Daniel Kottke Georg Krempl
Andreas Holzinger Barbara Hammer

IAL@ECML PKDD 2022

Workshop on Interactive Adaptive Learning

Proceedings

The European Conference on Machine Learning and
Principles and Practice of Knowledge Discovery in Databases
(ECML PKDD 2022)

Grenoble, France, September 23, 2022

Copyright © 2022 for the individual papers by the papers' authors.

Copyright © 2022 for the volume as a collection by its editors. This volume and its papers are published under the Creative Commons License Attribution 4.0 International (CC BY 4.0).

Preface

Science, technology, and commerce increasingly recognise the importance of machine learning approaches for data-intensive, evidence-based decision making. This is accompanied by increasing numbers of machine learning applications and volumes of data. Nevertheless, the capacities of processing systems or human supervisors or domain experts remain limited in real-world applications. Furthermore, many applications require fast reaction to new situations, which means that first predictive models need to be available even if little data is yet available. Therefore approaches are needed that optimise the whole learning process, including the interaction with human supervisors, processing systems, and data of various kind and at different timings: techniques for estimating the impact of additional resources (e.g. data) on the learning progress; techniques for the active selection of the information processed or queried; techniques for reusing knowledge across time, domains, or tasks, by identifying similarities and adaptation to changes between them; techniques for making use of different types of information, such as labeled or unlabeled data, constraints or domain knowledge. Such techniques are studied for example in the fields of adaptive, active, semi-supervised, and transfer learning. However, this is mostly done in separate lines of research, while combinations thereof in interactive and adaptive machine learning systems that are capable of operating under various constraints, and thereby address the immanent real-world challenges of volume, velocity and variability of data and data mining systems, are rarely reported. Therefore, this workshop aims to bring together researchers and practitioners from these different areas, and to stimulate research in interactive and adaptive machine learning systems as a whole. It continues a successful series of events at ECML PKDD 2017 in Skopje (Workshop and Tutorial), IJCNN 2018 in Rio (Tutorial), ECML PKDD 2018 in Dublin (Workshop), ECML PKDD 2019 in Würzburg (Workshop and Tutorial), virtual ECML PKDD 2020, and 2021 (Workshop).

The workshop aims at discussing techniques and approaches for optimising the whole learning process, including the interaction with human supervisors, processing systems, and includes adaptive, active, semi-supervised, and transfer learning techniques, and combinations thereof in interactive and adaptive machine learning systems. Our objective is to bridge the communities researching and developing these techniques and systems in machine learning and data mining. Therefore, we welcome contributions that present a novel problem setting, propose a novel approach, or report experience with the practical deployment of such a system and raise unsolved questions to the research community.

All in all, we accepted 6 papers (9 papers submitted) to be published in these workshop proceedings. The authors discuss approaches, identify challenges and gaps between active learning research and meaningful applications, as well as define new application-relevant research directions. We thank the authors for their submissions and the program committee for their hard work.

September 2022

Daniel Kottke, Georg Krempl,
Andreas Holzinger, Barbara Hammer

Organization

Organizing Committee

Daniel Kottke	University of Kassel
Georg Krempl	Utrecht University
Andreas Holzinger	University of Natural Resources and Life Sciences, Vienna
Barbara Hammer	Bielefeld University

Program Committee

Mirko Bunse	Dortmund University
Marek Herde	University of Kassel
Martin Holena	Institute of Computer Science
Denis Huseljic	University of Kassel
Dino Ienco	INRAE Montpellier
Jörg Schlötterer	University of Duisburg-Essen
Christin Seifert	University of Duisburg-Essen
Vinicius Souza	Pontifícia Universidade Católica do Paraná
Myra Spiliopoulou	University of Magdeburg

Table of Contents

Research Papers

A Concept for Automated Polarized Web Content Annotation based on Multimodal Active Learning

Marek Herde[1 [0000−0003−4908−122X]], Denis Huseljic[1 [0000−0001−6207−1494]], Jelena Mitrović[2 [0000−0003−3220−8749]], Michael Granitzer[2 [0000−0003−3566−5507]], and Bernhard Sick[1 [0000−0001−9467−656X]]

[1] University of Kassel, Wilhelmshöher Allee 73, 34121 Kassel, Germany
{marek.herde | dhuseljic | bsick}@uni-kassel.de
[2] University of Passau, Innstrasse 43, 94032 Passau, Germany
{michael.granitzer | jelena.mitrovic}@uni-passau.de

Abstract. Active learning (AL) techniques hardly cope with complex annotations tasks, where, for example, annotations might express relationships across data modalities. As a use case, we consider the task of automatically detecting and reporting multimodal, polarized web content (PWC). Samples of this content type emerge dynamically, covering a broad spectrum of topics. Thus, training machine learning systems for detecting PWC is challenging, particularly if it needs to be done with minimum annotation cost. In this article, we propose the concept of multimodal AL for complex annotations in the context of PWC detection and formulate the resulting challenges as questions for future research.

Keywords: Active Learning · Multimodal Data · Semantic Annotation · Polarized Web Content · Hateful Memes.

1 Motivation

Supervised *machine learning* (ML) relies on vast amounts of annotated data often provided by human annotators in a labor-intensive process. *Active learning* (AL) addresses this problem of costly data annotation by intelligently querying annotators [2]. The goal is to maximize an ML system's performance while minimizing the annotation cost. Although AL techniques have shown their benefit for classification and regression tasks [7], they hardly cope with more complex annotation tasks, where annotations might

- express relationships across data modalities (A1),
- describe (semantic) relationships between concepts (A2),
- come along with a high level of error-proneness and potential disagreement among annotators due to an ambiguous context (A3),
- or require modeling background knowledge and sociodemographic factors of annotators to estimate the quality of annotations (A4).

© 2022 for this paper by its authors. Use permitted under CC BY 4.0.

As a use case, we consider the task of automatically detecting and reporting potential multimodal [9], abusive web content in political communication, which is in most cases strongly polarized. We use *polarized web content* (PWC) instead of related expressions such as hateful memes [4,5] to highlight this polarized nature. Generally, PWC comes in many forms, is subjective, depends on the context, and frequently requires background knowledge to be understood [13]. In this article, we refer to PWC as multimodal online content, mainly text and images, that can be found on social media and has, e.g., defamatory or abusive characteristics (at least from the viewpoint of certain groups of persons). The left side of Fig. 1 shows a PWC sample composed of an image of the burning World Trade Center on 09/11 and an image of a Muslim congresswoman, Mrs. Ilhan Abdullahi Omar. These two images are combined with a textual contradiction of "never forget" and "you have forgotten". The polarized context arises from combining images and text (A1), which relates the concepts Twin Towers to Muslims and terrorism (A2). Identifying this polarization requires knowledge about American history and politics (A4) or otherwise may result in erroneous annotations (A3). Such PWC samples emerge dynamically and unforeseeably, covering a broad spectrum of concepts. Thus, training ML systems for detecting PWC is challenging, particularly if it needs to be done annotation cost-efficiently.

Within this article, we view PWC detection as a challenging sample application with real-world impact [11] to initiate research on extending AL systems toward complex annotations of multimodal data. Therefore, we propose our concept of *multimodal active learning for complex annotations* (MALCOM) and formulate the associated challenges as questions for future research.

2 Concept

We envision MALCOM as an extension of traditional AL [2], which assumes a single omniscient annotator providing categorical labels as annotations, toward (1) *semantic annotation graphs* (SAGs) [15] as complex, multimodal annotations and (2) an AL strategy selecting pairs of annotators and queries, e.g., samples. The objective is to semi-automatically build models that can identify PWC and analyze it by annotating a potential PWC sample with an SAG. Such an SAG describes the PWC samples' contents, explains why its contents can be seen as polarized, and reflects the potential uncertainty in that analysis. Fig. 1 shows a PWC sample and its SAG to illustrate this objective. In the following, we outline our two envisioned extensions of AL and PWC detection in more detail.

Extension 1 – Complex, Multimodal Annotations: Existing PWC detection approaches focus on standard supervised learning settings with categorical labels as annotations [1,6,16]. The outputs or embeddings of vision and language models are typically combined as input for a final decision model. Our proposed SAGs represent an alternative combination strategy for the two modalities of images and text. SAGs allow decisions on a higher semantic level, which fosters explainability and decouples objective annotation tasks such as concept analysis of images and texts from more subjective decisions on polarization. We

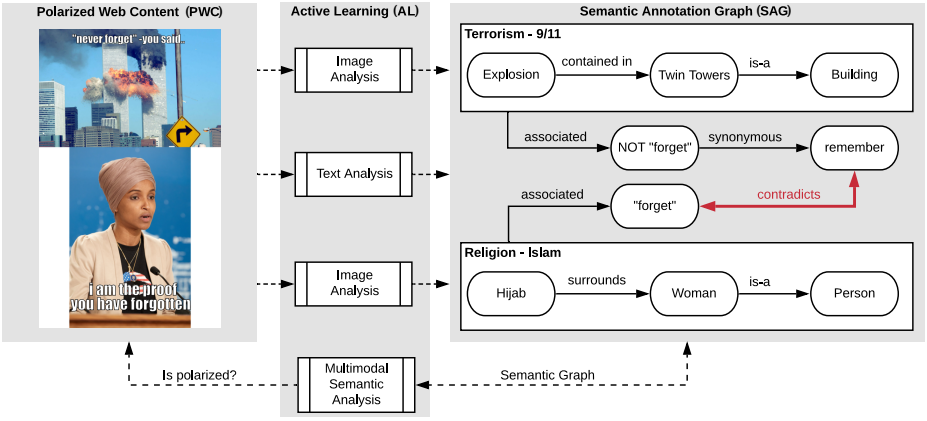


Fig. 1. PWC sample[1] with racist motive (left) and corresponding SAG (right) obtained by combined image and text analysis: Rounded rectangles represent concepts, arrows represent relations, and rectangular boxes represent inferred concepts. As a typical indicator of PWC, a contradicting relation is highlighted in red. AL (center) is applied for (1) unimodal image and text analysis and for (2) inferring whether a sample is polarized from the SAG through multimodal semantic analysis. In this simplified figure, we do not show additional information that is provided with the SAG, e.g., uncertainty regarding object classes or positions in images, relations beyond contradictions, etc.

argue that this is a more efficient way of generating precise automatic classifications of PWC. Methodologically, we have to go far beyond annotating images or text individually but considering their relationships. Annotations may describe positions of objects in images (regions of interest), comparisons of two images or texts, the importance of specific contexts for decisions, a degree of polarization, confidence estimates regarding decisions, etc. We need to develop a proper semantic model, e.g., ontologies [8,12], covering the different modalities and being understandable for annotators. This also includes the ability to express very different PWC concepts over different modalities that go beyond contradictions but include more fuzzy concepts such as antitheses or correlations between concepts.

Extension 2 – Query and Annotator Selection: Identifying PWC requires contextual knowledge of (very recent) events, e.g., pandemics [14]. So instead of building one generic model, we aim at building specialized models for different kinds of PWC, which use pre-trained models (per modality), and fine-tune them in an AL cycle. Extending the AL cycle towards complex annotations of multimodal data, as sketched in Fig. 2, starts with the question of integrating different modalities. First, we consider a pool of annotated unimodal data, i.e., texts and images, which we use to create unimodal models that can annotate

[1] Image above is a compilation of assets, including ©Getty Images/Spencer Platt and ©Getty Images/Adam Bettcher, used under the "Hateful Memes Dataset License Agreement". It is taken from "The Hateful Memes Challenge" [5] for illustrative purposes only and any person depicted in the content is a model.

