to determine the optimal starting point for including pseudo-labels. As a solution, we presented VERIPS, a novel active learning approach capable of safely exploiting the power of pseudo-labeling. The key idea is to use a dedicated verification network that helps identifying incorrect artificial labels and reduces the dependence on external parameters. Our experiments demonstrate the effectiveness and stability of our proposed method on various image classification datasets.

We leave further investigation on whether we can omit the threshold parameter completely and how applicable and successful our verification component is in combination with other active learning strategies open for future work.

REFERENCES

[1] Y. Gal, R. Islam, and Z. Ghahramani, "Deep bayesian active learning with image data," in *ICML*. PMLR, 2017, pp. 1183–1192.

[2] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," in *ICLR*, 2018.

[3] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal, "Deep batch active learning by diverse, uncertain gradient lower bounds," in *ICLR*, 2019.

[4] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," *Advances in neural information processing systems*, vol. 29, 2016.

[5] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in neural information processing systems*, vol. 30, 2017.

[6] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in Neural Information Processing Systems*, vol. 33, pp. 596–608, 2020.

[7] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[8] M. Gao, Z. Zhang, G. Yu, S. Ö. Arık, L. S. Davis, and T. Pfister, "Consistency-based semi-supervised active learning: Towards minimizing labeling cost," in *ECCV*. Springer, 2020, pp. 510–526.

[9] M. Ducoffe and F. Precioso, "Adversarial active learning for deep networks: a margin based approach," *arXiv preprint arXiv:1802.09841*, 2018.

[10] S. Song, D. Berthelot, and A. Rostamizadeh, "Combining mixmatch and active learning for better accuracy with fewer labels," *arXiv preprint arXiv:1912.00594*, 2019.

[11] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, "Cost-effective active learning for deep image classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 12, pp. 2591–2600, 2016.

[12] M. N. Rizve, K. Duarte, Y. S. Rawat, and M. Shah, "In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning," in *ICLR*, 2020.

[13] A. L. Chandra, S. V. Desai, C. Devaguptapu, and V. N. Balasubramanian, "On initial pools for deep active learning," in *NeurIPS 2020 Workshop on Pre-registration in Machine Learning*. PMLR, 2021, pp. 14–32.

[14] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," in *IJCNN*. IEEE, 2020, pp. 1–8.

[15] M. Berrendorf, E. Faerman, and V. Tresp, "Active learning for entity alignment," in *ECIR*. Springer, 2021, pp. 48–62.

[16] W. Zhang, Y. Wang, Z. You, M. Cao, P. Huang, J. Shan, Z. Yang *et al.*, "Rim: Reliable influence-based active learning on graphs," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[17] H. Cai, V. W. Zheng, and K. C.-C. Chang, "Active learning for graph embedding," *arXiv preprint arXiv:1705.05085*, 2017.

[18] Y. Wu, Y. Xu, A. Singh, Y. Yang, and A. Dubrawski, "Active learning for graph neural networks via node feature propagation," *arXiv preprint arXiv:1910.07567*, 2019.

[19] L. Gao, H. Yang, C. Zhou, J. Wu, S. Pan, and Y. Hu, "Active discriminative network representation learning," in *IJCAI*, 2018.

[20] W. Zhang, Y. Shen, Y. Li, L. Chen, Z. Yang, and B. Cui, "Alg: Fast and accurate active learning framework for graph convolutional networks," in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 2366–2374.

[21] C. Schröder and A. Niekler, "A survey of active learning for text classification using deep neural networks," *arXiv preprint arXiv:2008.07267*, 2020.

[22] A. Zhang, B. Li, W. Wang, S. Wan, and W. Chen, "Mii: A novel text classification model combining deep active learning with bert," *Computers Materials and Continua*, vol. 63, no. 3, pp. 1499–1514, 2020.

[23] J. Lu, M. Henchion, and B. Mac Namee, "Investigating the effectiveness of representations based on word-embeddings in active learning for labelling text datasets," *arXiv preprint arXiv:1910.03505*, 2019.

[24] R. Hazra, P. Dutta, S. Gupta, M. A. Qaathir, and A. Dukkipati, "Active^2 learning: Actively reducing redundancies in active learning methods for sequence tagging and machine translation," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tür, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, Eds. ACL, 2021, pp. 1982–1995. [Online]. Available: https://doi.org/10.18653/v1/2021.naacl-main.159

[25] B. Settles, "Active learning literature survey," 2009.

[26] P. Liu, L. Wang, G. He, and L. Zhao, "A survey on active deep learning: From model-driven to data-driven," *arXiv preprint arXiv:2101.09933*, 2021.

[27] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM Computing Surveys (CSUR)*, vol. 54, no. 9, pp. 1–40, 2021.

[28] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE mobile computing and communications review*, vol. 5, no. 1, pp. 3–55, 2001.

[29] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Machine learning*, vol. 28, no. 2, pp. 133–168, 1997.

[30] F. Zhdanov, "Diverse mini-batch active learning," *arXiv preprint arXiv:1901.05954*, 2019.

[31] A. Kirsch, J. Van Amersfoort, and Y. Gal, "Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning," *Advances in neural information processing systems*, vol. 32, pp. 7026–7037, 2019.

[32] V. Prabhu, A. Chandrasekaran, K. Saenko, and J. Hoffman, "Active domain adaptation via clustering uncertainty-weighted embeddings," in *ICCV*, 2021, pp. 8505–8514.

[33] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, no. 2, 2013, p. 896.

[34] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki, "Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[35] O. Siméoni, M. Budnik, Y. Avrithis, and G. Gravier, "Rethinking deep active learning: Using unlabeled data at model training," in *ICPR*. IEEE, 2021, pp. 1220–1227.

[36] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *ICML*. PMLR, 2017, pp. 1321–1330.

[37] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *ICML*. PMLR, 2016, pp. 1050–1059.

[38] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[39] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[40] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

Supplementary Materials to "VERIPS: Verified Pseudo-label Selection for Deep Active Learning"

Sandra Gilhuber[1,2], Philipp Jahn[1,2], Yunpu Ma[1,2], Thomas Seidl[1,2]
[1]LMU Munich, Germany
[2]Munich Center for Machine Learning, Germany
{gilhuber, jahn, ma, seidl}@dbs.ifi.lmu.de

APPENDIX

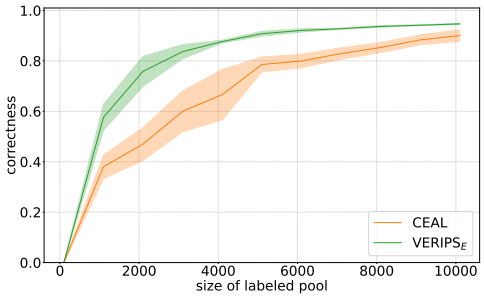
A. Further Details on the Pseudo-label Selection

As shown in Figure 1a, VERIPS has higher pseudo-label correctness than CEAL. Intuitively one could think that for VERIPS$_E$ the amount of pseudo-labels that are kept for training is smaller than for CEAL due to the intended filtering of pseudo-labels. However, Fig. 1b reveals that VERIPS (green) almost selects as many pseudo-labels as CEAL (orange) over the whole active learning course. The bars indicate the total amount of pseudo-labels and labeled samples for each class incorporated in training. In Figure 1c, we illustrate the pseudo-label selection behavior of VERIPS (green) and CEAL (orange). The line styles in the bars indicate the amount of **kept correct** pseudo-labels (lines directed right), **kept incorrect** pseudo-labels (stars), **removed correct** pseudo-labels (lines directed left), and **removed incorrect** pseudo-labels (grid). CEAL does not remove pseudo-labels and does not have "remove" bars.

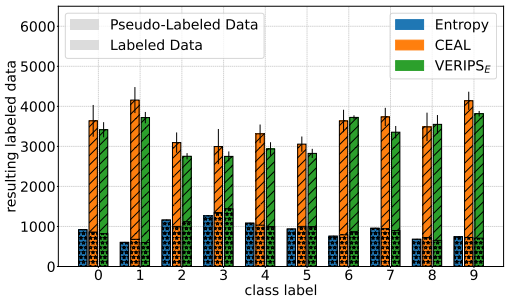
Both methods would choose a similar number of pseudo-labels in the first and second iterations. However, VERIPS discards many of them, and fewer wrong pseudo-labels are kept for training. As a result, the model gets more confident, and more pseudo-labels fall over the threshold at the third and subsequent rounds. To summarize, CEAL and VERIPS$_E$ both select similarly many pseudo-labeled samples in total, but the fraction of correct pseudo-labels is much better for VERIPS.

B. Effect of Imbalanced and Small Initial Pools

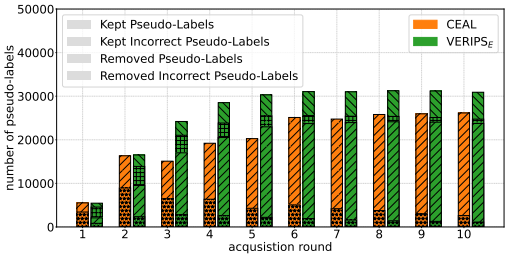
The initial pool, which is usually drawn randomly, can affect the performance of the active learner [1]. For instance, imbalanced or too small initial pools can lead to poorly trained models in the early iterations, leading to unreliable uncertainty scores and pseudo-labels. To analyze how sensitive VERIPS is regarding such situations, we conduct additional experiments with different initial pool settings. Fig. 2 depicts the performance of VERIPS$_E$ and CEAL on CIFAR10 using an imbalanced initial data distribution (50% of the classes are not included in the initial pool) (Fig. 2a) and a smaller initial pool with only 50 samples (Fig. 2b). VERIPS$_E$ is largely not affected by the imbalanced initial set and the small starting set and yields stable results in the long term. In contrast, CEAL yields unstable and worse results.