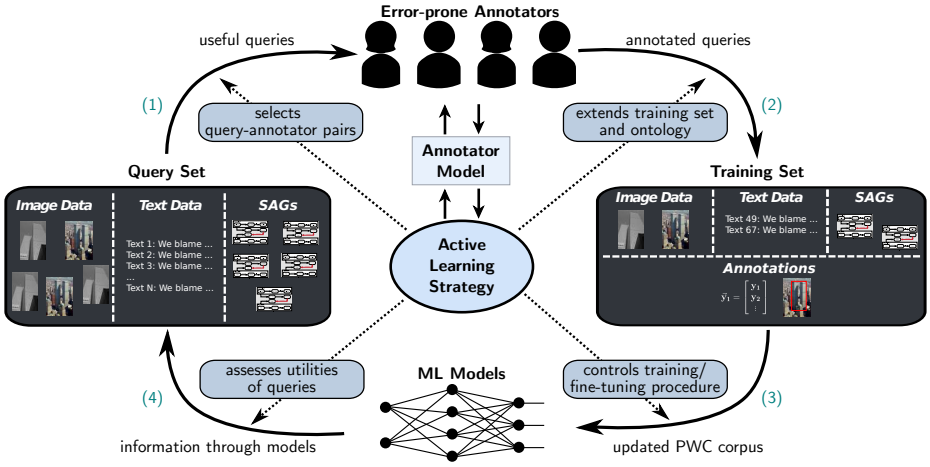


Fig. 2. AL cycle for MALCOM with four main steps: (1) Useful queries are selected from a set of all possible queries regarding potential PWC. For example, we may query annotations for the objects in an image or ask whether an SAG is polarized. (2) Selected queries are presented to a subset of annotators with possibly different (e.g., educational) backgrounds. This subset is determined through an ML-based annotator model estimating the annotators' qualifications. Subsequently, the annotated queries update the training set. (3) The training set representing the current PWC corpus is used to (re-)train several ML models, e.g., an object detection model. (4) The trained models provide information regarding the query set such that the AL cycle starts again using this information for query selection.

the unimodal data semantically. This process in each case results in an SAG, i.e., a typed, attributed graph defined through an ontology-based annotation scheme. Later, the SAGs are merged into a joint, multimodal SAG. Similar to traditional AL strategies, we need to identify promising candidates – initially images and texts, later multimodal SAGs – to be annotated. To consider the problem's multimodal nature, the annotations' semantic properties, and the annotators' diverse backgrounds, we must develop new AL selection strategies that account not only for the respective data sample but also for the different kinds of queries and the qualifications of certain annotators regarding the PWC sample at hand. These qualifications (also referred to as annotator performance [2]) may depend on various aspects such as the respective PWC category (e.g., politics) or educational background (e.g., Master's degree in political sciences). The annotator model predicting such qualifications needs to be sensitive to annotator minorities, e.g., by estimating similarities between annotators. Otherwise, we risk ignoring annotator minorities' opinions regarding PWC. Moreover, we must consider that answers regarding the degree to which content is polarized may be highly subjective, i.e., uncertain from an ML perspective [3]. Establishing an objective definition of PWC, similar to hate speech research [10], is a possible way of reducing the subjectivity of PWC annotation.

3 Research Questions

We conclude this article with the following six research questions derived from the above key research objective and the required extensions.

- How can we define ontology-based annotation schemes to express a human's reasoning over classifying web content as (gradually) polarized or not?
- How can we extract image descriptions (part of the SAG) from potentially polarized images (part of the PWC) considering different uncertainty types?
- How can we extend AL for object detection in potentially polarized images?
- How can we extend AL over text extracted from the images to identify rhetorical figures and automatically analyze textual content to create semantic annotations automatically?
- How can we merge unimodal SAGs and extend AL to train models, e.g., graph convolutional networks [17], assessing PWC via multimodal SAGs?
- How can we evaluate the above techniques and build or extend data corpora [5] for research?

Acknowledgements

SPONSORED BY THE
Federal Ministry of Education and Research

The project on which this article is based was partly funded by the German Federal Ministry of Education and Research (BMBF) under the funding code 01|S20049. The authors are responsible for the content of this publication. Furthermore, the authors thank Chandana Priya Nivarthi, Stephan Vogt, Mohammad Wazed Ali, and the anonymous reviewers for their insightful comments to improve this article.

References

1. Gomez, R., Gibert, J., Gomez, L., Karatzas, D.: Exploring Hate Speech Detection in Multimodal Publications. In: WACV. pp. 1470–1478. Snowmass Village, CO (2020)
2. Herde, M., Huseljic, D., Sick, B., Calma, A.: A Survey on Cost Types, Interaction Schemes, and Annotator Performance Models in Selection Algorithms for Active Learning in Classification. IEEE Access **9**, 166970–166989 (2021)
3. Huseljic, D., Sick, B., Herde, M., Kottke, D.: Separation of Aleatoric and Epistemic Uncertainty in Deterministic Deep Neural Networks. In: ICPR. pp. 9172–9179. Virtual (2021)
4. Kiela, D., Firooz, H., Mohan, A., Goswami, V., Singh, A., Fitzpatrick, C.A., Bull, P., Lipstein, G., Nelli, T., Zhu, R., et al.: The Hateful Memes Challenge: Competition Report. In: NeurIPS 2020 Competition and Demonstration Track. pp. 344–360. Virtual (2021)
5. Kiela, D., Firooz, H., Mohan, A., Goswami, V., Singh, A., Ringshia, P., Testuggine, D.: The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes. In: NeurIPS. pp. 2611–2624. Virtual (2020)

6. Kumar, A., Sachdeva, N.: Multimodal cyberbullying detection using capsule network with dynamic routing and deep convolutional neural network. Multimed. Syst. (2021)
7. Kumar, P., Gupta, A.: Active Learning Query Strategies for Classification, Regression, and Clustering: A Survey. JCST **35**(4), 913–945 (2020)
8. Kühn, R., Mitrović, J., Granitzer, M.: GRhOOT: Ontology of Rhetorical Figures in German. In: LREC. Marseille, France (2022)
9. Lahat, D., Adali, T., Jutten, C.: Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects. Proceedings of the IEEE **103**(9), 1449–1477 (2015)
10. MacAvaney, S., Yao, H.R., Yang, E., Russell, K., Goharian, N., Frieder, O.: Hate speech detection: Challenges and solutions. PLOS ONE **14**(8), 1–16 (2019)
11. Mishra, P., Yannakoudakis, H., Shutova, E.: Tackling Online Abuse: A Survey of Automated Abuse Detection Methods. arXiv:1908.06024 (2019)
12. Mitrović, J., O'Reilly, C., Mladenović, M., Handschuh, S.: Ontological representations of rhetorical figures for argument mining. Argument & Computat. **8**(3), 267–287 (2017)
13. Sood, S.O., Antin, J., Churchill, E.: Using Crowdsourcing to Improve Profanity Detection. In: AAAI Spring Symposium 2012 – Wisdom of the Crowd. pp. 69–74. Palo Alto, CA (2012)
14. Uyheng, J., Carley, K.M.: Bots and online hate during the COVID-19 pandemic: case studies in the United States and the Philippines. JCSS **3**(2), 445–468 (2020)
15. Vidal, J.C., Lama, M., Otero-García, E., Bugarín, A.: Graph-based semantic annotation for enriching educational content with linked data. KBS **55**, 29–42 (2014)
16. Yang, F., Peng, X., Ghosh, G., Shilon, R., Ma, H., Moore, E., Predovic, G.: Exploring Deep Multimodal Fusion of Text and Photo for Hate Speech Classification. In: ALW. pp. 11–18. Florence, Italy (2019)
17. Zhang, S., Tong, H., Xu, J., Maciejewski, R.: Graph convolutional networks: a comprehensive review. Comput. Soc. Netw. **6**(1), 1–23 (2019)

BioSegment: Active Learning segmentation for 3D electron microscopy imaging

Benjamin Rombaut[1,2][0000−0002−4022−715X], Joris Roels[2,3][0000−0002−2058−8134], and Yvan Saeys[1,2][0000−0002−0415−1506]

[1] Department of Applied Mathematics, Computer Science and Statistics, Faculty of Science, Ghent University, Ghent, Belgium
{benjamin.rombaut, yvan.saeys}@ugent.be
[2] Data Mining and Modelling for Biomedicine, VIB-UGent Center for Inflammation Research, Ghent, Belgium
[3] VIB Bioimaging Core, VIB-UGent Center for Inflammation Research, Ghent, Belgium

Abstract. Large 3D electron microscopy images require labor-intensive segmentation for further quantitative analysis. Recent deep learning segmentation methods automate this computer vision task, but require large amounts of labeled training data. We present **BioSegment**, a turnkey platform for experts to automatically process their imaging data and fine-tune segmentation models. It provides a user-friendly annotation experience, integration with familiar microscopy annotation software and a job queue for remote GPU acceleration. Various active learning sampling strategies are incorporated, with maximum entropy selection being the default. For mitochondrial segmentation, these strategies can improve segmentation quality by 10 to 15% in terms of intersection-over-union score compared to random sampling. Additionally, a segmentation of similar quality can be achieved using 25% of the total annotation budget required for random sampling. By comparing the state-of-the-art in human-in-the-loop annotation frameworks, we show that **BioSegment** is currently the only framework capable of employing deep learning and active learning for 3D electron microscopy data.

Keywords: Active learning · Electron microscopy · Computer vision.

1 Introduction

Volume electron microscopy (vEM or 3D EM) describes a set of high-resolution imaging techniques used in biomedical research to reveal the 3D structure of cells, tissues and small model organisms at nanometer resolution. EM techniques have emerged over the past 20 years, largely in response to the demands of the connectomics field in neuroscience, and vEM is expected to be adopted into mainstream biological imaging [23]. Generally, vEM data processing can be divided into four consecutive steps: preprocessing, segmentation, post-processing and downstream analysis.

© 2022 for this paper by its authors. Use permitted under CC BY 4.0.

For the imaging data to be used by deep learning networks, some additional *preprocessing* transformations include normalization and data augmentation. An imaging experiment often includes metadata of the multiple samples, which need to be compared against each other in downstream analysis. This is documented using a folder structure or a data table. Some preprocessing steps to improve imaging data include denoising [29,38], histogram equalization [39] and artifact removal. Usually, the imaging data is downsampled or *binned* in order to reduce data size and to speed up expert and model annotation, while still retaining enough resolution to allow correct segmentation.

Next is *segmentation*, the detection and delineation of structures of interest. Segmentation is required for extraction of quantitative information from rich vEM data sets. Non-discriminant contrast, diversity of appearance of structures and large image volumes turn vEM segmentation into a highly non-trivial problem, where cutting-edge methods relying on state-of-the-art computer vision techniques are still far from reaching human parity in segmentation accuracy [23]. Here, we only consider segmentation of mitochondria, but other cellular components or tissue regions can also be of interest. Pretrained models can be applied to a small sample in order to evaluate segmentation quality. If no model of sufficient quality is available, a new model is created by using some training data annotated by an expert (microscopist or biologist). Machine learning methods can be trained to produce different flavors of segmentation, labelling the pixels either by semantics (for example, label all mitochondria pixels as 1 and the rest as 0) or by the objects they belong to (for example, label all pixels of the first mitochondrion as 1, of the second mitochondrion as 2, of the nth mitochondrion as n, with non-mitochondrion pixels as 0).

There are various *post-processing* steps to transform a semantic segmentation to an object instance segmentation, such as connected components and watershed transform. To further clean up the segmentation, there is usually some filtering based on instance size.

After processing all samples of the experiment, a research question is answered in a *downstream analysis*. Statistics of interest are calculated such as number of mitochondria, mitochondria surface and volume. These statistics are summarized in a data table and combined with the experiment metadata to quantify effects.

Although significant progress has been made in recent years, largely owing to the introduction of deep learning-based methods, there is not yet a single reliable and easy-to-use solution for fully automated segmentation of vEM images. Imaging experts must choose between (or combine) manual, semi-automated and fully automated solutions based on the difficulty of the segmentation problem, the data size and the computational expertise and resources of their team or institution. Furthermore, almost all automated solutions rely on machine learning and may require large amounts of example segmentations to train a model, although in some cases models trained for the same task on similar data sets are available and can be applied directly [23].

Machine learning-based segmentation models can be divided into two categories: feature-based learning and deep learning. Feature-based learning methods use a set of predefined features (usually linear and non-linear image filters) as input to a non-linear classifier such as a support vector machine or a random forest that outputs the (semantic) segmentation. They need few examples and are available via user-friendly tools. Methods using deep learning do not rely on pre-computed features but, instead, learn features and segmentation jointly. They can solve more difficult segmentation problems, but their superior accuracy requires much larger amounts of examples, and the training must be performed on graphics processing units (GPUs). Efficient training and post-processing procedures for deep learning methods in vEM constitute an active area of research [23].

For successful application, the deep learning model needs to be trained on data very similar to the data at hand, but annotated vEM training data is time-consuming to create. Various approaches try to alleviate this problem: increasing annotator efficiency using professional annotation software (*i.e.* MIB or Imaris), sparse labeling [36] or refining model predictions using only points [10]. Additionally, model performance can increase through self-supervised learning on large unlabeled and heterogeneous data sets [14], generalizability-enhancing tricks such as data augmentation or domain adaptation [27]. In any case, additional fine-tuning on some labeled domain-specific data will improve segmentation performance and may be even required [11]. When fine-tuning, model performance can be further increased by choosing the most interesting samples to annotation using active learning [24].

Active learning (AL) is a subdomain of machine learning that aims to minimize label effort without sacrificing model performance. This is achieved by iteratively querying a batch of samples to a label providing oracle, adding them to the train set and retraining the predictor. The challenge is to come up with a smart selection criterion to query samples and maximize the steepness of the training curve [33]. In the setting of vEM segmentation, the oracle is a human imaging expert, such as a microscopist or biologist. This makes our application human or expert-in-the-loop, as the expert will be queried to provide labels through an annotation interface. We consider the total volume of EM data as an offline pool of unlabeled 2D training patches. A general overview of a human-in-the-loop annotation workflow using AL for semantic segmentation is given in Figure 1.

To our knowledge, segmentation of vEM data in an AL setting is not an established practice, *i.e.* the recent Empanada napari plugin [11] for vEM only supports random sampling. In other fields, various tools employ AL to great effect: Label Studio [35] is a flexible data annotation tool that supports semantic segmentation, AL and prediction refinement. MONAI Label [12] is an open source image labeling and learning tool that helps researchers and clinicians to collaborate, create annotated datasets, and build AI models. It features 3D segmentation refinement using 3D Slicer and AL sample selection. Kaibu [21] is a web application for visualizing and annotating multidimensional images, featuring deep learning powered interactive segmentation. Ilastik [7] is an easy-to-use

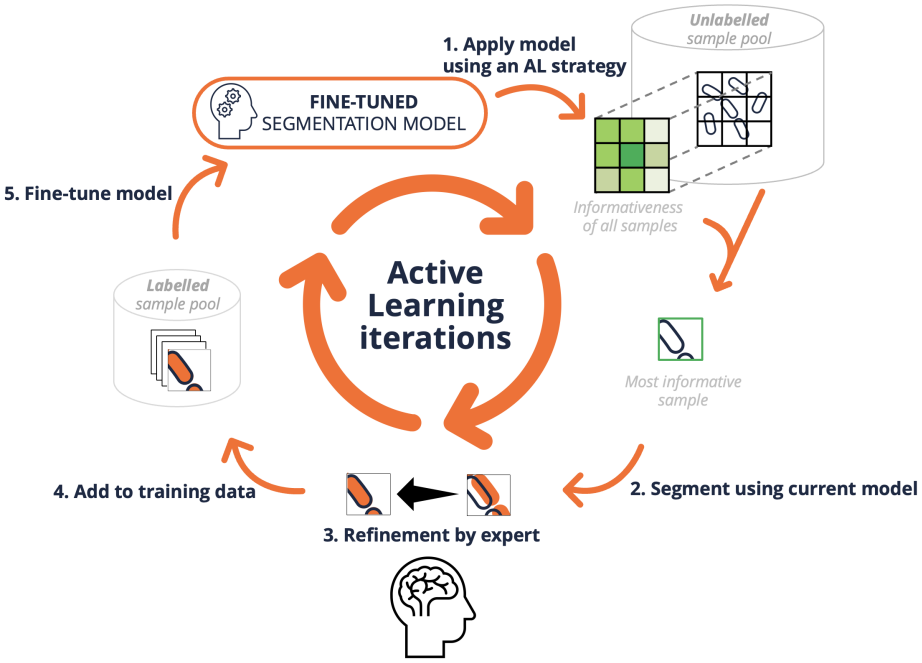


Fig. 1: Overview of active learning for image segmentation. A human imaging expert starts an AL iteration and, using the existing segmentation model and an active learning sampling strategy, ranks the unlabeled samples for labeling. Batches of the most informative samples are annotated by the expert and added to the labeled data pool. After enough new training data is created, the model is fine-tuned on the labeled pool and model performance is expected to improve. The expert can run subsequent AL iterations with the updated model on the remaining unlabeled data, or stop the iterations when model performance is sufficient or the annotation budget is spent.

interactive tool that brings machine-learning-based (bio)image analysis to end users without substantial computational expertise. It contains pre-defined workflows for image segmentation, object classification, counting and tracking.

In this paper, we propose three new contributions:

1. A comparison of five AL strategies for semantic segmentation on three vEM datasets, on which we previously reported in our preprint [28].
2. A feature comparison between current state-of-the-art software frameworks for human-in-the-loop active learning using deep learning segmentation models.
3. **BioSegment**, an integrated platform for imaging experts to process vEM datasets using AL strategies.

First, we describe the software architecture of an AL semantic segmentation framework in Section 2.1, the deep learning models in Section 2.2. We continue with used AL strategies in Section 2.3 and validation datasets in Section 2.4. Our three contributions are presented and discussed in Section 3. Lastly, we envision future work in Section 4 and conclude in Section 5.

2 Methods

2.1 Software Architecture

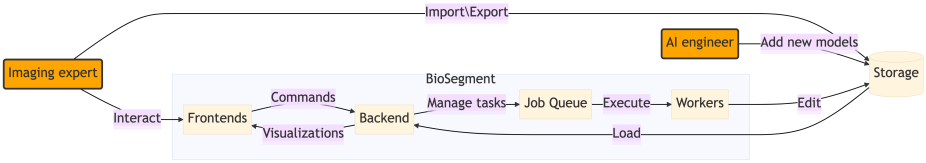


Fig. 2: Flowchart of the **BioSegment** software stack. Users interact with a frontend using their browser. They can visualize a dataset, edit annotations and create segmentations using AI models. The **BioSegment** backend handles the tasks given by the frontend and fetches the datasets from disk storage. For long-running tasks like conversion, active learning, segmentation and fine-tuning, separate workers are used.

We give an overview of the **BioSegment** software architecture in Figure 2. A central database is managed by a backend, implemented using FastAPI. It features a documented REST API, database schemas for all modelled objects and a job queue using Celery and Redis. For long-running tasks like conversion and fine-tuning, separate workers are used, communicating via the messaging bus of the job queue. For data conversion and viewing AICSImageIO [3] and BioFormats [18] are used. The only communication requirement for the workers is access to the Redis server port and the data storage. They can run on a

different machine with GPU acceleration or a network with access to secure and confidential imaging data. Segmentation models and tasks are implemented in PyTorch, and models are serialized to disk. Tensorboard is used to visualize training progression and predicted segmentation performance on selected image samples.

The **BioSegment** software stack is reproducible using *conda* environments and Docker containers. Staging and production deployments are managed using *Docker Swarm*. Restrictive enterprise firewalls can be overcome through the *Traefik* reverse-proxy, which also provides security with automated HTTPS certificate management. Admin interfaces for network, user, database and job queue management are also implemented. Clients can communicate with the backend REST API to add imaging data, manage jobs and visualize results. Using a code generation tool like OpenAPI Generator, the documented REST API from the backend can automatically generate the client code library. This automated step improves maintainability of multiple client interfaces and annotation software plugins. A JavaScript frontend implements most of the backend API and provides management of all data objects like users, datasets, segmentation, annotations and models. A Dash dashboard provides an interface for sparse semantic labelling. Datasets are accessed using file system paths in the backend and workers. These paths resolve to a local mount of the remote disk storage. The mount point is set up using *sshfs*.

2.2 Deep learning methods

We build on the PyTorch Lightning framework, which allows high-level but advanced training loops without the boilerplate code. It supports different accelerator architectures and allows for reproducible and maintainable code. It also features fine-tuning strategies, automated learning rate, batch size finders and support for multiple GPUs and mixed integer training. Various segmentation models are available: our own advanced U-Net implementations in the published *neuralnets* [26] package and *torchvision* [5] which features pretrained model weights.

2.3 Active learning strategies

Five AL strategies were implemented by us and are explained here. We consider the task of semantic segmentation, *i.e.* given an image $\boldsymbol{x} \in X \subset \mathbb{R}^N$ with a total amount of N pixels, we aim to compute a pixel-level labeling $\boldsymbol{y} \in Y$, where $Y = \{0, \ldots, C-1\}^N$ is the label space and C is the number of classes. In particular, we focus on the case of binary segmentation, *i.e.* $C = 2$. Let $\boldsymbol{p}_j(\boldsymbol{x}) = [\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})]_j$ be the probability class distribution of pixel j of a parameterized segmentation algorithm $\boldsymbol{f}_{\boldsymbol{\theta}}$ (*i.e.* an encoder-decoder network, such as U-Net[30]).
Consider a large pool of n i.i.d. sampled data points over the space $Z = X \times Y$ as $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i \in [n]}$, where $[n] = \{1, \ldots, n\}$, and an initial pool of m randomly chosen distinct data points indexed by $S_0 = \{i_j | i_j \in [n]\}_{j \in [m]}$. An active learning

algorithm initially only has access to $\{\boldsymbol{x}_i\}_{i\in[n]}$ and $\{\boldsymbol{y}_i\}_{i\in S_0}$ and iteratively extends the currently labeled pool S_t by querying k samples from the unlabeled set $\{\boldsymbol{x}_i\}_{i\in[n]\setminus S_t}$ to an oracle. After iteration t, the predictor is retrained with the available samples $\{\boldsymbol{x}_i\}_{i\in[n]}$ and labels $\{\boldsymbol{y}_i\}_{i\in S_t}$, thereby improving the segmentation quality. Note that, without loss of generalization, the active learning approaches below are described for $k = 1$. We can also query $k > 1$ samples for k iterations, without retraining, to achieve a batch of samples. The complete active learning workflow is shown in Figure 1.

Maximum entropy sampling [16,17] Maximum entropy is a straightforward selection criterion that aims to select samples for which the predictions are uncertain. Formally speaking, we adjust the selection criterion to a pixel-wise entropy calculation as follows:

$$\boldsymbol{x}^*_{t+1} = \arg\max_{\boldsymbol{x}\in[n]\setminus S_t} -\sum_{j=0}^{N-1}\sum_{c=0}^{C-1} \left[\boldsymbol{p}_j(\boldsymbol{x})\right]_c \log\left[\boldsymbol{p}_j(\boldsymbol{x})\right]_c. \tag{1}$$

In other words, the entropy is calculated for each pixel and summed up. Note that a high entropy will be obtained when $\boldsymbol{p}_j(\boldsymbol{x}) = \frac{1}{C}$, this is exactly when there is no real consensus on the predicted class (*i.e.* high uncertainty).

Least confidence selection [9] Similar to maximum entropy sampling, the least confidence criterion selects samples for which the predictions are uncertain:

$$\boldsymbol{x}^*_{t+1} = \arg\min_{\boldsymbol{x}\in[n]\setminus S_t} \sum_{j=0}^{N-1} \max_{c=0,\ldots,C-1} \left[\boldsymbol{p}_j(\boldsymbol{x})\right]_c. \tag{2}$$

As the name suggests, the least confidence criterion selects the probability that corresponds to the predicted class. Whenever this probability is small, the predictor is not confident about its decision. For image segmentation, we sum up the maximum probabilities in order to select the least confident samples.

Bayesian active learning disagreement [13] The Bayesian active learning disagreement (BALD) approach is specifically designed for convolutional neural networks (CNNs). It makes use of Bayesian CNNs in order to cope with the small amounts of training data that are usually available in active learning workflows. A Bayesian CNN assumes a prior probability distribution placed over the model parameters $\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$. The uncertainty in the weights induces prediction uncertainty by marginalizing over the approximate posterior:

$$\left[\boldsymbol{p}_j(\boldsymbol{x})\right]_c \approx \frac{1}{T}\sum_{t=0}^{T-1} \left[\boldsymbol{p}_j(\boldsymbol{x};\hat{\boldsymbol{\theta}}_t)\right]_c, \tag{3}$$

where $\hat{\boldsymbol{\theta}}_t \sim q(\boldsymbol{\theta})$ is the dropout distribution, which approximates the prior probability distribution p. In other words, a CNN is trained with dropout and inference is obtained by leaving dropout on. This causes uncertainty in

the outcome that can be used in existing criteria such as maximum entropy (Equation (1)).

K-means sampling [8] Uncertainty-based approaches typically sample close to the decision boundary of the classifier. This introduces an implicit bias that does not allow for data exploration. Most explorative approaches that aim to solve this problem transform the input $\boldsymbol{x}$ to a more compact and efficient representation $\boldsymbol{z} = \boldsymbol{g}(\boldsymbol{x})$ (*e.g.* the feature representation before the fully connected stage in a classification CNN). The representation that we used in our segmentation approach was the middle bottleneck representation in the U-Net, which is the learned encoded embedding of the model. The k-means sampling approach in particular then finds k clusters in this embedding using k-means clustering. The selected samples are then the k samples in the different clusters that are closest to the k centroids.