(a) Correctness Ratio.

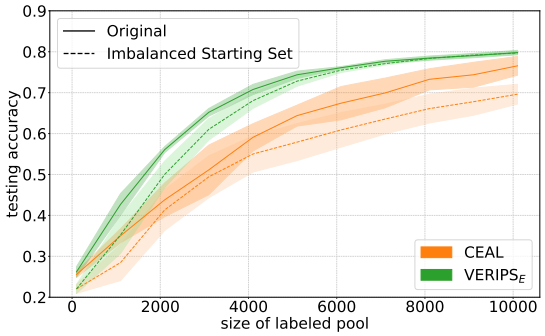


(b) Number of pseudo-labels and annotated labels (y-axis) divided by class (x-axis). Dashed bars indicate pseudo-labels, and star-filled bars indicate oracle-given labels. Despite the filtering, VERIPS$_E$ exhibits only slightly less overall training data than CEAL.

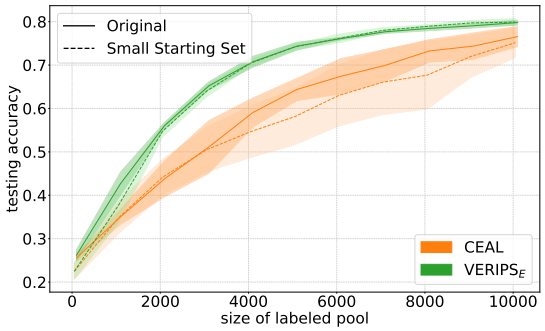


(c) Number of pseudo-labels (y-axis) above threshold λ for each acquisition round (x-axis). After the second iteration, VERIPS$_E$ receives much more confident instances falling over the pseudo-label threshold. Moreover, VERIPS$_E$ keeps many more correct pseudo-labels (dashed right bars) and fewer incorrect ones (star-filled bars).

Fig. 1: Effect of pseudo-label refinement with VERIPS.



(a) Imbalanced initial pool comprising samples from only 50% of the available classes.



(b) Small initial pool containing 50 (Small Starting Set) vs 100 (Original) samples.

Fig. 2: Robustness of VERIPS and CEAL regarding a smaller or an imbalanced initial pool. Dashed lines indicate the manipulated version of the data and solid lines the data used in the main experiments.

TABLE I: Average test accuracy and standard deviation of accelerated and original VERIPS, active learning without pseudo-labeling (in %) and CEAL evaluated on **CIFAR10**.

Budget	2,000	4,000	8,000	10,000
	Margin			
Accelerated	53.5 ± 1.0	69.4 ± 1.0	79.1 ± 0.3	**80.2 ± 0.4**
VERIPS$_M$	**58.9 ± 1.3**	**71.7 ± 1.3**	**79.2 ± 0.2**	**80.2 ± 0.4**
w/o PL	52.8 ± 0.9	66.6 ± 0.5	70.9 ± 0.6	71.7 ± 0.8
	Entropy			
Accelerated	51.9 ± 1,7	67.4 ± 0.7	76.7 ± 1.7	77.9 ± 1.3
VERIPS$_E$	55.7 ± 2.5	71.0 ± 0.6	79.1 ± 0.4	79.9 ± 0.2
w/o PL	54.3 ± 1.7	63.2 ± 1.7	68.0 ± 0.8	70.1 ± 0.6
CEAL	43.8 ± 4.1	59.0 ± 3.3	73.2 ± 2.5	76.5 ± 2.2

C. *Accelerated* VERIPS

In VERIPS, the verification step in acquisition round rd utilizes the updated verification model trained on the instances chosen by the target model in that round. Thereby, always the newest instances are included in the verification to enhance the pseudo-label selection. However, this handling requires training the verification model before we can update the target model. Utilizing more than a single network for acquisition has been done in previous works [2]–[4]. However, it negatively affects the training times and computational costs within the AL process. To overcome this, we can consider an accelerated variant of VERIPS. In this variant, in round rd, the pseudo-labels produced by the task model θ_{Tar}^{rd} are verified by comparing them with the pseudo label suggestions produced by the verification model of the previous round θ_{Ver}^{rd-1}. That way, we can parallelize the training of the verification and target model in the current round and do not have to wait for the updated verification model. We compare the test accuracy of the accelerated version, the original version, and the baseline without pseudo-labeling for the entropy and margin-based methods in Table I. The accelerated version of VERIPS does not reach the same accuracy as the original versions of VERIPS in the earlier stages, but it yields similar results in later rounds. Moreover, accelerated VERIPS outperforms CEAL in all iterations. It also outperforms the baselines without pseudo-labels except for the very first iteration of the entropy-based variant. To summarize, which variant is best suited depends on the circumstances. Accelerated VERIPS is a good choice if the training time is a limiting factor. However, it does not yield the same robustness as VERIPS in early iterations. We recommend using the original version when the dataset is highly complex or the initial pool is suboptimal.

REFERENCES

[1] A. L. Chandra, S. V. Desai, C. Devaguptapu, and V. N. Balasubramanian, "On initial pools for deep active learning," in *NeurIPS 2020 Workshop on Pre-registration in Machine Learning*. PMLR, 2021, pp. 14–32.
[2] S. Sinha, S. Ebrahimi, and T. Darrell, "Variational adversarial active learning," in *ICCV*, 2019, pp. 5972–5981.
[3] K. Kim, D. Park, K. I. Kim, and S. Y. Chun, "Task-aware variational adversarial active learning," in *CVPR*, 2021, pp. 8166–8175.
[4] J. W. Cho, D.-J. Kim, Y. Jung, and I. S. Kweon, "Mcdal: Maximum classifier discrepancy for active learning," *arXiv preprint arXiv:2107.11049*, 2021.

E How to Overcome Confirmation Bias in Semi-Supervised Image Classification by Active Learning

Authors

Sandra Gilhuber*, Rasmus Hvingelby*, Mang Ling Ada Fok, and Thomas Seidl
* These authors contributed equally to this work.

Venue

Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD), pages 330–347. Springer, 2023

DOI

https://doi.org/10.1007/978-3-031-43415-0_20

Code

https://github.com/lmu-dbs/HOCOBIS-AL

Declaration of Authorships

Sandra Gilhuber and Rasmus Hvingelby proposed the research idea, developed and conceptualized it, and discussed it with Thomas Seidl. Sandra Gilhuber and Rasmus Hvingelby started the literature review, and Ada Fok finalized and summarized it. Sandra Gilhuber did the implementation. Sandra Gilhuber designed and conducted the experiments and analyzed their results. Sandra Gilhuber and Rasmus Hvingelby wrote and revised the manuscript and discussed it with Thomas Seidl.

How to Overcome Confirmation Bias in Semi-Supervised Image Classification by Active Learning

Sandra Gilhuber[1,2( )], Rasmus Hvingelby[3], Mang Ling Ada Fok[3], and Thomas Seidl[1,2,3]

[1] LMU Munich, Munich, Germany
{gilhuber,seidl}@dbs.ifi.lmu.de
[2] Munich Center for Machine Learning (MCML), Munich, Germany
[3] Fraunhofer IIS, Erlangen, Germany
rasmus.hvingelby@iis.fraunhofer.de

Abstract. Do we need active learning? The rise of strong deep semi-supervised methods raises doubt about the usability of active learning in limited labeled data settings. This is caused by results showing that combining semi-supervised learning (SSL) methods with a random selection for labeling can outperform existing active learning (AL) techniques. However, these results are obtained from experiments on well-established benchmark datasets that can overestimate the external validity. However, the literature lacks sufficient research on the performance of active semi-supervised learning methods in realistic data scenarios, leaving a notable gap in our understanding. Therefore we present three data challenges common in real-world applications: between-class imbalance, within-class imbalance, and between-class similarity. These challenges can hurt SSL performance due to confirmation bias. We conduct experiments with SSL and AL on simulated data challenges and find that random sampling does not mitigate confirmation bias and, in some cases, leads to worse performance than supervised learning. In contrast, we demonstrate that AL can overcome confirmation bias in SSL in these realistic settings. Our results provide insights into the potential of combining active and semi-supervised learning in the presence of common real-world challenges, which is a promising direction for robust methods when learning with limited labeled data in real-world applications.

1 Introduction

The success of supervised deep learning models largely depends on the availability of sufficient, qualitative labeled data. Since manual annotation is time-consuming and costly, various research directions focus on machine learning with limited labeled data. While Active Learning (AL) [5,40] aims to label only

S. Gilhuber and R. Hvingelby—Equal contribution.

D. Koutra et al. (Eds.): ECML PKDD 2023, LNAI 14170, pp. 330–347, 2023.
https://doi.org/10.1007/978-3-031-43415-0_20

the most informative and valuable data intelligently, semi-supervised learning (SSL) [8,13,41] aims to exploit the information in the unlabeled pool without asking for new labels. Given the complementary nature of SSL and AL, it is intuitive to explore their integration within a unified framework to maximize the utilization of the available data. However, the effectiveness of AL has been questioned recently [7,11,31,33]. Some works show that other learning paradigms capable of exploiting the unlabeled data do not experience added value from biased and intelligent data selection through AL [11].