Core set active learning [32] The core set approach is an active learning approach for CNNs that is not based on uncertainty or exploratory sampling. Similar to k-means, samples are selected from an embedding $\boldsymbol{z} = \boldsymbol{g}(\boldsymbol{x})$ in such a way that a model trained on the selection of samples would be competitive for the remaining samples. Similar as before, the representation that we used in our segmentation approach was the bottleneck representation in the U-Net. In order to obtain such competitive samples, this approach aims to minimize the so-called core set loss. This is the difference between the average empirical loss over the set of labeled samples (*i.e.* S_t) and the average empirical loss over the entire dataset that includes the unlabeled points (*i.e.* $[n]$).

2.4 Validation datasets

Three public EM datasets were used to validate our approach:

- The EPFL dataset[4] represents a $5 \times 5 \times 5$ μm^3 section taken from the CA1 hippocampus region of the brain, corresponding to a $2048 \times 1536 \times 1065$ volume. Two $1048 \times 786 \times 165$ subvolumes were manually labelled by experts for mitochondria. The data was acquired by a focused ion-beam scanning EM, and the resolution of each voxel is approximately $5 \times 5 \times 5$ nm^3.
- The VNC dataset[5] represents two $4.7 \times 4.7 \times 1$ μm^3 sections taken from the *Drosophila melanogaster* third instar larva ventral nerve cord, corresponding to a $1024 \times 1024 \times 20$ volume. One stack was manually labelled by experts for mitochondria. The data was acquired by a transmission EM and the resolution of each voxel is approximately $4.6 \times 4.6 \times 45$ nm^3.
- The MiRA dataset[6][37] represents a $17 \times 17 \times 1.6$ μm^3 section taken from the mouse cortex, corresponding to a $8624 \times 8416 \times 31$ volume. The complete volume was manually labelled by experts for mitochondria. The data was

[4] Data available at https://cvlab.epfl.ch/data/data-em/
[5] Data available at https://github.com/unidesigner/groundtruth-drosophila-vnc/
[6] Data available at http://95.163.198.142/MiRA/mitochondria31/

acquired by an automated tape-collecting ultramicrotome scanning EM, and the resolution of each voxel is approximately $2 \times 2 \times 50$ nm^3.

In order to properly validate the discussed approaches, we split the available labeled data in a training and testing set. In the cases of a single labeled volume (VNC and MiRA), we split these datasets halfway along the y axis. A smaller U-Net (with 4 times less feature maps) was initially trained on $m = 20$ randomly selected 128×128 samples in the training volume (learning rate of $1e^{-3}$ for 500 epochs). Next, we consider a pool of $n = 2000$ samples in the training data to be queried. Each iteration, $k = 20$ samples are selected from this pool based on one of the discussed selection criteria, and added to the labeled set S_t, after which the segmentation network is fine-tuned (learning rate of $5e^{-4}$ for 200 epochs). This procedure is repeated for $T = 25$ iterations, leading to a maximum training set size of 500 samples. We validate the segmentation performance using the intersection-over-union (IoU) metric, also known as the Jaccard score:

$$J(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{\sum_i [\boldsymbol{y} \cdot \hat{\boldsymbol{y}}]_i}{\sum_i [\boldsymbol{y}]_i + \sum_i [\hat{\boldsymbol{y}}]_i - \sum_i [\boldsymbol{y} \cdot \hat{\boldsymbol{y}}]_i} \tag{4}$$

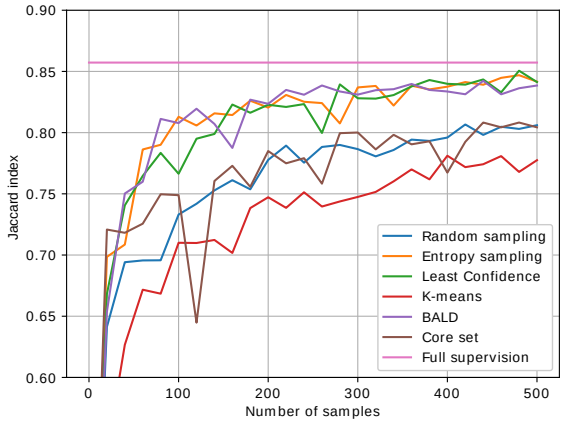
3 Results

3.1 Active Learning validation

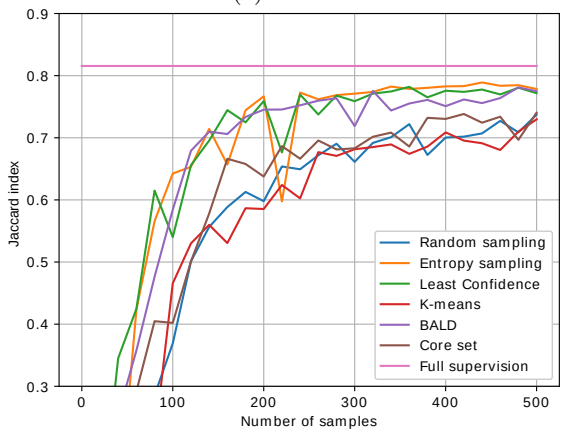
We validated five AL strategies on three public EM datasets. The resulting learning curves of the discussed approaches on the three datasets are shown in Figure 3. We additionally show the performance obtained by full supervision (*i.e.* all labels are available during training), which is the maximum achievable segmentation performance. There is an indication that maximum entropy sampling, least confidence selection and BALD outperform the random sampling baseline. These methods obtain about 10 to 15% performance increase for the same amount of available labels for all datasets. Additionally, a segmentation of similar quality can be achieved using 25% of the total annotation budget required for random sampling. The core set approach performs similar to slightly better than the baseline. We expect that this method can be improved by considering alternative embeddings. Lastly, we see that k-means performs significantly worse than random sampling. Even though this could also be an embedding problem such as with the core set approach, we think that exploratory sampling alone will not allow the predictor to learn from challenging samples, which are usually outliers. We expect that a hybrid approach based on both exploration and uncertainty might lead to better results, and consider this future work.

Figure 4 shows qualitative segmentation results on the EPFL dataset. In particular, we show results of the random, k-means and maximum entropy sampling methods using 120 samples, and compare this to the fully supervised approach. The maximum entropy sampling technique is able to improve the others by a large margin and closes the gap towards fully supervised learning significantly.

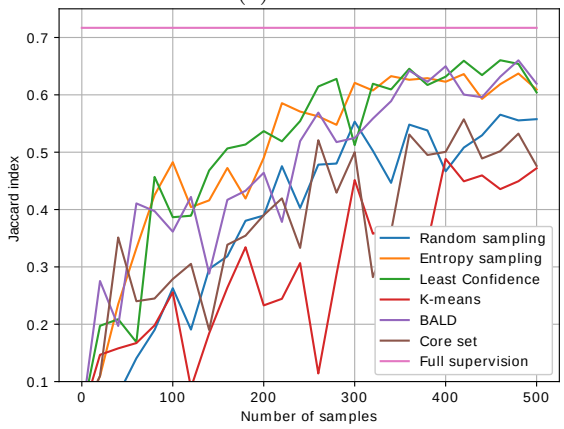
Lastly, we are interested in what type of samples the active learning approaches select for training. Figure 5 shows 4 samples of the VNC dataset that



(a) EPFL

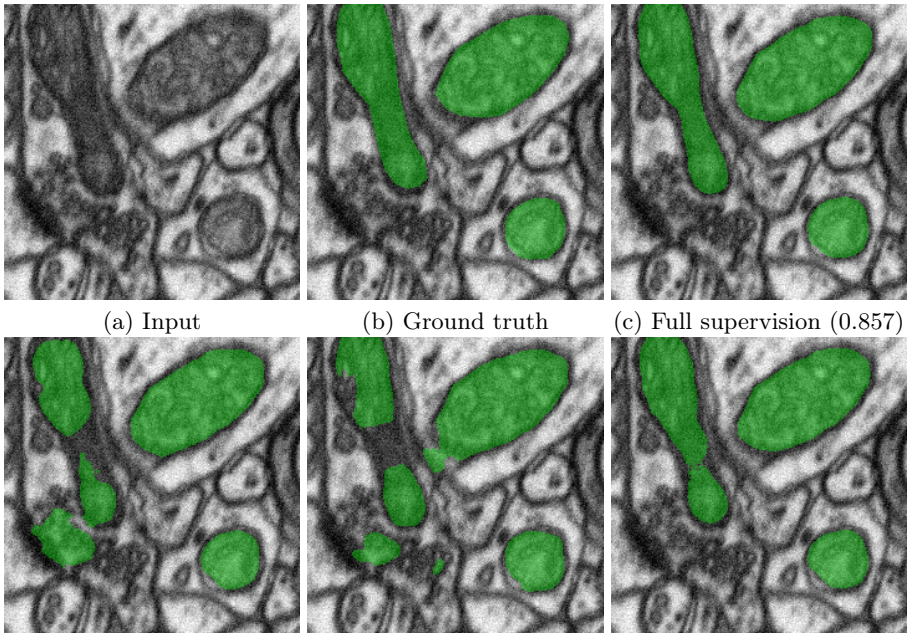


(b) VNC



(c) MiRA

Fig. 3: Learning curves for the five discussed active learning approaches, random sampling and full supervision for the three different datasets. Entropy sampling performs well across the datasets. Note that for entropy and random sampling for EPFL, the difference in model performance for the same number of samples (difference in y-axis) is 15% and difference in number of samples needed for the same model performance (difference in x-axis) is 25%.



(a) Input (b) Ground truth (c) Full supervision (0.857)

(d) Random (0.733) (e) k-means (0.710) (f) Maximum entropy (0.813)

Fig. 4: Segmentation results obtained from an actively learned U-Net with 120 samples of the EPFL dataset based on random, k-means and maximum entropy sampling, and a comparison to the fully supervised approach. Jaccard scores are indicated between brackets.

correspond to the highest prioritized samples, according to the least confidence criterion, that were selected in the first 4 iterations. The top row illustrates the probability predictions of the network at that point in time, whereas the bottom row shows the pixel-wise uncertainty of the sample (*i.e.* the maximum in Equation (2)). Note that the initial predictions at $t = 1$ are of poor quality, as the network was only trained on 20 samples. Moreover, the uncertainty is high in regions where the network is uncertain, but it is low in regions where the network is wrong. The latter is a common issue in active learning and related to the exploration vs. uncertainty trade-off. However, over time, we see that the network performance improves, and more challenging samples are being queried to the oracle.

3.2 Feature comparison

We define five software features of interest for an AL software framework for vEM data:

Interactive fine-tuning The expert should be able to fine-tune a segmentation model with their own newly annotated data. For deep learning models, this

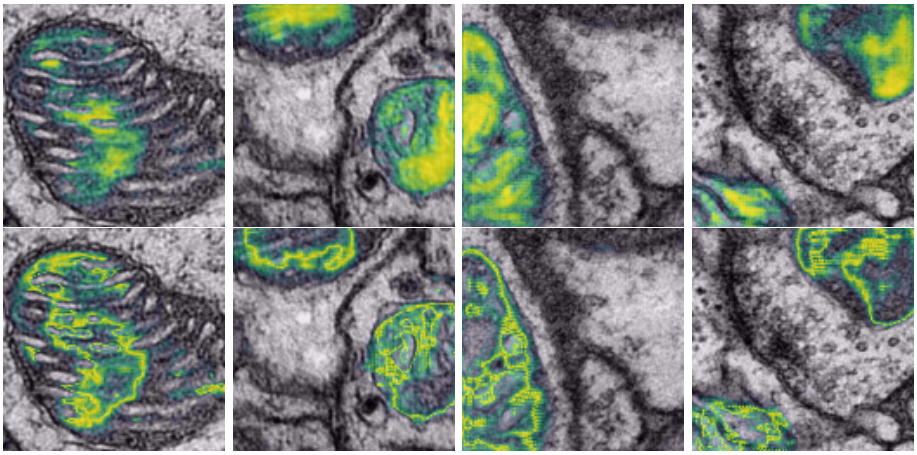


Fig. 5: Illustration of the selected samples in the VNC dataset over time in the active learning process. The top row shows the pixel-wise prediction of the selected samples at iterations 1 through 4. The bottom row show the pixel-wise least confidence score on the corresponding images.

	Software features				
Software frameworks	interactive fine-tuning	active learning	large datasets	3D support	remote resources
Label Studio [35]	x	x			x
Kaibu [21]	x				x
napari-empanada [11]	x		x	x	
ilastik [7]	x	x	x	x	x
MONAI Label [12]	x	x	x	x	x
BioSegment	x	x	x	x	x

Table 1: Comparison of open-source software frameworks for human-in-the-loop active learning using segmentation models.

involves optional GPU acceleration and reporting on training status and accuracy. All considered frameworks have this feature.

Active learning The framework should support sampling the unlabeled data using an AL strategy. Some frameworks have only proposed this feature for future work and only implemented a random sampling strategy.

Large datasets The expert should be able to apply existing and newly trained models on their whole dataset, no matter the size. This feature is the most lacking, as it requires support for tiled inference and long-running jobs.

3D support The supported annotation interfaces of the framework should allow the expert to freely browse consecutive slices or volumes in 3D.

Remote resources In order to process large datasets, large storage and computational resources such as workstations and GPU's are needed. This usually

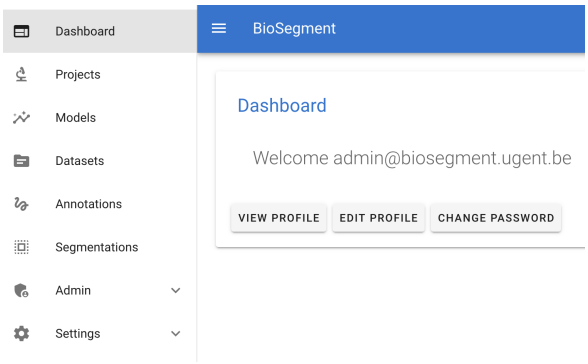
requires a flexible software architecture and communication over a network interface or software worker queue.

BioSegment combines the desirable software features needed for analyzing vEM data in one framework and is the only AL framework currently used as such. Ilastik is an established interactive annotation tool with support for standard ML segmentation. Recently, it has added beta support for a remote GPU task server (*tiktorch* [2]) and an active learning ML segmentation workflow [1] using SLIC features and supervoxels for vEM. All this functionality is however still in beta, sparsely documented and not yet applied for deep learning models or mitochondria segmentation. napari-empanada is the most recent development in vEM segmentation, but has no support for AL. The lack of support for remote resources could however be solved by running napari remotely using VirtualGL [22] or using a remote Dask cluster or data store [25]. Lastly, the recently developed feature set of MONAI Label is exciting. However, it is little over a year old, has no reported usage by the EM community, and mostly targets radiology and pathology use cases. Nevertheless, it can be adapted for EM and integrated in our **BioSegment** workflow, as shown in 6c. We note that a remote GPU-accelerated model execution is a hallmark of most frameworks: the worker queue in **BioSegment** and MONAI Label, the Label Studio ML backend and the ilastik tiktorch server.

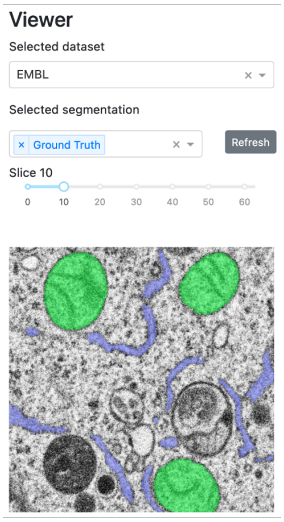
3.3 BioSegment workflow

After image capturing and storing the raw microscopy data on disk, experts start the **BioSegment** workflow. Through a dedicated dashboard (Figure 6a), the expert can create a new dataset holder and import the imaging data directly by providing the folder path. This starts a new annotation workflow. The expert can start preprocessing and segmentation jobs for the whole dataset and visualize the result (Figure 6b).
If no existing model has the desired quality, experts can choose a model to fine-tune. A batch of sampled images from the unlabeled dataset is chosen for annotation. An interface for sparse semantic labelling is provided, and the subset can be exported to different bioimaging annotation software like 3D Slicer (Figure 6c), Amira (ThermoFisher Scientific), Imaris (Oxford Instruments), Fiji [31] or napari [34]. The chosen model can be fine-tuned on the created training data and model performance can again be evaluated.

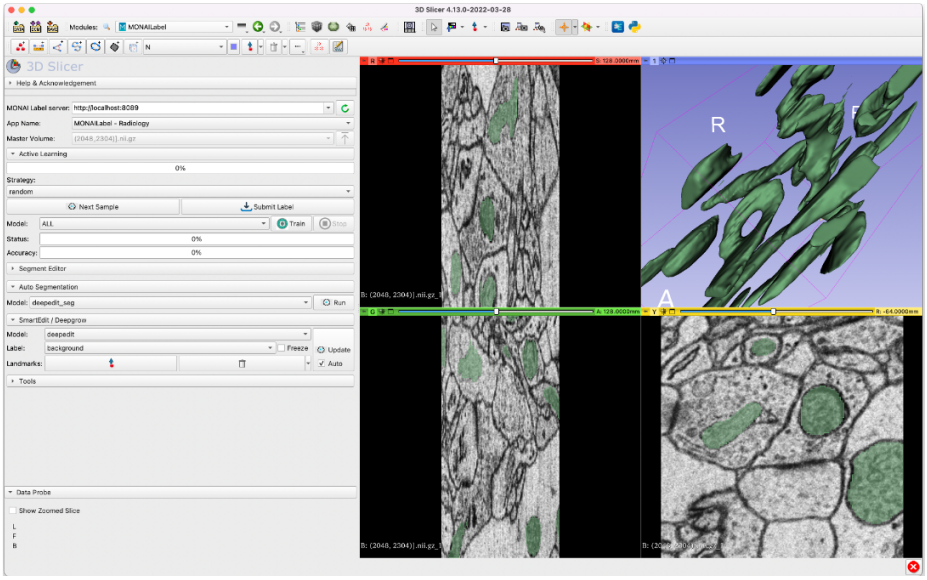
The annotation workflow can be augmented using active learning loops: the subset of images to be sampled can be selected by one of the five implemented active learning strategies, informed by the chosen model. After annotation by the expert, this model will be fine-tuned and again be used for selecting the following batch of images, creating an active learning loop and immediately incorporating the expert feedback in the sampling process. By empowering imaging experts with a dashboard to run by themselves multiple active learning iterations and segmentation jobs on their datasets, active learning can be incorporated into their normal annotation workflow. The expert can stop the iterations when they



(a) **BioSegment** Dashboard



(b) **BioSegment** model viewer



(c) External viewer (3D Slicer)

Fig. 6: Three example **BioSegment** interfaces. 6a: The dashboard where users can manage all settings. 6b: Models can be viewed and fine-tuned with training data using the viewer interface data. 6c: Results can be exported and used in external programs such as 3D Slicer and MONAI Label.

are satisfied with the segmentation quality in the preview or their annotation budget is depleted. The number of iterations is usually three or higher, but this highly depends on the dataset and on the computer vision task.
When a segmentation model of high enough quality is achieved, it can be applied to the whole dataset like the other pre-existing models. The labelled data can be added to a pool of general training data in order to train better performing models for future fine-tuning tasks. Experts can download the segmented dataset for further downstream analysis.
The **BioSegment** software stack is deployed at biosegment.ugent.be and used internally at the Flemish Institute for Biotechnology (VIB) for annotating new vEM datasets. It automates the previous manual active learning loops between imaging experts at a partnering imaging facility and deep learning scientists in our computational lab. The code is available at GitHub and features a documentation site.

4 Future work

Computer vision is not limited to single class semantic segmentation problems. Mitochondria form 3D shapes and networks, requiring 3D post-processing to achieve accurate instance segmentation. Other cell organelles are of equal interest, and large amounts of existing data are now available through the OpenOrganelle data portal [15]. Multi-class semantic segmentation is currently implemented, but the label map is not standardized. Interfacing with the BioImage Model Zoo [20] would help in this regard. We also plan to further integrate preprocessing steps like denoising, as these are still done with a separate script. Beside image enhancement, volume reconstruction and multimodal registration are two different data processing workflows in EM that would be beneficial to implement.
Recent advances in tooling include napari, an interactive, multidimensional image viewer for Python and the Java-based Paintera [4] for dense labeling of large 3D datasets. Together with cloud-based file formats like NGFF [19] these would facilitate annotating and processing large imaging experiments. Integration with Dask [25], a flexible open-source Python library for parallel computing, would allow immediate preview of complex workflows and scaling for the whole dataset using long-running jobs. These advances allow for new annotation experiences. For example, a region-of-interest free approach where the annotator freely browses the whole dataset and the current model prediction and uncertainty is lazily updated depending on the view-port. By creating multi-resolution maps of the model uncertainty, the expert is informed on the model performance over the whole dataset and is free to choose which regions to annotate.
Complexity of the software stack can be out-sourced to existing free software libraries. Lightning AI further removes boilerplate code in deep learning models by providing App and Flow interfaces. Data management and worker communication in **BioSegment** can be handled by *Girder*, which also utilizes the Celery job queue. By creating or integrating with plugins for already established an-

notation tools, adoption of the **BioSegment** workflow can be improved. Active development in the 3D Slicer and napari communities for chunked and multidimensional file formats, instance segmentation and collaborative annotation proofreading tools will also improve the future **BioSegment** feature set. For AL research, it would be valuable to add instrumentation to these annotation tools in order to better capture the burden of the annotation work by the expert. Currently, number of samples and total annotated pixels can be measured, but actual time and number of clicks would be more accurate metrics. **BioSegment** can be adapted to capture these interesting metrics. Greater model performance can be achieved by including automated hyperparameter optimization such as Optuna [6]. This and other AutoML strategies would further automate model training.

5 Conclusions

We present **BioSegment**, a turnkey solution for Active Learning segmentation of vEM imaging. It provides a user-friendly annotation experience, integration with familiar microscopy annotation software and a job queue for remote GPU acceleration. Expert annotation is augmented using active learning strategies. For mitochondrial segmentation, these strategies can improve segmentation quality by 10 to 15% in terms of intersection-over-union score compared to random sampling. Additionally, a segmentation of similar quality can be achieved using 25% of the total annotation budget required for random sampling. The software stack is maintainable through various automated tests, and the code base is published under an open-source license. By comparing the state-of-the-art in human-in-the-loop annotation frameworks, we show that **BioSegment** is currently the only framework capable of employing deep learning and active learning for 3D electron microscopy data.

Acknowledgements The computational resources and services used in this work were provided by NVIDIA, VIB IRC IT and the VSC (Flemish Supercomputer Center), funded by the Research Foundation – Flanders (FWO) and the Flemish Government. Imaging data and feedback was provided by the VIB BioImaging Core. Funding was provided by the Flanders AI Research Program.