However, these findings are mainly based on experiments on well-established, clean benchmark datasets. But, an excessive emphasis on benchmark performance can result in diminishing returns where increasingly large efforts lead to ever-decreasing performance gains on the actual task [29,45]. As a result, an exclusive evaluation of such benchmarks can raise concerns about the transferability of these results to challenges in real-world applications. Therefore, we review the literature on AL to understand which datasets are commonly used for evaluation and to what extent AL has been combined with SSL.

Toward a better understanding, we first categorize existing AL methods into four groups, namely uncertainty sampling, representativeness sampling, coverage-based sampling, and balanced sampling. Second, we introduce the following three real-world challenges: (1) *Between-class imbalance* (BCI), where the distribution over class instances is non-uniform, (2) *within-class imbalance* (WCI), where the intra-class distribution is non-uniform, and (3) *between-class similarity* (BCS), where the class boundaries are ambiguous. In our experiments, we demonstrate that each of these real-world challenges introduces confirmation bias reinforcing biased or misleading concepts toward SSL. Moreover, randomly increasing the labeled pool may not effectively address the posed challenges. In fact, the results stagnate early or are even worse than plain supervised learning. In contrast, we evaluate simple AL heuristics on the introduced challenges and show that active data selection leads to much better generalization performance in these cases. This provides empirical evidence of the benefits of incorporating AL techniques to mitigate the impact of real-world challenges in SSL.

Our main contributions are:

- We provide a thorough literature review on the real-world validity of current evaluation protocols for active and semi-supervised learning. We find that the combination is especially understudied in real-world datasets.
- We explore well-established SSL methods in three real-world challenges and find that confirmation bias in SSL is a problem in all studied challenges and leads to degraded performance.
- We show that, in contrast to random selection, *actively* increasing the labeled pool can mitigate these problems.

2 Related Work

The advantages of AL have been questioned due to the strong performance of methods exploiting knowledge available in unlabeled data [7,11,33].

Given AL aims to increase model performance while decreasing annotation efforts, it is important not to focus on AL in isolation when other training techniques can lead to improvements in model performance. This makes the evaluation of AL challenging [32] as there are many ways to configure AL, and it can be hard to know upfront what works in a real-world scenario.

Our focus is specifically on three realistic data scenarios that can lead SSL to underperform due to confirmation bias.

2.1 Real World Considerations in Machine Learning

The evaluation of the algorithmic progress on a task can be separated into *internal* validity and *external* validity [29]. When benchmark results are internally valid, the improvements caused by an algorithm are valid within the same dataset. However, the overuse of the same test sets in benchmarks can lead to adaptive overfitting where the models and hyperparameters yielding strong performance are reused, and the improvements are not necessarily caused by algorithmic improvements. On the other hand, external validity refers to whether improvements also translate to other datasets for the same task. It has been observed that an excessive emphasis on benchmark performance can result in diminishing returns where increasingly large efforts lead to smaller and smaller performance gains on the actual task [29,45]. To improve the validity of benchmark results, it is important that the datasets used for evaluation reflect the data challenges that occur in real-world scenarios.

Considering data challenges has been a well-studied field in machine learning. Lopez et al. [30] investigate how data intrinsic characteristics in imbalanced datasets affect classification performance and specify six problems that occur in real-world data. Both [42] and [50] also focus on imbalanced data and discuss difficulty factors that deteriorate classification performance. [42] further demonstrates that these factors have a larger impact than the imbalance ratio or the size of the minority class. [14] investigates data irregularities that can lead to a degradation in classification performance. However, to the best of our knowledge studying data challenges in limited labeled scenarios has not yet been well studied [32,35,49].

2.2 Evaluation of AL in the Literature

To get an understanding of the data commonly used for evaluation in limited labels scenarios, we performed a literature overview of the papers published in 13 top-venue conferences[1] within Artificial Intelligence, Machine Learning, Computer Vision, Natural Language Processing and Data Mining between 2018 and 2022. We selected papers for screening if "*active learning*" occurs in the title and abstract, resulting in 392 papers. When screening, we included papers that empirically study the improvement of machine learning models for image

[1] ACL, AAAI, CVPR, ECCV, ECML PKDD, EMNLP, ICCV, ICDM, ICLR, ICML, IJCAI, KDD, and NeurIPS.

classification when expanding the pool of labeled data, as is common in AL papers. Based on this inclusion criteria, we first screened the title and abstracts, and if we could not exclude a study only on the title and abstract, we did a full-text screening. Following this screening process, we identified 51 papers.

We find that 47 (94%) of the studies experimented on at least one benchmark dataset, and 38 (75%) of the studies experiments solely on benchmark datasets[2]. To understand how common it is to evaluate AL in more realistic data scenarios, we count how many papers consider the data challenges BCI, WCI, or BCS or experiments on non-benchmark datasets. We find that 23 (45%) papers consider real-world data challenges or evaluate non-benchmark datasets. The most common data challenge is BCI which 15 (29%) of the papers are considering. As AL can be improved with other training techniques, we look at how many papers combine AL and SSL and find that this is done by 13 (25%) of the papers. However, only 5 (10%) evaluate the performances in realistic scenarios.

3 Learning with Limited Labeled Data

Given an input space $\mathcal{X}$ and a label space $\mathbf{Y}$, we consider the limited labeled scenario where we assume a small labeled pool $\mathcal{X}^l \subset \mathcal{X}$ and a large unlabeled data pool $\mathcal{X}^u = \mathcal{X} \setminus \mathcal{X}^l$. We want to obtain a model $f(x;\theta) \rightarrow \mathbb{R}^C$ where parameters θ map a given input $x \in \mathcal{X}$ to a C-dimensional vector. Supervised learning trains a model on $\mathcal{X}^l$ while SSL utilizes both $\mathcal{X}^l$ and $\mathcal{X}^u$.

3.1 Semi-Supervised Learning (SSL)

Many approaches to leverage both labeled and unlabeled data have been suggested in the literature [13,44]. More recently, the utilization of deep learning in SSL has shown impressive performance, and especially different variants of consistency regularization and pseudo-labeling have been studied [49].

Pseudo-labeling [25] uses the model's prediction on the instances in $\mathcal{X}^u$ to filter highly confident samples and include those with their respective pseudo-label in the next training iteration. Pseudo-labeling is a simple and powerful technique for utilizing $\mathcal{X}^u$. However, a model producing incorrect predictions reuses wrong information in training. This is known as confirmation bias [3] and can greatly impact model performance.

Consistency Regularization [8,41] exploits $\mathcal{X}^u$ by encouraging invariant predictions when the input is perturbated, thereby making the model robust to different perturbed versions of unlabeled data. Perturbations of the data can be obtained by introducing random noise to the input data or utilizing data augmentations [41]. Some methods rely heavily on data augmentations which assume that label-preserving data augmentations are available when applying

[2] We consider benchmark datasets as the well-established MNIST, CIFAR10/100, SVHN, FashionMNIST, STL-10, ImageNet (and Tiny-ImageNet), as well as Caltech-101 and Caltech-256.

such methods in real-world use cases. Using consistency regularization in combination with pseudo-labeling helps improve the generalizability through the perturbed data, which can further enforce the confirmation bias if the model predictions are wrong.

3.2 Active Learning (AL)

AL alternates between querying instances for annotation, and re-training the model $f(x;\theta)$ on the increased labeled pool until an annotation budget is exhausted or a certain performance is reached. The so-called acquisition function of an AL strategy determines which instances in $\mathcal{X}^u$ are most valuable and should be labeled to maximize the labeling efficiency. We use the following taxonomy to distinguish between active acquisition types (Fig. 1).

Instance-Level Acquisition. Each unlabeled instance $x \in \mathcal{X}^u$ is assigned a scoring individually, independent of already selected instances, and enables a final ranking of all unlabeled instances.

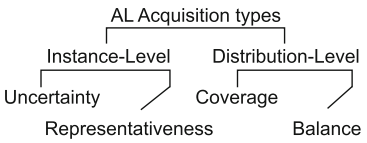


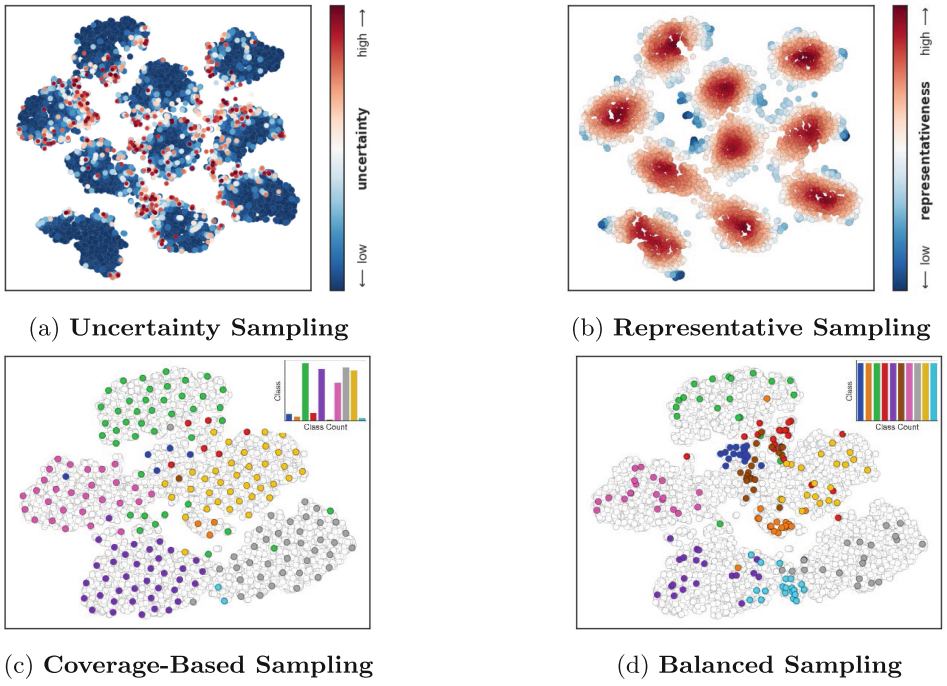
Fig. 1. AL Acquisition Types

Uncertainty sampling aims to query instances carrying the most novel information for the current learner. Popular estimates are least-confidence, min margin, or max entropy selection [40]. These methods usually query near the class boundaries as illustrated in Fig. 2a. The 2D t-SNE visualization of MNIST shows a mapping of margin uncertainty, where red indicates high and blue indicates low uncertainty. Other methods aim to measure model confidence [18] and to distinguish between aleatoric and epistemic uncertainty [34].

Representative sampling assigns higher scores to instances representative of their class or a certain local region. The central idea is not to select instances to eliminate knowledge gaps in the current learning phase but to find instances that have the *highest impact* on most other instances because, e.g., they are representative of a class or they are similar to many other instances.

One way to define representativeness is to measure centrality, for instance, by exploiting a preceding partitioning and selecting the most central instance of each partitioning [37,55]. Another estimate for representativeness is density, i.e., how many (similar) instances are in the near surrounding of a data point [15, 47]. In Fig. 2b, the colors indicate the negative local outlier score [10] mapped onto the 2D representation of MNIST, which is here used as an indicator for representativeness. A representativeness selection strategy would favor instances in the denser red regions in the center of the clusters.

Distribution-Level Acquisition. In contrast to instance-level, distribution-level acquisition refers to selection strategies that do not consider individual scores for each instance but strive to optimize the distribution of all selected



(a) **Uncertainty Sampling** (b) **Representative Sampling**

(c) **Coverage-Based Sampling** (d) **Balanced Sampling**

Fig. 2. Exemplary illustration of different acquisition types.

instances. A clear ranking is usually not possible because the worthiness of the next best instance depends on which instance(s) is (have been) selected before.

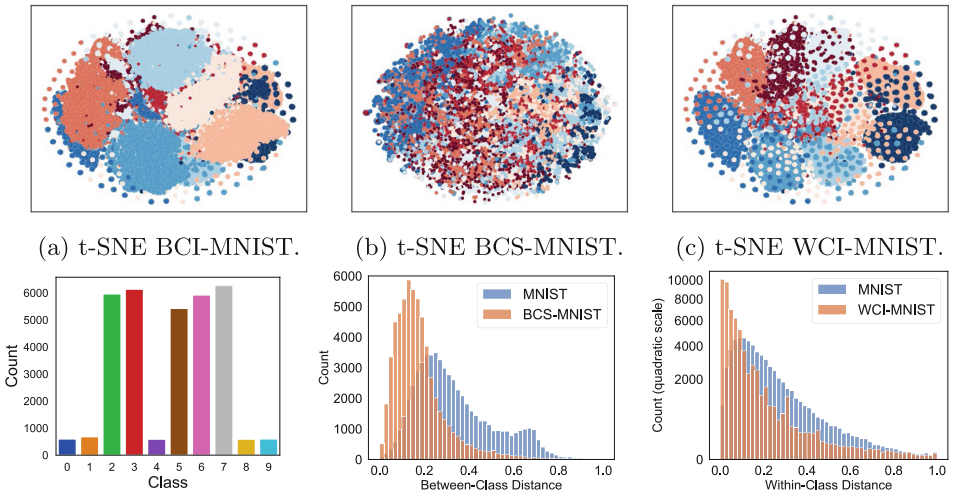
Coverage-based sampling, sometimes referred to as diversity sampling, aims to cover the given data space to avoid overlap of information best. The goal is to select as diverse instances as possible to maximize the richness of information in the labeled dataset. The most prominent method of this category is k-Center-Greedy which maximizes the distance in the feature space between the queried and the labeled instances [39]. Coverage, or diversity, is a popular companion in hybrid approaches to assist batch-selection acquisitions [4,23,37].

Balanced sampling aims to balance the number of samples per class and is especially suited for imbalanced datasets. This subtype is often combined with other acquisition types, as it does not necessarily select the most valuable instances on its own [1,6,16]. Figure 2c depicts coverage sampling on an imbalanced version of MNIST where the data space is evenly covered. In contrast, Fig. 2d shows balanced sampling where the selected class counts are uniformly distributed.

There is an abundance of hybrid methods combining two or more of the described concepts [4,17,23,37,52]. However, in this work, we focus on highlighting the potential of AL in general and only consider disjoint baseline methods from each category. For an overview of deep AL methods, we refer to [38,51,53].

4 Three Real-World Data Challenges

In the following, we introduce three realistic data challenges. We then present three datasets that implement these challenges on the well-known MNIST task, which we later analyze in our experiments.



(a) t-SNE BCI-MNIST. (b) t-SNE BCS-MNIST. (c) t-SNE WCI-MNIST.

(d) Class distribution BCI-MNIST. (e) Between-class distances of BCS-MNIST vs. MNIST. (f) Within-class distances of WCI-MNIST vs. MNIST.

Fig. 3. Three realistic challenges (BCI, BCS, WCI) demonstrated on MNIST.

4.1 Between-Class Imbalance (BCI)

Among our challenges, Between-Class Imbalance (BCI) is the most considered in the literature and is a well-known challenge for supervised machine learning models. Imbalanced class distributions pose a problem for SSL methods where unlabeled data is often assumed to be distributed similarly to the labeled data and balanced class distributions. BCI can pose a problem for SSL when there is a mismatch between the labeled and unlabeled class distributions [35] or simply because some classes are generally underrepresented in both the unlabeled and labeled pool [21]. However, class distributions in real-world datasets often follow a long-tail distribution. While class imbalance has been studied for both AL and SSL separately, an open question remains regarding how to leverage AL techniques to address the negative effects of class imbalance in SSL.

4.2 Between-Class Similarity (BCS)

Another category of data challenges is Between-Class Similarity (BCS). In real-world datasets, the boundaries between classes can be hard to draw. Instances

within the same class can differ widely, and conversely, instances from different classes can be very similar. High within-class diversity and similarity between classes happens naturally in many image classification tasks, e.g., diatom or plankton classification [46] or within histopathology [43].

Datasets with BCS are a challenge for techniques that rely on unlabeled data for model training, since that contradicts the basic assumptions of SSL. For instance, according to [12], Fixmatch exacerbates confusion when instances across classes are similar. The degree of BCS determines whether it is advantageous to sample from class boundaries while the classes can still be differentiated or to prioritize selecting representative instances without ambiguity in the class assignment. Consequently, this challenge presents an opportunity for AL to identify and label such samples. This problem does not only occur on hard-to-solve tasks with high aleatoric uncertainty. Ambiguous label information can also occur due to the labeling procedure e.g. when data is labeled by multiple annotators which can introduce labeling variations [36], or when labels are acquired automatifcally [27,45]. Label noise can have a large impact on SSL as the model is more prone to confirm learned mistakes leading to confirmation bias [28].

Common usage of SSL methods for noisily labeled data is to simply remove noisy labels and continue training with conventional SSL [2]. Alternatively, some algorithms distinguish between cleanly labeled, noisily labeled, and unlabeled data enabling the usage of a massive amount of unlabeled and noisy data under the supervision of a few cleanly annotated data. However, directly coupling the data selection actively to the training can be an easy and thus attractive solution to directly account for label noise or ambiguous class labels without post-processing wrong labels or complex algorithms and wasted labeling efforts.

4.3 Within-Class Imbalance (WCI)

Imbalance is not only a problem across classes but also within classes [20,22]. Although instances might belong to the same class, they can have a high variability due to, e.g., pose, lighting, viewpoint, etc. To obtain a model with the most discriminative capabilities, it must be exposed to the variation within the class.

Within-class imbalance (WCI) occurs in many real-world problems. In medical imaging, subgroups such as race or gender exist within classes and are often imbalanced [48]. Similarly, in microscopic classification, the images might have different viewpoints forming diverse [46] and imbalanced [26] subclusters. In automatic defect detection for manufacturing systems, the different types of defects are often all grouped into the same superordinate class and can be very diverse and imbalanced [20]. It has also been shown that repetition of subclasses containing highly similar samples occurs in commonly used image classification benchmark datasets [9], leading to some subclasses that contain redundant semantic information being overrepresented.