References

1. ilastik - Voxel Segmentation Workflow (beta), `https://www.ilastik.org/documentation/voxelsegmentation/voxelsegmentation`
2. tiktorch (Dec 2021), `https://github.com/ilastik/tiktorch`, original-date: 2017-07-18T10:25:47Z
3. AICSImageIO (Jun 2022), `https://github.com/AllenCellModeling/aicsimageio`, original-date: 2019-06-27T16:43:22Z
4. Paintera (Jun 2022), `https://github.com/saalfeldlab/paintera`, original-date: 2018-04-26T21:55:50Z

5. pytorch/vision (Jun 2022), https://github.com/pytorch/vision, original-date: 2016-11-09T23:11:43Z
6. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A Next-generation Hyperparameter Optimization Framework (Jul 2019). https://doi.org/10.48550/arXiv.1907.10902, http://arxiv.org/abs/1907.10902, number: arXiv:1907.10902 arXiv:1907.10902 [cs, stat]
7. Berg, S., Kutra, D., Kroeger, T., Straehle, C.N., Kausler, B.X., Haubold, C., Schiegg, M., Ales, J., Beier, T., Rudy, M., Eren, K., Cervantes, J.I., Xu, B., Beuttenmueller, F., Wolny, A., Zhang, C., Koethe, U., Hamprecht, F.A., Kreshuk, A.: ilastik: interactive machine learning for (bio)image analysis. Nature Methods **16**(12), 1226–1232 (Dec 2019). https://doi.org/10.1038/s41592-019-0582-9, https://www.nature.com/articles/s41592-019-0582-9, number: 12 Publisher: Nature Publishing Group
8. Bodó, Z., Minier, Z., Csató, L.: Active Learning with Clustering. In: Active Learning and Experimental Design workshop In conjunction with AISTATS 2010. pp. 127–139. JMLR Workshop and Conference Proceedings (Apr 2011), https://proceedings.mlr.press/v16/bodo11a.html, iSSN: 1938-7228
9. Brinker, K.: Incorporating diversity in active learning with support vector machines. In: Proceedings of the Twentieth International Conference on International Conference on Machine Learning. pp. 59–66. ICML'03, AAAI Press, Washington, DC, USA (Aug 2003)
10. Chen, X., Zhao, Z., Zhang, Y., Duan, M., Qi, D., Zhao, H.: FocalClick: Towards Practical Interactive Image Segmentation. arXiv:2204.02574 [cs] (Apr 2022), http://arxiv.org/abs/2204.02574, arXiv: 2204.02574 version: 1
11. Conrad, R.W., Narayan, K.: Instance segmentation of mitochondria in electron microscopy images with a generalist deep learning model. Tech. rep., bioRxiv (May 2022). https://doi.org/10.1101/2022.03.17.484806, https://www.biorxiv.org/content/10.1101/2022.03.17.484806v2, section: New Results Type: article
12. Diaz-Pinto, A., Alle, S., Ihsani, A., Asad, M., Nath, V., Pérez-García, F., Mehta, P., Li, W., Roth, H.R., Vercauteren, T., Xu, D., Dogra, P., Ourselin, S., Feng, A., Cardoso, M.J.: MONAI Label: A framework for AI-assisted Interactive Labeling of 3D Medical Images. arXiv:2203.12362 [cs, eess] (Mar 2022), arXiv: 2203.12362
13. Gal, Y., Islam, R., Ghahramani, Z.: Deep Bayesian Active Learning with Image Data. Tech. Rep. arXiv:1703.02910, arXiv (Mar 2017). https://doi.org/10.48550/arXiv.1703.02910, http://arxiv.org/abs/1703.02910, arXiv:1703.02910 [cs, stat] type: article
14. Han, H., Dmitrieva, M., Sauer, A., Tam, K.H., Rittscher, J.: Self-Supervised Voxel-Level Representation Rediscovers Subcellular Structures in Volume Electron Microscopy. pp. 1874–1883 (2022), https://openaccess.thecvf.com/content/CVPR2022W/CVMI/html/Han_Self-Supervised_Voxel-Level_Representation_Rediscovers_Subcellular_Structures_in_Volume_Electron_Microscopy_CVPRW_2022_paper.html
15. Heinrich, L., Bennett, D., Ackerman, D., Park, W., Bogovic, J., Eckstein, N., Petruncio, A., Clements, J., Xu, C.S., Funke, J., Korff, W., Hess, H.F., Lippincott-Schwartz, J., Saalfeld, S., Weigel, A.V., Team, C.P.: Automatic whole cell organelle segmentation in volumetric electron microscopy (Nov 2020). https://doi.org/10.1101/2020.11.14.382143, https://www.biorxiv.org/content/10.1101/2020.11.14.382143v1, pages: 2020.11.14.382143 Section: New Results

16. Joshi, A.J., Porikli, F., Papanikolopoulos, N.: Multi-class active learning for image classification. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2372–2379 (Jun 2009). https://doi.org/10.1109/CVPR.2009.5206627, iSSN: 1063-6919
17. Li, X., Guo, Y.: Adaptive Active Learning for Image Classification. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition. pp. 859–866 (Jun 2013). https://doi.org/10.1109/CVPR.2013.116, iSSN: 1063-6919
18. Linkert, M., Rueden, C.T., Allan, C., Burel, J.M., Moore, W., Patterson, A., Loranger, B., Moore, J., Neves, C., MacDonald, D., Tarkowska, A., Sticco, C., Hill, E., Rossner, M., Eliceiri, K.W., Swedlow, J.R.: Metadata matters: access to image data in the real world. Journal of Cell Biology **189**(5), 777–782 (May 2010). https://doi.org/10.1083/jcb.201004104, https://doi.org/10.1083/jcb.201004104
19. Moore, J., Allan, C., Besson, S., Burel, J.M., Diel, E., Gault, D., Kozlowski, K., Lindner, D., Linkert, M., Manz, T., Moore, W., Pape, C., Tischer, C., Swedlow, J.R.: OME-NGFF: a next-generation file format for expanding bioimaging data-access strategies. Nature Methods pp. 1–3 (Nov 2021). https://doi.org/10.1038/s41592-021-01326-w, https://www.nature.com/articles/s41592-021-01326-w, bandiera_abtest: a Cc_license_type: cc_by Cg_type: Nature Research Journals Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Computational platforms and environments;Data publication and archiving Subject_term_id: computational-platforms-and-environments;data-publication-and-archiving
20. Ouyang, W., Beuttenmueller, F., Gómez-de Mariscal, E., Pape, C., Burke, T., Garcia-López-de Haro, C., Russell, C., Moya-Sans, L., de-la Torre-Gutiérrez, C., Schmidt, D., Kutra, D., Novikov, M., Weigert, M., Schmidt, U., Bankhead, P., Jacquemet, G., Sage, D., Henriques, R., Muñoz-Barrutia, A., Lundberg, E., Jug, F., Kreshuk, A.: BioImage Model Zoo: A Community-Driven Resource for Accessible Deep Learning in BioImage Analysis (Jun 2022). https://doi.org/10.1101/2022.06.07.495102, https://www.biorxiv.org/content/10.1101/2022.06.07.495102v1, pages: 2022.06.07.495102 Section: New Results
21. Ouyang, W., Le, T., Xu, H., Lundberg, E.: Interactive biomedical segmentation tool powered by deep learning and ImJoy. Tech. Rep. 10:142, F1000Research (Feb 2021). https://doi.org/10.12688/f1000research.50798.1, https://f1000research.com/articles/10-142, type: article
22. Paradis, D.J., Segee, B.: Remote Rendering and Rendering in Virtual Machines. In: 2016 International Conference on Computational Science and Computational Intelligence (CSCI). pp. 218–221 (Dec 2016). https://doi.org/10.1109/CSCI.2016.0048
23. Peddie, C.J., Genoud, C., Kreshuk, A., Meechan, K., Micheva, K.D., Narayan, K., Pape, C., Parton, R.G., Schieber, N.L., Schwab, Y., Titze, B., Verkade, P., Weigel, A., Collinson, L.M.: Volume electron microscopy. Nature Reviews Methods Primers **2**(1), 1–23 (Jul 2022). https://doi.org/10.1038/s43586-022-00131-9, https://www.nature.com/articles/s43586-022-00131-9, number: 1 Publisher: Nature Publishing Group
24. Ren, P., Xiao, Y., Chang, X., Huang, P.Y., Li, Z., Gupta, B.B., Chen, X., Wang, X.: A Survey of Deep Active Learning. ACM Computing Surveys **54**(9), 180:1–180:40 (Oct 2021). https://doi.org/10.1145/3472291, https://doi.org/10.1145/3472291
25. Rocklin, M.: Dask: Parallel Computation with Blocked algorithms and Task Scheduling. Proceedings of the 14th Python in Science Conference pp.

126–132 (2015). https://doi.org/10.25080/Majora-7b98e3ed-013, https://conference.scipy.org/proceedings/scipy2015/matthew_rocklin.html, conference Name: Proceedings of the 14th Python in Science Conference
26. Roels, J.: NeuralNets (May 2022), https://github.com/JorisRoels/neuralnets, original-date: 2019-11-29T09:59:01Z
27. Roels, J., Hennies, J., Saeys, Y., Philips, W., Kreshuk, A.: Domain Adaptive Segmentation In Volume Electron Microscopy Imaging. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). pp. 1519–1522. IEEE, Venice, Italy (Apr 2019). https://doi.org/10.1109/ISBI.2019.8759383, https://ieeexplore.ieee.org/document/8759383/
28. Roels, J., Saeys, Y.: Cost-efficient segmentation of electron microscopy images using active learning. arXiv:1911.05548 [cs] (Nov 2019), http://arxiv.org/abs/1911.05548, arXiv: 1911.05548
29. Roels, J., Vernaillen, F., Kremer, A., Gonçalves, A., Aelterman, J., Luong, H.Q., Goossens, B., Philips, W., Lippens, S., Saeys, Y.: An interactive ImageJ plugin for semi-automated image denoising in electron microscopy. Nature Communications **11**(1), 771 (Feb 2020). https://doi.org/10.1038/s41467-020-14529-0, https://www.nature.com/articles/s41467-020-14529-0, number: 1 Publisher: Nature Publishing Group
30. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation (May 2015). https://doi.org/10.48550/arXiv.1505.04597, http://arxiv.org/abs/1505.04597, number: arXiv:1505.04597 arXiv:1505.04597 [cs]
31. Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J.Y., White, D.J., Hartenstein, V., Eliceiri, K., Tomancak, P., Cardona, A.: Fiji: an open-source platform for biological-image analysis. Nature Methods **9**(7), 676–682 (Jul 2012). https://doi.org/10.1038/nmeth.2019, https://www.nature.com/articles/nmeth.2019, number: 7 Publisher: Nature Publishing Group
32. Sener, O., Savarese, S.: Active Learning for Convolutional Neural Networks: A Core-Set Approach. arXiv:1708.00489 [cs, stat] (Jun 2018), http://arxiv.org/abs/1708.00489, arXiv: 1708.00489
33. Settles, B.: Active Learning Literature Survey. Technical Report, University of Wisconsin-Madison Department of Computer Sciences (2009), https://minds.wisconsin.edu/handle/1793/60660, accepted: 2012-03-15T17:23:56Z
34. Sofroniew, N., Lambert, T., Evans, K., Nunez-Iglesias, J., Bokota, G., Winston, P., Peña-Castellanos, G., Yamauchi, K., Bussonnier, M., Doncila Pop, D., Can Solak, A., Liu, Z., Wadhwa, P., Burt, A., Buckley, G., Sweet, A., Migas, L., Hilsenstein, V., Gaifas, L., Bragantini, J., Rodríguez-Guerra, J., Muñoz, H., Freeman, J., Boone, P., Lowe, A., Gohlke, C., Royer, L., PIERRÉ, A., Har-Gil, H., McGovern, A.: napari: a multi-dimensional image viewer for Python (May 2022). https://doi.org/10.5281/zenodo.6598542, https://zenodo.org/record/6598542
35. Tkachenko, M., Malyuk, M., Holmanyuk, A., Liubimov, N.: Label Studio: Daata labeling software (2020), https://github.com/heartexlabs/label-studio, original-date: 2019-06-19T02:00:44Z
36. Wolny, A., Yu, Q., Pape, C., Kreshuk, A.: Sparse Object-level Supervision for Instance Segmentation with Pixel Embeddings (Apr 2022). https://doi.org/10.48550/arXiv.2103.14572, http://arxiv.org/abs/2103.14572, number: arXiv:2103.14572 arXiv:2103.14572 [cs]

37. Xiao, C., Chen, X., Li, W., Li, L., Wang, L., Xie, Q., Han, H.: Automatic Mitochondria Segmentation for EM Data Using a 3D Supervised Convolutional Network. Frontiers in Neuroanatomy **12** (2018). https://doi.org/10.3389/fnana.2018.00092, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6224513/, publisher: Frontiers Media SA
38. Zhang, K., Li, Y., Zuo, W., Zhang, L., Van Gool, L., Timofte, R.: Plug-and-Play Image Restoration with Deep Denoiser Prior (Jul 2021). https://doi.org/10.48550/arXiv.2008.13751, http://arxiv.org/abs/2008.13751, number: arXiv:2008.13751 arXiv:2008.13751 [cs, eess]
39. Zuiderveld, K.: VIII.5. - Contrast Limited Adaptive Histogram Equalization. In: Heckbert, P.S. (ed.) Graphics Gems, pp. 474–485. Academic Press (Jan 1994). https://doi.org/10.1016/B978-0-12-336156-1.50061-6, https://www.sciencedirect.com/science/article/pii/B9780123361561500616

Enhancing Active Learning with Weak Supervision and Transfer Learning by Leveraging Information and Knowledge Sources

Lukas Rauch, Denis Huseljic, and Bernhard Sick

University of Kassel, Wilhelmshöher Allee 73, 34121 Kassel, Germany
{lukas.rauch, dhuseljic, bsick}@uni-kassel.de

Abstract. One of the major limitations of deploying a machine learning model is the availability of labeled training data and the resulting expensive annotation process. Although active learning (AL) methods may reduce the annotation cost by actively selecting the most-useful instances, a costly human annotator usually provides the labels. Therefore, even with AL, we still consider the annotation process to be time-consuming and expensive. Besides human annotators, though, companies often have a vast amount of information and knowledge sources available that can generate low-cost labels (e.g., a black-box model) or improve the learning process (e.g., a pre-trained model). We present a novel approach that enhances AL with weak supervision (WS) and transfer learning (TL) to reduce the annotation cost by leveraging these sources. Specifically, we consider a black-box model like a rule-based system as an error-prone and weakly-supervised annotator that inexpensively provides labels. We estimate its performance with an annotator model to decide whether a human annotation is required. Additionally, we utilize unlabeled internal and external data by transferring knowledge from a pre-trained model to the AL cycle. We sequentially investigate the impact of WS and TL on annotation cost and model performance in an AL cycle through a use case. Our evaluation shows that our approach can reduce annotation cost by 51% while achieving nearly identical model performance compared to a traditional AL approach.