WCI, similar to BCI, leads to the minority subclass being exposed less in the optimization process and contributing less to the final model. This leads

to a bias towards the majority subclass and suboptimal performance of the learned model. The difference between WCI and BCI lies in the lack of subclass labels. This deems common solutions for BCI that rely on sampling or cost-aware learning irrelevant for WCI as they rely on class labels.

4.4 Challenge Construction

To gain insights into how SSL and AL perform when the data challenges are present, we construct three datasets based on MNIST to reflect the challenges. We intentionally use MNIST as we can isolate any effects of the data challenges instead of the potential complexity of the learning task.

BCI-MNIST. We construct a between-class imbalanced version of MNIST (BCI-MNIST), where 50% of the classes only contain approximately 10% of the instances. Figure 3a and Fig. 3d illustrate the distribution of the imbalanced version in a 2D t-SNE plot, and a barplot respectively.

BCS-MNIST. Figure 3b shows a 2D t-SNE-plot of an ambiguous version of MNIST proposed in [34]. The dataset consists of normal MNIST and Ambiguous MNIST, containing a large fraction of ambiguous instances with questionable labels, thus increasing the class overlap. Figure 3e shows the similarities of each instance to all instances not belonging to the same class. Compared to the original MNIST, the similarity among instances across classes is much higher. In our experiment, we select 5% of instances from the original MNIST dataset and 95% of instances from Ambiguous MNIST and refer to it as BCS-MNIST.

WCI-MNIST. The WCI version of MNIST is constructed with the following procedure: (1) For each class, we create a sub-clustering using the K-means algorithm on the original input features with $k = 300$. (2) For each constructed within-class cluster, we select one instance as the underrepresented subclass except for one majority subclass and remove the remaining instances. (3) We copy all the instances within the majority subclass multiple times to restore the original training set size and randomly add Gaussian noise to create slightly different versions. The 2D t-SNE representation is shown in Fig. 3c. While the class boundaries are sharper than in Fig. 3b, many subgroups within each class are spread around all the data space. Figure 3f shows the summed distance of each instance to the remaining instances of their respective class for MNIST (blue) and our constructed WCI-MNIST (orange). WCI-MNIST has more highly similar instances, and the number of medium distances is much smaller, resulting in a non-linear decrease in intra-class distances and higher within-class imbalance.

5 Experiments

In this section, we evaluate established SSL methods combined with simple AL heuristics on the previously described challenges that ostensibly occur in real-world scenarios. We use the following experimental setup[3].

Backbone and Training. For all experiments, we use a LeNet [24] as backbone as is commonly used for digit recognition. We do not use a validation set as proposed in [35] since it is unrealistic to assume having a validation set when there is hardly any label information. Instead, we train the model for 50 epochs and use early stopping if the model reaches 99% training accuracy following [4]. The learning rate is set to 0.001, and we do not use any scheduler.

SSL. We include pseudo-labeling [25] (PL) with a threshold of 0.95 as baseline without consistency regularization. We further include Fixmatch [41] as it is a well-established consistency regularization technique and Flexmatch [54] as a strong method tackling confirmation bias [49]. Furthermore, we report results on a plain supervised baseline (SPV).

Evaluation. We report average test accuracies over five random seeds for different labeling budgets. Initially, we select 20 labeled instances randomly. Then, we increase the labeled pool to budgets of 50, 100, 150, 200, and 250 labels.

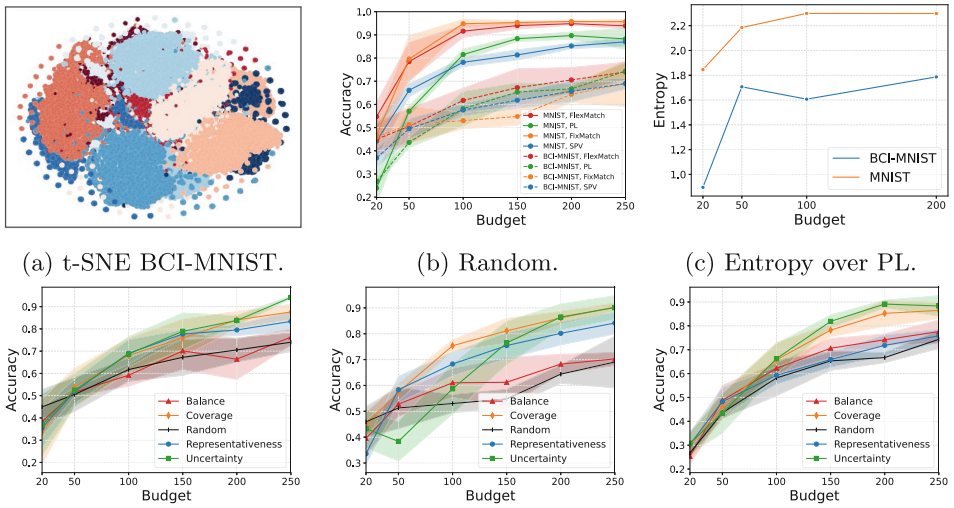
AL. We choose one representative from each of the described categories in Sect. 3.2 to better assess the strength and weaknesses of each acquisition type. We use margin uncertainty [40] as an uncertainty baseline. For representativeness, we perform k-means clustering on the latent features and select the instance closest to the centroid similar to [19,37]. As a coverage-based technique, we include the k-Center-Greedy method proposed in [39]. For balanced sampling, we create a baseline that selects instances proportional to the sum of inverse class frequencies in the current labeled set and the corresponding prediction probability. Though this might not be a strong AL baseline in general, we expect to see a slight improvement in the BCI challenge.

Datasets. We use the three constructed datasets explained in Sect. 4.4 to form the unlabeled pool, as well as the original MNIST. For testing, we use the original MNIST test set to ensure comparable results.

5.1 Experiment "BCI-MNIST"

Figure 4b depicts the average accuracy of supervised learning (SPV, blue), pseudo-labeling (PL, green), Fixmatch (orange), and Flexmatch (red) for different labeling budgets on MNIST (solid) and BCI-MNIST (dashed) with random

[3] See also https://github.com/lmu-dbs/HOCOBIS-AL.



(a) t-SNE BCI-MNIST. (b) Random. (c) Entropy over PL.

(d) AL Flexmatch. (e) AL Fixmatch. (f) AL PL.

Fig. 4. (a) t-SNE of BCI-MNIST challenge. (b) Average test accuracy of all learners evaluated on BCI-MNIST (dashed) and MNIST (solid). In (d), we observe for BCI-MNIST, the entropy over the selected pseudo-labels falling over the threshold for each class is much smaller. This indicates that the distribution of selected pseudo-labels for BCI-MNIST is more imbalanced, repeatedly confirming the imbalance. (d), (e) and (f) show the selected AL curves for Flexmatch, Fixmatch, and PL compared to random sampling (black). (Color figure online)

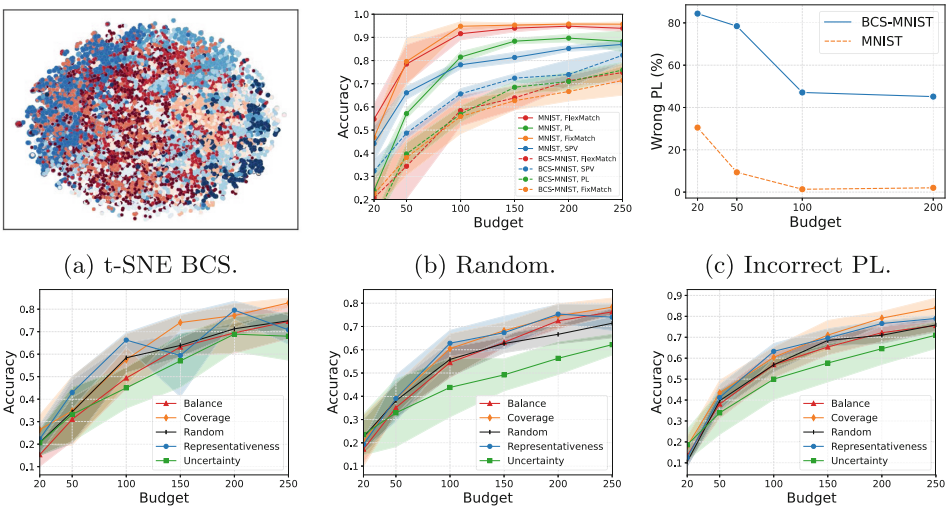
labeling. BCI has a severe impact on the performance of all learners. However, Fixmatch is affected most and even performs worse than SPV. Since training takes much longer for SSL, [35] argue that these methods should clearly outperform SPV to be considered useful. This is no longer true in our experiment, even on a simple task like MNIST. Figure 4c visualizes the entropy over the number of pseudo-labeled instances per class that Fixmatch would choose for training for BCI-MNIST (blue) and MNIST (orange). On MNIST the entropy is much higher, indicating that the distribution over the classes is more uniformly distributed. The problem is not only that the selected labeled data is imbalanced, but the chosen pseudo-labels repeatedly *confirm* the imbalance, such that the underrepresented classes get even more underrepresented.

However, the AL curves in Figs. 4e and 4f demonstrate that the choice of data selection methods has a substantial impact on the performance of each learner. Fixmatch largely benefits from coverage-based sampling, representative sampling, and uncertainty sampling for later iterations. For the final budget of 250, the gap between coverage and uncertainty acquisition and random selection is around 20%. PL and Flexmatch also greatly benefit from coverage and uncertainty sampling. Coverage sampling is even able to restore the accuracy achieved on MNIST with random sampling, yielding 88.3% for PL and 94.1% for Flexmatch. Interestingly, balanced sampling is not among the best active methods.

Even though the performance is slightly better than random sampling, the other methods are much stronger. This is probably because balanced sampling without the combination of any other method does select less informative and more redundant information.

5.2 Experiment "BCS-MNIST"

Figure 5b illustrates the learning curves for the learners on MNIST and BCS-MNIST. All methods suffer, but Fixmatch clearly suffers the most and is no longer better than plain supervision. In this scenario, there is no additional benefit of exploiting the unlabeled pool, but the training times are multiple times larger. Figure 5c illustrates the fraction of wrong pseudo-labels surpassing the threshold when training Fixmatch on MNIST (orange) and BCS-MNIST (blue). Over 40% of the predicted pseudo-labels over the threshold are wrong up to a labeling budget of 200 instances. Figures 5e and 5f denote the learning curves of Flexmatch, Fixmatch, and PL when increasing the labeled pool actively. Notably, all learners benefit from coverage-based sampling. Representative sampling is beneficial for Fixmatch. This method promotes instances representative of a certain class or region and probably selects instances that are less ambiguous for training. However, as expected, employing the uncertainty baseline in this context proves to be a poor choice. The strategy lacks the ability to differentiate



(a) t-SNE BCS. (b) Random. (c) Incorrect PL.

(d) AL Flexmatch. (e) AL Fixmatch. (f) AL PL.

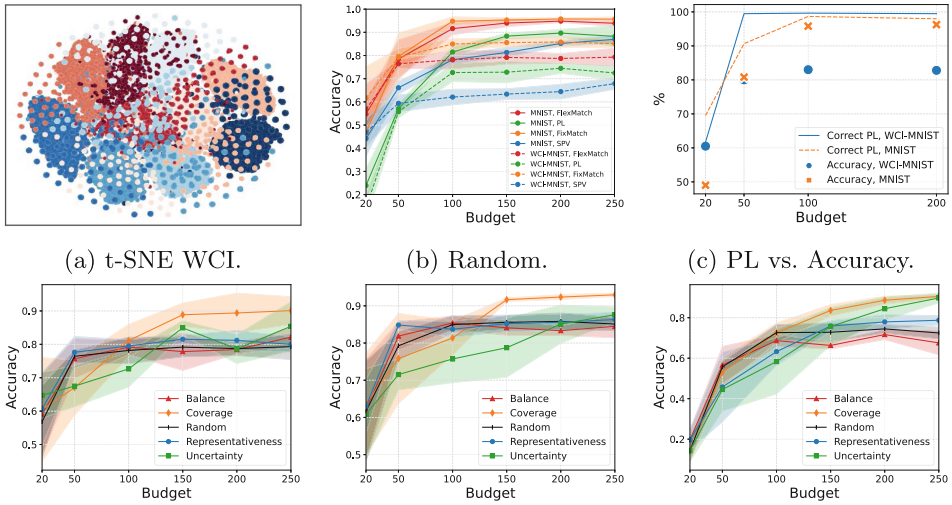
Fig. 5. (b) Average test accuracy of all learners with random selection for BCS-MNIST (dashed) and MNIST (solid). (c) shows the amount of wrongly predicted pseudo-labels falling over the threshold using Fixmatch for BCS-MNIST is much larger than for MNIST. (d), (e) and (f) show the AL curves for Flexmatch, Fixmatch, and PL compared to random sampling (black). (Color figure online)

between aleatoric and epistemic uncertainty, leading to the selection of many ambiguous instances, further misleading the training.

5.3 Experiment "WCI-MNIST"

Figure 6b shows that for WCI-MNIST, the accuracy of all learners stagnates around 10% to 15% earlier compared to MNIST. Using random sampling does not find the underrepresented diverse instances, and only the same concepts are entrenched and further confirmed over the training procedure. Even though the correctness ratio of the pseudo-labels surpassing the threshold using Fixmatch is larger for WCI-MNIST than for MNIST, the achieved mean test accuracy stops at roughly 82% (see Fig. 6c).

However, using AL, we can find more diverse and valuable instances than the already known concepts and reach a better final accuracy overall for SSL (see Figs. 6d to 6f). Especially coverage-based sampling seems to be a viable choice. For PL, the final average accuracy using uncertainty-based and coverage-based sampling on WCI-MNIST is even equally good as the performance on the original MNIST using random sampling. In the early stages, uncertainty sampling is the worst method probably because it lacks diversity aspects, and the predictions in early iterations might not be very reliable. However, for the final budget, uncertainty sampling matches or surpasses most other methods. The representative



(a) t-SNE WCI. (b) Random. (c) PL vs. Accuracy.

(d) AL Flexmatch. (e) AL Fixmatch. (f) AL PL.

Fig. 6. (b) Average test accuracy of all learners for WCI-MNIST (dashed) and MNIST (solid). In (d), we observe that even though more pseudo-labels are chosen correctly using Fixmatch for WCI-MNIST (blue line), the test accuracy is much smaller (blue markers) than for MNIST (orange) because only the same concepts are confirmed over and over again. (d), (e) and (f) show the selected AL curves for Flexmatch, Fixmatch, and PL compared to random sampling (black). (Color figure online)

baseline focuses on instances that are most central in clusters, probably resulting in only selecting the already known and easy-to-classify concepts lacking novel information and does not outperform random sampling in most situations.

6 Key Findings

Table 1 shows the average test accuracies of SPV, Fixmatch, PL, and Flexmatch on BCI-MNIST, BCS-MNIST, and WCI-MNIST for all AL heuristics compared to random sampling, where bold and red numbers indicate best- and worst-performing methods per column respectively for 50 and 250 labeled instances. Our key findings can be summarized as follows:

- For all introduced data challenges, the SSL methods suffer from confirmation bias. There is no consistent winner among all query strategies, but random sampling is never the best query method for the SSL methods when faced with BCS, WCI, and BCI. This provides empirical evidence that AL is a useful tool to overcome confirmation bias in SSL.
- In the early stages, representative sampling is often beneficial. In contrast, uncertainty sampling usually performs better in later iterations where model predictions are more reliable. As expected, uncertainty sampling is not a good choice for BCS since it queries from overlapping, confusing regions.
- Coverage sampling is often the best strategy for SSL methods. We assume that is because more diverse queried instances bring in new aspects to the data, and the easier concepts can already be learned by pseudo-labeling and consistency regularization.
- Our balance baseline often performs on par with random selection. However, for the BCI challenge, it yields slightly better results. We conclude that it should mainly be used in combination with other selection heuristics.
- Overall, the most challenging dataset for SSL and AL is BCS-MNIST. By using AL, we can mitigate confirmation bias more effectively for the challenges BCI and WCI compared to random sampling.

Table 1. Average test accuracy for SPV, Fixmatch, PL, and Flexmatch for BCI-MNIST, BCS-MNIST, and WCI-MNIST for all sampling methods and budgets 50 and 250 (L). **Bold** and red numbers indicate column-wise best- and worst-performing methods, respectively.

	Supervised						Fixmatch						Pseudo-Labeling						Flexmatch					
	BCI		BCS		WCI		BCI		BCS		WCI		BCI		BCS		WCI		BCI		BCS		WCI	
L	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250
Rnd	49.4	68.9	48.6	82.3	59.3	67.8	51.2	69.0	38.1	71.3	79.2	85.1	43.6	74.3	39.6	75.9	55.7	72.4	50.5	74.0	34.3	74.8	76.4	79.3
Unc	52.0	**82.5**	53.0	80.5	47.8	70.0	38.3	90.1	32.8	62.1	71.5	87.6	43.3	**88.3**	33.8	70.9	44.5	89.7	52.5	**94.1**	33.2	67.9	67.4	85.3
Cov	47.9	79.2	**55.2**	**83.8**	**63.0**	**87.6**	57.4	**90.3**	38.9	**78.2**	75.8	**92.9**	46.2	86.4	**43.4**	**84.0**	53.0	**90.5**	**53.8**	87.6	34.7	**82.8**	67.0	**90.1**
Bal	48.2	68.7	50.6	78.5	58.0	64.1	52.7	70.2	35.0	76.2	81.8	84.4	**48.8**	77.6	38.0	75.5	56.5	67.5	51.5	76.3	31.0	74.5	75.7	82.1
Rep	**54.7**	66.8	47.7	75.8	61.6	66.1	**58.3**	84.1	39.0	73.9	**84.8**	86.3	48.2	75.8	41.2	78.9	45.7	78.7	51.8	83.3	**42.9**	70.8	**77.6**	79.9

7 Conclusion

In this work, we study the real-world transferability of critique points on the combination of SSL and AL on benchmark datasets. Our experiments show that AL is a useful tool to overcome confirmation bias in various real-world challenges. However, it is not trivial to determine which AL method is most suitable in a real-world scenario. This study is limited to providing insights into confirmation bias in SSL when confronted with between-class imbalance, between-class similarity, and within-class similarity and the potential of simple AL heuristics. In the future, we intend to extend our experiments to a broader range of datasets, with a strong focus on real-world examples. Moreover, we aim to include existing hybrid AL methods in our evaluation and to design a robust active semi-supervised method capable of consistently overcoming confirmation bias in SSL on diverse challenges.