Keywords: Active Learning · Weak Supervision · Transfer Learning · Information and Knowledge Sources.

1 Introduction

In recent years, there has been an increasing interest in machine learning applications across all industries [25]. In particular, (deep) neural networks (NNs) have proven beneficial for unstructured data types such as image or text data. However, one of the major real-world bottlenecks in deploying a NN is the need for large labeled training data sets to reach peak performance [25,30]. To reduce annotation cost for the training process, active learning (AL) [4,31] is a part of human-in-the-loop learning [13] where we actively select the most-useful instances. The goal is to reduce annotation cost while maximizing the performance

© 2022 for this paper by its authors. Use permitted under CC BY 4.0.

of a model trained on an actively selected subset from an unlabeled data pool [32,12]. However, since a human annotator (HA) usually provides the labels, the annotation process may still be time-consuming and expensive [11]. Besides HAs, companies usually have a wide range of information and knowledge sources [10] available such as an established black-box model (BBM) like a rule-based system [23] or external data and a pre-trained model from the Internet. These sources can provide labels (information source) or contain beneficial knowledge for training NNs (knowledge source). Nevertheless, they are often ignored or not fully utilized in practice. This raises the question of *how to efficiently leverage and extract* information and knowledge from available sources to further reduce the annotation cost in AL.

To address this question, research fields such as weak supervision (WS) [1,5] and transfer learning (TL)[26] provide suitable methods. Specifically, WS methods generate noisy labels at low cost, e.g., with expert-defined rules or labeling heuristics [2,30] and are typically applied after obtaining a high-quality labeled data set. In TL [26], acquired knowledge of a pre-trained model is transferred to a different but related downstream task. Combining AL-WS and AL-TL has already shown promising results to further reduce the annotation cost in AL [7,33]. However, to the best of our knowledge, there has not yet been a combination of all three fields in which multiple available information and knowledge sources are exploited. Therefore, we investigate the following research questions in this work:

Question 1. How can we enhance AL with WS so that we can leverage an available BBM as an information source to reduce the annotation cost with a competitive model performance compared to a traditional AL approach?

Question 2. How far can the inclusion of TL to leverage unlabeled internal and external data as knowledge sources empower the combination of AL-WS and, thus, further reduce the annotation cost and improve the model performance?

To answer those research questions, we conduct experiments in a real-world use case where we thematically classify banking transactions based on text data. We extend an AL cycle with WS, training a classification and annotator model simultaneously. Specifically, we consider an available BBM (a rule-based system in our use case) as an error-prone and weakly-supervised annotator (WSA). The annotator model allows us to decide whether annotations can be performed at low cost by the WSA without a costly HA (a domain expert in our use case). In addition, we further enhance the AL-WS cycle with TL. We fine-tune a pre-trained model (a language model in our use case) from an external source on unlabeled internal data for the downstream task with unsupervised learning. This allows us to use labeled and unlabeled data to train our models in the AL cycle. By doing so, we are the first to provide an approach to combine AL with WS and TL by leveraging multiple available information and knowledge sources. Based on the evaluation of our experiments, we summarize our contributions as follows:

1. Enhancing AL with WS by leveraging a rule-based system as an information source through an annotator model leads to a reduction of the annotation cost by 43% with a nearly identical model performance compared to a traditional AL approach. Our approach applies without any adjustments to a rule-based system and any BBM that provides class labels (e.g., a classification model).
2. With the addition of TL, we leverage unlabeled internal data for the downstream task and unlabeled external data through a pre-trained model as knowledge sources for the learning process. This enables us to reduce the annotation cost by 51% compared to a traditional AL approach and improve the model performance compared to the combination of AL-WS.

The remainder of this article is structured as follows. Section 2 presents related approaches and illustrates the difference in our work. Subsequently, we propose our approach in Section 3 and evaluate it in Section 4 within a use case. Finally, we conclude our work and present future challenges in Section 5.

2 Related Work

Since AL is the backbone of our approach, we focus on related work regarding combinations of AL-WS and AL-TL. To the best of our knowledge, there has been no attempt yet to enhance AL with WS and TL.

Active Learning and Weak Supervision. Similar to our approach, [24] and [2] combine AL and WS. However, in their approaches, human experts actively select and annotate instances to improve a generative model that converts one-hot-encoded into probabilistic labels. Moreover, the authors of [3] use this combination to improve the expert rules of a WS model with interactive user feedback. In contrast to our approach, these methods primarily focus on WS and try to improve it with AL techniques. Instead, we focus on an AL cycle and enhance it with WS to reduce the annotation cost. Additionally, these works require labeling functions that are created from scratch. We, on the contrary, can automatically leverage information from any existing BBM that generates class labels without necessarily designing labeling functions. This simplification saves the effort to decompose an existing BBM for a generative model and enables us to treat it as a WSA in an AL cycle.

In comparison, [7] and [28] follow a similar objective as we do since they also aim to enhance a traditional AL cycle with WS techniques to reduce human interaction. The authors of [7] assign a pseudo label for a given instance in a self-training setting if the classifier's predicted probability exceeds a certain threshold. Additionally, they automatically assign the majority class label of similar instances to all unlabeled instances in a cluster. Moreover, instead of annotating single instances, [28] use human labels to annotate a cluster of similar instances to reduce human effort. However, these works do not consider a BBM that generates class labels in a real-world setting. We automatically leverage this existing knowledge source through an annotator model, reducing the annotation cost in an AL cycle.

Active Learning and Transfer Learning. The authors of [27] combine AL and TL but from a different perspective. While we aim to improve AL with TL, they enhance TL by actively selecting the most-suitable instances for the source domain from the target domain. Furthermore, [14] actively fine-tune a pre-trained model based on the contribution of an instance for the feature representation and performance of a classification model on a target task to reduce the annotation cost. In contrast, we do not actively select instances in the TL process but enhance a classification and annotator model within an AL cycle with transferred knowledge. Additionally, [17] investigate how TL mitigates the random initialization cold start and reduces label queries. The authors of [33] also leverage available unlabeled data but through unsupervised feature learning at the beginning of an AL cycle and semi-supervised learning during the cycle. They employ unsupervised pre-training by clustering the features and train a semi-supervised model by generating pseudo-labels for unlabeled instances. This way, they improve the model's performance while requiring less labeled data [33]. In our approach, however, we apply unsupervised learning not only on existing internal data but also propose to utilize external knowledge sources with TL.

3 Proposed Approach

In Section 3.1, we first give a formal definition of our problem setting. Consequently, we describe our proposed approach in Section 3.2 as shown in Figure 1. We design a modular approach so that we can selectively combine AL with WS and TL. This enables us to compare the influence of the individual components on the model performance and annotation cost.

3.1 Problem Setting

Problem. We consider a classification problem where we have a D-dimensional instance that is described by a feature vector $\mathbf{x} \in \mathcal{X}$ where $\mathcal{X} = \mathbb{R}^D$ describes the feature space. An instance $\mathbf{x}$ is drawn independently from the same distribution and belongs to a ground truth class label $y \in \mathcal{Y}$ where the set $\mathcal{Y} = \{1, ..., C\}$ defines the space of all class labels and C is the number of classes. In a pool-based AL scenario, we are given an unlabeled pool data set $\mathcal{U}(t) \subseteq \mathcal{X}$ without class labels. At each cycle iteration $t \in \mathbb{N}$, we aggregate the most-useful instances $\mathbf{x}^*$ in a batch $\mathcal{B}(t) \subset \mathcal{U}(t)$ with the size $b \in \mathbb{N}$. These instances require labels for the next cycle $t+1$ that annotators provide. Therefore, we define a set of annotators $\mathcal{A} = \{\text{HA}, \text{WSA}\}$, where we treat the HA as omniscient, providing a costly ground truth class label and an available BBM as a WSA, providing an error-prone class label at a low cost. Besides the class labels y to train the classification model, we also add a binary agreement label $z \in \mathcal{Z}$ with the set $\mathcal{Z} = \{0, 1\}$ to every instance in a batch to train the annotator model. We determine z based on the agreement between the labels provided by the HA and the WSA. It represents which instances were correctly classified (1) or misclassified (0) by the WSA. This means that we have to retrieve the WSA label at every

selected instance. Thus, we denote the annotated batch as $\mathcal{B}^*(t) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ and the labeled data set as $\mathcal{L}(t) \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$.

Model training. We express the classification model (e.g., a NN) through its parameters at cycle iteration t as $\boldsymbol{\theta}_t$. This model is trained on the labeled data set $\mathcal{L}(t)$ where either the HA or the WSA provide the class label y. It maps an instance to a vector of class probabilities with $f^{\boldsymbol{\theta}_t} : \mathcal{X} \to \Delta_{C-1}$, where Δ_{C-1} is the $C-1$ probability simplex spanned by C classes. Given an instance $\mathbf{x} \in \mathcal{X}$, the classification model predicts the probability vector $\hat{\mathbf{p}} = f^{\boldsymbol{\theta}_t}(\mathbf{x})$. This vector corresponds to an estimate of the categorical distribution of the classes made by the model f^{θ_t}. Additionally, we describe the annotator model through its parameters $\boldsymbol{\omega}_t$ which result from training on the binary agreement label z of the labeled data set $\mathcal{L}(t)$. With the function $g^{\boldsymbol{\omega}_t} : \mathcal{X} \to [0, 1]$ the annotator model maps an instance $\mathbf{x} \in \mathcal{X}$ to a probability $\hat{q} = g^{\boldsymbol{\omega}_t}(\mathbf{x})$. Its task is to estimate the probability that the WSA can provide a true class label. Thus, both models receive the same input instances from $\mathcal{L}(t)$ but are trained either on class or binary agreement labels. Moreover, we denote the parameters extracted from a pre-trained model as $\boldsymbol{\phi}$. Since the pre-trained model is only trained once, these parameters are independent of the cycle iterations.

3.2 Proposed Cycle

Our proposed AL cycle is illustrated in Figure 1. In the following paragraphs, we will give a detailed explanation of the steps in our approach.

Step 1 - Initialize Cycle. Before the cycle starts, we fine-tune a pre-trained model on the unlabeled data $\mathcal{U}$ and all additional data that we do not consider for AL with unsupervised learning. This model supplies initial parameters $\boldsymbol{\phi}$ for the classification and annotator model and provides feature representations that are helpful for AL [33]. Thus, we do not randomly initialize the parameters of a model at each cycle iteration. In our case, we utilize a pre-trained language model to extract word embeddings for the downstream task. In the first step, ① at iteration t, the classification and annotator model are initially trained on a small labeled data set $\mathcal{L}(t)$ where the instances $\mathbf{x}$ are drawn randomly from the unlabeled pool data set $\mathcal{U}(t)$. Here, the HA provides the ground truth class labels, and the WSA the error-prone class labels allowing us to compute the binary agreement label, which is utilized for training the annotator model. After the initialization step, we assume to have a trained classification model with the parameters $\boldsymbol{\theta}_t$ and a trained annotator model with the parameters $\boldsymbol{\omega}_t$.