Acknowledgements. This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy through the Center for Analytics Data Applications (ADA-Center) within the framework of BAYERN DIGITAL II (20-3410-2-9-8) as well as the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A.

References

1. Aggarwal, U., Popescu, A., Hudelot, C.: Active learning for imbalanced datasets. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1428–1437 (2020)
2. Algan, G., Ulusoy, I.: Image classification with deep learning in the presence of noisy labels: A survey. Knowl.-Based Syst. **215**, 106771 (2021)
3. Arazo, E., Ortego, D., Albert, P., O'Connor, N.E., McGuinness, K.: Pseudo-labeling and confirmation bias in deep semi-supervised learning. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2020). https://doi.org/10.1109/IJCNN48605.2020.9207304
4. Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020. OpenReview.net (2020). https://openreview.net/forum?id=ryghZJBKPS
5. Beck, N., Sivasubramanian, D., Dani, A., Ramakrishnan, G., Iyer, R.: Effective evaluation of deep active learning on image classification tasks. arXiv preprint arXiv:2106.15324 (2021)
6. Bengar, J.Z., van de Weijer, J., Fuentes, L.L., Raducanu, B.: Class-balanced active learning for image classification. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1536–1545 (2022)
7. Bengar, J.Z., van de Weijer, J., Twardowski, B., Raducanu, B.: Reducing label effort: self-supervised meets active learning. In: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), pp. 1631–1639. IEEE Computer Society, Los Alamitos (2021). https://doi.org/10.1109/ICCVW54120.2021.00188. https://doi.ieeecomputersociety.org/10.1109/ICCVW54120.2021.00188

8. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: a holistic approach to semi-supervised learning. Adv. Neural Inf. Process. Syst. **32**, 1–11 (2019)
9. Birodkar, V., Mobahi, H., Bengio, S.: Semantic redundancies in image-classification datasets: the 10% you don't need. arXiv preprint arXiv:1901.11409 (2019)
10. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 93–104 (2000)
11. Chan, Y.-C., Li, M., Oymak, S.: On the marginal benefit of active learning: Does self-supervision eat its cake? In: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3455–3459 (2021). https://doi.org/10.1109/ICASSP39728.2021.9414665
12. Chang, H., Xie, G., Yu, J., Ling, Q., Gao, F., Yu, Y.: A viable framework for semi-supervised learning on realistic dataset. In: Machine Learning, pp. 1–23 (2022)
13. Chapelle, O., Scholkopf, B., Zien, A.: Semi-supervised learning. IEEE Trans. Neural Netw. **20**(3), 542–542 (2009)
14. Das, S., Datta, S., Chaudhuri, B.B.: Handling data irregularities in classification: foundations, trends, and future challenges. Pattern Recogn. **81**, 674–693 (2018)
15. Donmez, P., Carbonell, J.G., Bennett, P.N.: Dual strategy active learning. In: Kok, J.N., Koronacki, J., Mantaras, R.L., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 116–127. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74958-5_14
16. Ertekin, S., Huang, J., Bottou, L., Giles, L.: Learning on the border: active learning in imbalanced data classification. In: Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, pp. 127–136 (2007)
17. Fu, B., Cao, Z., Wang, J., Long, M.: Transferable query selection for active domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7272–7281 (2021)
18. Gal, Y., Islam, R., Ghahramani, Z.: Deep bayesian active learning with image data. In: International Conference on Machine Learning, pp. 1183–1192. PMLR (2017)
19. Gilhuber, S., Berrendorf, M., Ma, Y., Seidl, T.: Accelerating diversity sampling for deep active learning by low-dimensional representations. In: Kottke, D., Krempl, G., Holzinger, A., Hammer, B. (eds.) Proceedings of the Workshop on Interactive Adaptive Learning co-located with European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2022), Grenoble, France, 23 September 2022. CEUR Workshop Proceedings, vol. 3259, pp. 43–48. CEUR-WS.org (2022). https://ceur-ws.org/Vol-3259/ialatecml_paper4.pdf
20. Huang, L., Lin, K.C.J., Tseng, Y.C.: Resolving intra-class imbalance for gan-based image augmentation. In: 2019 IEEE International Conference on Multimedia and Expo (ICME), pp. 970–975 (2019). https://doi.org/10.1109/ICME.2019.00171
21. Hyun, M., Jeong, J., Kwak, N.: Class-imbalanced semi-supervised learning. arXiv preprint arXiv:2002.06815 (2020)
22. Japkowicz, N.: Concept-learning in the presence of *between-class* and *within-class* imbalances. In: Stroulia, E., Matwin, S. (eds.) AI 2001. LNCS (LNAI), vol. 2056, pp. 67–77. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45153-6_7
23. Kirsch, A., Van Amersfoort, J., Gal, Y.: Batchbald: efficient and diverse batch acquisition for deep bayesian active learning. Adv. Neural Inf. Process. Syst. **32**, 1–12 (2019)
24. LeCun, Y., et al.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**(4), 541–551 (1989)

25. Lee, D.H., et al.: Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning, ICML, vol. 3, p. 896 (2013)
26. Lee, H., Park, M., Kim, J.: Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 3713–3717 (2016). https://doi.org/10.1109/ICIP.2016.7533053
27. Li, J., et al.: Learning from large-scale noisy web data with ubiquitous reweighting for image classification. IEEE Trans. Pattern Anal. Mach. Intell. **43**(5), 1808–1814 (2019)
28. Li, J., Socher, R., Hoi, S.C.: Dividemix: learning with noisy labels as semi-supervised learning. In: International Conference on Learning Representations (2020). https://openreview.net/forum?id=HJgExaVtwr
29. Liao, T., Taori, R., Raji, I.D., Schmidt, L.: Are we learning yet? a meta review of evaluation failures across machine learning. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2) (2021). https://openreview.net/forum?id=mPducS1MsEK
30. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. Inf. Sci. **250**, 113–141 (2013)
31. Lowell, D., Lipton, Z.C., Wallace, B.C.: Practical obstacles to deploying active learning. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 21–30 (2019)
32. Lüth, C.T., Bungert, T.J., Klein, L., Jaeger, P.F.: Toward realistic evaluation of deep active learning algorithms in image classification (2023)
33. Mittal, S., Tatarchenko, M., Çiçek, Ö., Brox, T.: Parting with illusions about deep active learning. ArXiv abs/1912.05361 (2019)
34. Mukhoti, J., Kirsch, A., van Amersfoort, J., Torr, P.H.S., Gal, Y.: Deep deterministic uncertainty: A new simple baseline. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 24384–24394 (2023). https://doi.org/10.1109/CVPR52729.2023.02336
35. Oliver, A., Odena, A., Raffel, C.A., Cubuk, E.D., Goodfellow, I.: Realistic evaluation of deep semi-supervised learning algorithms. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 31. Curran Associates, Inc. (2018). https://proceedings.neurips.cc/paper_files/paper/2018/file/c1fea270c48e8079d8ddf7d06d26ab52-Paper.pdf
36. Plank, B.: The "problem" of human label variation: On ground truth in data, modeling and evaluation. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Abu Dhabi (2022)
37. Prabhu, V., Chandrasekaran, A., Saenko, K., Hoffman, J.: Active domain adaptation via clustering uncertainty-weighted embeddings. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8505–8514 (2021)
38. Ren, P., et al.: A survey of deep active learning. ACM Comput. Surv. (CSUR) **54**(9), 1–40 (2021)
39. Sener, O., Savarese, S.: Active learning for convolutional neural networks: a core-set approach. In: International Conference on Learning Representations (2018)
40. Settles, B.: Active learning literature survey (2009)

41. Sohn, K., et al.: FixMatch: simplifying semi-supervised learning with consistency and confidence. In: Advances in Neural Information Processing Systems (2020)
42. Stefanowski, J.: Dealing with data difficulty factors while learning from imbalanced data. In: Matwin, S., Mielniczuk, J. (eds.) Challenges in Computational Statistics and Data Mining. SCI, vol. 605, pp. 333–363. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-18781-5_17
43. Su, L., Liu, Y., Wang, M., Li, A.: Semi-hic: a novel semi-supervised deep learning method for histopathological image classification. Comput. Biol. Med. **137**, 104788 (2021)
44. Van Engelen, J.E., Hoos, H.H.: A survey on semi-supervised learning. Mach. Learn. **109**(2), 373–440 (2020)
45. Varoquaux, G., Cheplygina, V.: Machine learning for medical imaging: methodological failures and recommendations for the future. NPJ Dig. Med. **5**(1), 1–8 (2022)
46. Venkataramanan, A., Laviale, M., Figus, C., Usseglio-Polatera, P., Pradalier, C.: Tackling inter-class similarity and intra-class variance for microscopic image-based classification. In: Vincze, M., Patten, T., Christensen, H.I., Nalpantidis, L., Liu, M. (eds.) ICVS 2021. LNCS, vol. 12899, pp. 93–103. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87156-7_8
47. Wang, M., Min, F., Zhang, Z.H., Wu, Y.X.: Active learning through density clustering. Expert Syst. Appl. **85**, 305–317 (2017)
48. Wang, Q.: Wgan-based synthetic minority over-sampling technique: improving semantic fine-grained classification for lung nodules in ct images. IEEE Access **7**, 18450–18463 (2019). https://doi.org/10.1109/ACCESS.2019.2896409
49. Wang, Y., et al.: Usb: a unified semi-supervised learning benchmark for classification. In: Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2022). https://doi.org/10.48550/ARXIV.2208.07204. https://arxiv.org/abs/2208.07204
50. Wojciechowski, S., Wilk, S.: Difficulty factors and preprocessing in imbalanced data sets: an experimental study on artificial data. Found. Comput. Decis. Sci. **42**(2), 149–176 (2017). https://doi.org/10.1515/fcds-2017-0007
51. Wu, M., Li, C., Yao, Z.: Deep active learning for computer vision tasks: methodologies, applications, and challenges. Appl. Sci. **12**(16), 8103 (2022)
52. Xie, B., Yuan, L., Li, S., Liu, C.H., Cheng, X.: Towards fewer annotations: active learning via region impurity and prediction uncertainty for domain adaptive semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8068–8078 (2022)
53. Zhan, X., Wang, Q., Huang, K.H., Xiong, H., Dou, D., Chan, A.B.: A comparative survey of deep active learning. arXiv preprint arXiv:2203.13450 (2022)
54. Zhang, B., et al.: Flexmatch: boosting semi-supervised learning with curriculum pseudo labeling. Adv. Neural Inf. Process. Syst. **34**, 18408–18419 (2021)
55. Zhdanov, F.: Diverse mini-batch active learning. arXiv preprint arXiv:1901.05954 (2019)

Appendix to "How To Overcome Confirmation Bias in Semi-Supervised Image Classification By Active Learning"

Sandra Gilhuber*[1,2], Rasmus Hvingelby*[3], Mang Ling Ada Fok[3], and Thomas Seidl[1,2,3]

[1] LMU Munich, Germany {gilhuber,seidl}@dbs.ifi.lmu.de
[2] Munich Center for Machine Learning (MCML), Germany
[3] Fraunhofer IIS, Germany rasmus.hvingelby@iis.fraunhofer.de

A Literature Overview

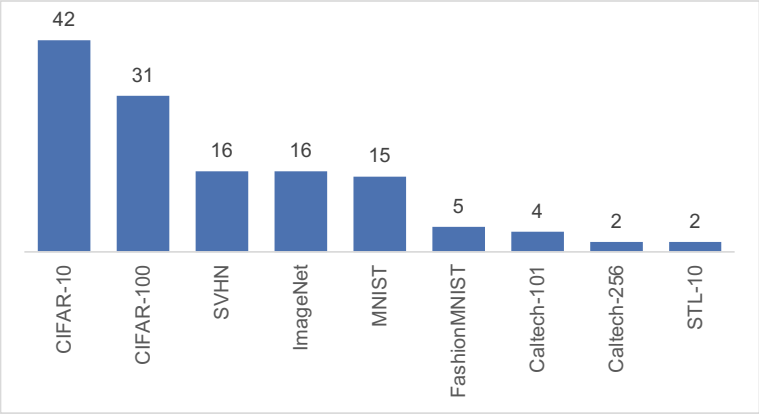


Fig. A.1: The number of papers experimenting on the well-established datasets

Paper	Real-World Considerations	BCI	WCI	BCS	SSL
Kothawade et al. [9]	Imbalance or rare classes, out-of-distribution data, redundancy in the unlabeled set	✓	✓		
Park et al. [15]	Open-set noise				
Elenter et al. [4]	Dataset redundancy in STL-10		✓		
Kirsch et al. [8]	Repetition in MNIST		✓		
Liang et al. [10]	Incorporation with natural language explanation			✓	
Hacohen et al. [6]	An imbalanced subset of CIFAR-10	✓			✓
Zhang et al. [22]	Extreme class imbalance	✓			
Beluch et al. [1]	Highly class-imbalanced diabetic retinopathy dataset (in medical diagnosis)	✓			
Ning et al. [14]	Open-set annotation problem				
Munjal et al. [12]	Class imbalance	✓			
Zhang et al. [23]	Poor data utilization and missing informative sample in medical data				✓
Choi et al. [2]	The heavily imbalanced NEU dataset	✓			
Zhang et al. [20]	Class imbalance in Caltech-101	✓			
Gudovskiy et al. [5]	Biased class imbalance	✓			✓
Wang et al. [18]	Imbalanced data	✓			
Ning et al. [13]	Unexpected noise			✓	
Sinha et al. [17]	Noisy data caused by an inaccurate oracle			✓	✓
Mullapudi et al. [11]	Imbalanced data	✓	✓		✓
Du et al. [3]	Class distribution mismatch				
Yi et al. [19]	Imbalanced data, cold-start problem	✓			
Kim & Shin [7]	Redundancy and highly similar samples		✓		
Shao et al. [16]	Highly imbalanced classes and cold-start problem	✓			
Zhang et al. [21]	Class imbalance	✓			

Table A.1: Overview of studies evaluated in realistic scenarios. BCI is between-class imbalance, WCI is within-class imbalance and BCS is between-class similarity. SSL denotes if a study combines AL with SSL.

References

1. Beluch, W.H., Genewein, T., Nürnberger, A., Köhler, J.M.: The power of ensembles for active learning in image classification. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 9368–9377 (2018)
2. Choi, J., Yi, K.M., Kim, J., Choo, J., Kim, B., Chang, J.Y., Gwon, Y., Chang, H.J.: Vab-al: Incorporating class imbalance and difficulty with variational bayes for active learning. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 6745–6754 (2020)
3. Du, P., Zhao, S., Chen, H., Chai, S., Chen, H., Li, C.: Contrastive coding for active learning under class distribution mismatch. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 8907–8916 (2021)
4. Elenter, J., Naderializadeh, N., Ribeiro, A.: A lagrangian duality approach to active learning. ArXiv **abs/2202.04108** (2022)
5. Gudovskiy, D.A., Hodgkinson, A., Yamaguchi, T., Tsukizawa, S.: Deep active learning for biased datasets via fisher kernel self-supervision. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 9038–9046 (2020)
6. Hacohen, G., Dekel, A., Weinshall, D.: Active learning on a budget: Opposite strategies suit high and low budgets. ArXiv **abs/2202.02794** (2022)
7. Kim, Y., Shin, B.: In defense of core-set: A density-aware core-set selection for active learning. Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2022)
8. Kirsch, A., van Amersfoort, J.R., Gal, Y.: Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In: Neural Information Processing Systems (2019)
9. Kothawade, S., Beck, N., Killamsetty, K., Iyer, R.K.: Similar: Submodular information measures based active learning in realistic scenarios. ArXiv **abs/2107.00717** (2021)
10. Liang, W., Zou, J.Y., Yu, Z.: Alice: Active learning with contrastive natural language explanations. ArXiv **abs/2009.10259** (2020)
11. Mullapudi, R.T., Poms, F., Mark, W.R., Ramanan, D., Fatahalian, K.: Learning rare category classifiers on a tight labeling budget. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 8403–8412 (2021)
12. Munjal, P., Hayat, N., Hayat, M., Sourati, J., Khan, S.: Towards robust and reproducible active learning using neural networks. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 223–232 (2020)
13. Ning, K.P., Tao, L., Chen, S., Huang, S.J.: Improving model robustness by adaptively correcting perturbation levels with active queries. ArXiv **abs/2103.14824** (2021)
14. Ning, K.P., Zhao, X., Li, Y., Huang, S.J.: Active learning for open-set annotation. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 41–49 (2022)
15. Park, D., Shin, Y., Bang, J., Lee, Y., Song, H., Lee, J.G.: Meta-query-net: Resolving purity-informativeness dilemma in open-set active learning. ArXiv **abs/2210.07805** (2022)
16. Shao, J., Wang, Q., Liu, F.: Learning to sample: An active learning framework. 2019 IEEE International Conference on Data Mining (ICDM) pp. 538–547 (2019)
17. Sinha, S., Ebrahimi, S., Darrell, T.: Variational adversarial active learning. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 5971–5980 (2019)

18. Wang, T., Li, X., Yang, P., Hu, G., Zeng, X., Huang, S., Xu, C.Z., Xu, M.: Boosting active learning via improving test performance. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 8566–8574 (2022)
19. Yi, J.S.K., seok Seo, M., Park, J., geol Choi, D.: Using self-supervised pretext tasks for active learning. In: European Conference on Computer Vision (2022)
20. Zhang, B., Li, L., Yang, S., Wang, S., Zha, Z., Huang, Q.: State-relabeling adversarial active learning. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 8753–8762 (2020)
21. Zhang, C., Tavanapong, W., Kijkul, G., Wong, J.S., de Groen, P.C., Oh, J.H.: Similarity-based active learning for image classification under class imbalance. 2018 IEEE International Conference on Data Mining (ICDM) pp. 1422–1427 (2018)
22. Zhang, J., Katz-Samuels, J., Nowak, R.D.: Galaxy: Graph-based active learning at the extreme. ArXiv **abs/2202.01402** (2022)
23. Zhang, W., Zhu, L., Hallinan, J., Makmur, A., Zhang, S., Cai, Q., Ooi, B.C.: Boostmis: Boosting medical image semi-supervised learning with adaptive pseudo labeling and informative active annotation. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 20634–20644 (2022)