Step 2 - Select Batch. The cycle continues in step ② with the selection algorithm of the **AL** module. We approximate the utility of all instances from the unlabeled pool $\mathcal{U}(t)$ based on the entropy of the predicted probability of the classification model $f^{\boldsymbol{\theta}_t}$. Given a probability vector $\hat{\mathbf{p}}$, the entropy is defined as

$$H(\hat{\mathbf{p}}) = -\sum_{c=1}^{C} \hat{p}_c \ln \hat{p}_c. \tag{1}$$

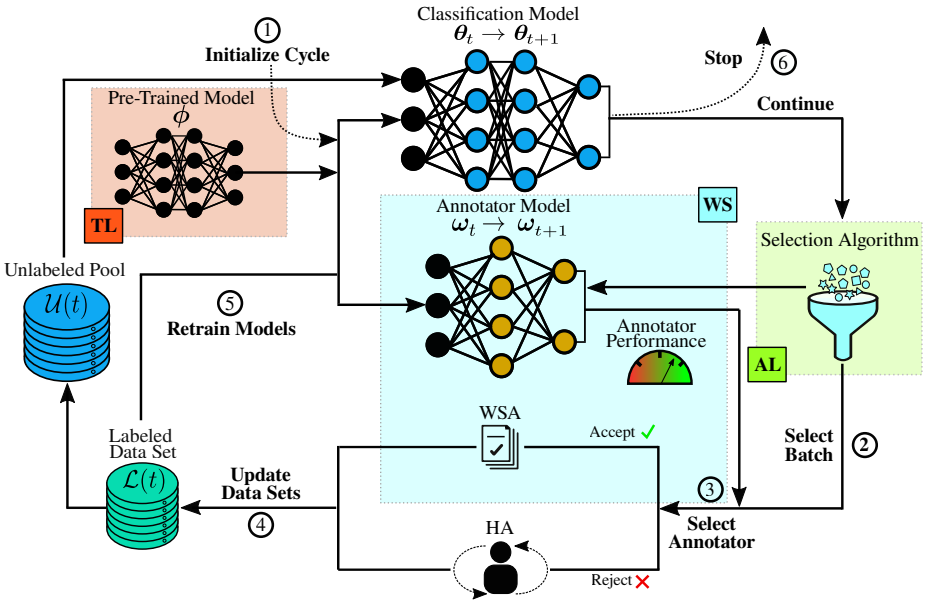


Fig. 1. A schematic illustration of the proposed AL cycle with WS and TL.

At cycle iteration t, we select the instance with maximum entropy according to

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathcal{U}(t)}{\arg\max}\, H\big(f^{\boldsymbol{\theta}_t}(\mathbf{x})\big)\,. \tag{2}$$

To aggregate a batch $\mathcal{B}(t) \subset \mathcal{U}(t)$, we greedily select the most-useful instances $\mathbf{x}^*$ until we reach the desired acquisition batch size $b \in \mathbb{N}$. We refer to this sampling strategy as max-entropy sampling.

Step 3 - Select Annotator. In step ③ with the **WS** module, we estimate the annotator performance of the WSA to decide whether it should provide the class labels for a specific instance. Therefore, we give each instance $\mathbf{x}^*$ of the selected batch $\mathcal{B}(t)$ to the annotator model $g^{\boldsymbol{\omega}_t}$ which estimates the probability $\hat{q}$. Intuitively, we interpret $\hat{q}$ as the probability that the WSA is capable of providing the ground truth class label. This way, the annotator model assesses the performance of the WSA. With the annotator performance estimation we decide whether to reject an error-prone class label of the WSA. In our approach, we investigate a simple reject function[1] that is based on threshold α and the estimated probability $\hat{q}$ as given by

$$r_\alpha(g^{\boldsymbol{\omega}_t}(\mathbf{x}^*)) = \begin{cases} 1, & \text{if } g^{\boldsymbol{\omega}_t}(\mathbf{x}^*) \geq \alpha \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

[1] It should be noted that more complex reject functions are available that could be the focus of future research.

If a class label of the WSA is rejected, the HA has to provide the true class label, enabling us to determine the binary agreement label z. However, suppose we decide that the WSA can provide a ground truth class label. In that case, the binary agreement label is set to 1 as a pseudo-label in the labeled pool. We refer to this as a pseudo-label because no ground truth is available. This technique can be considered semi-supervised learning [33].

Step 4 - Update Data Sets. In ④, we update the unlabeled pool data set $\mathcal{U}(t+1) = \mathcal{U}(t) \setminus \mathcal{B}(t)$ with the instances from the aggregated batch. Additionally, we update the labeled training set $\mathcal{L}(t+1) = \mathcal{L}(t) \cup \mathcal{B}^*(t)$ with the annotated batch including the class and the binary agreement labels.

Step 5 - Retrain Models. In ⑤, the classification and annotator model are re-trained from scratch simultaneously. Before training, we initialize the models' parameters with the parameters $\boldsymbol{\phi}$ we obtain from the unsupervised pre-trained model. This leads to an update of the model parameters $\boldsymbol{\theta}_{t+1}$ and $\boldsymbol{\omega}_{t+1}$.

Step 6 - Continue/Stop Cycle. At the end of an iteration, we decide in ⑥ whether to continue or stop the AL cycle with a stopping criterion. AL strategies in literature often use a simple pre-defined stopping criterion such as the desired size of the labeled pool or the maximum number of cycle iterations [20,31]. As this is not in the scope of this work, we choose the maximum number of instances as our stopping criterion.

4 Experimental Evaluation

In Section 4.1, we summarize the experimental setup for our use case. We design our experiments to enhance the AL cycle sequentially with the WS and TL modules to investigate their impact on model performance and annotation cost. The first experiments in Section 4.2 detail our findings where we enhance AL with WS to leverage an available BBM as an information source to reduce the annotation cost. Subsequently, Section 4.3 gives insights on how the addition of TL further improves our approach by utilizing internal and external unlabeled data with a pre-trained model as a knowledge source.

4.1 Experimental Setup

Use Case and Data. The data set in our use case consists of banking transactions. The goal is to predict an appropriate thematic class (e.g., household or insurance) based on short text descriptions of transactions with a NN. We do not have a labeled data set available, but the following information and knowledge sources are at our disposal:

1. **External Data**: Besides internal in-domain data for the downstream task, a vast amount of general-domain text data is available on the Internet [29].

As a pre-trained language model, we employ a fastText model [16] as a knowledge source. This model was trained on a general-domain corpus [8] and is available open-source[2]. We do not employ a deep transformer model in this preliminary investigation to avoid the issues of deep AL.

2. **Internal Data**: We leverage an extensive unlabeled data set with 7.7 million transactions to fine-tune the fastText model in an unsupervised manner with in-domain knowledge. To conduct the experiments efficiently, we randomly sample 9000 instances as the pool data set $\mathcal{U}$ and reserve 2000 instances with ground truth class labels for testing.
3. **Black-Box Model**: A rule-based system that classifies transactions with hand-crafted labeling rules is available. It was developed iteratively over several years by domain experts, and we consider it a BBM since the labeling rules are unavailable. We treat the BBM as the WSA that generates error-prone class labels at low cost. Preliminary studies show that it achieves an accuracy of approximately 86% on the test set.
4. **Human Annotator**: We assume a domain expert as an omniscient annotator that delivers ground truth class labels at a high cost. Specifically, the HA provides the class labels for the actively selected training instances when the label of the WSA is rejected and for the initialization step.

Models. The results are obtained by a classification model in our proposed AL cycle. The classification model is a multi-layer perceptron with an embedding layer to represent the text input with $D = 300$, a hidden layer with a ReLU activation function and an output layer with $C = 36$ neurons for each class. The annotator model is comprised of a similar structure, differing only in the output layer with $C = 1$ neuron as the annotator model solves a binary classification task. In each cycle iteration, we create a new vocabulary from the labeled pool and adapt the input layer of both models. We employ the Adam optimizer [18] to optimize the parameters, and the focal loss [22] as a loss criterion to address class imbalance. Additionally, we add dropout with 20% probability to the hidden neurons. We extract the static word embeddings from the pre-trained fastText model as initial weights of the embedding layers. This process can be considered as sequential TL [29].

Overall Experimental Design. To ensure comparability between our experiments, we define basic AL parameter configurations for all experiments. The configurations are generally based on results from preliminary studies in this use case. Specific settings for the experiments are highlighted in the corresponding sections. The initial labeled data set consists of 250 randomly sampled instances with ground truth labels provided by the HA. In preliminary work, this has proven to be a sufficient initial quantity of instances to enable the models to provide information to select the most-useful instances and suitable annotators. We set the desired size of the labeled data pool to 5370 as a pre-defined stopping criterion and the acquisition batch size b to 32 with 161 cycle iterations t. Our

[2] https://fasttext.cc/docs/en/crawl-vectors.html, accessed 2022-04-20

previous studies have shown that this relatively small number of instances leads to key results while enabling us to conduct experiments efficiently. We employ random sampling as a baseline sampling strategy and compare it to max-entropy sampling (Equation 2) for each experiment. Additionally, we decide between a costly (HA) or low-cost class label (WSA) based on our proposed reject option (Equation 3). Therefore, we define three different annotation scenarios to assess the influence of the WSA and the resulting annotation costs:

1. *full-human*: The HA provides the class labels for all of the selected instances, and we reject the class labels of the WSA. We consider this scenario a conventional AL approach without WS that should achieve the highest performance but generate the greatest baseline annotation cost.
2. *hybrid*: We select the WSA and the HA to provide the class labels based on the assessment of the annotator performance. In preliminary studies, 0.85 has proven to be a simple and promising reject threshold α, ensuring that we only accept labels of the WSA at high annotator performance estimations. At the same time, we ask the HA only for very uncertain instances to minimize annotation cost. Note that we must retrieve the class label of the WSA for every instance to determine the binary agreement label. This scenario reflects our approach combining AL with WS.
3. *full-WSA*: The WSA provides the class labels for all selected instances. This approach is the most inexpensive regarding the annotation cost, but we expect a deterioration of model performance. To ensure comparability, the HA still determines the ground truth class labels for the random initialization step.

As an exemplary cost scheme, we assign a cost of 1 to each annotation by the HA. Since the maintenance of the rule-based system as the BBM and automatically retrieving a class label also generates low cost, we assign 0.1 to an annotation of the WSA. Additionally, each experiment is repeated five times with different random seeds.

4.2 Experiments on AL with WS

This section shows the experimental results to answer research question 1. In these experiments, we utilize the HA and the available rule-based system as information sources with AL and WS.

Question 1. How can we enhance AL with WS so that we can leverage an available BBM as an information source to reduce the annotation cost with a competitive model performance compared to a traditional AL approach?

Findings. In Figure 2, we show the test accuracy and annotation cost for the aforementioned annotation scenarios and sampling strategies for each cycle iteration. Additionally, we report the final results in Table 1 after the AL cycle reaches the stopping criterion. The savings metric represents the cost saved relative to the highest baseline cost with conventional AL. As Figure 2 shows on

Table 1. Mean results (± standard error) of accuracy, annotation cost and savings of the AL cycle with different sampling strategies and annotation scenarios.

Sampling	Scenario	Accuracy(↑)	Cost(↓)	Savings(↑)
random	full-human	0.849±0.001	5370	0
	hybrid	0.842±0.001	1842±221	0.66
	full-wsa	0.823±0.004	762	0.86
max-entropy	full-human	0.873±0.001	5370	0
	hybrid	0.872±0.002	3045±46	0.43
	full-wsa	0.842±0.002	762	0.86

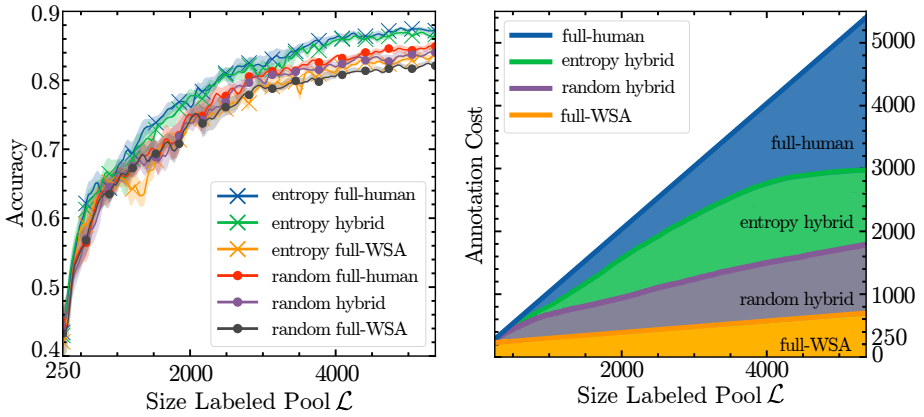


Fig. 2. Test accuracy and annotation cost with increasing size of the labeled data pool in the AL cycle with different sampling strategies and annotation scenarios.

the right, the annotation costs for the annotation scenarios *full-human* (highest baseline annotation cost and traditional AL) and *full-wsa* (lowest annotation cost without HA) are constant and independent of the sampling strategy. The former cost is identical to the size of the labeled pool since the HA provides labels for each instance. For the latter cost, only the initial labels are provided by the costly HA while the WSA generates the remaining labels at a low cost. With our approach in the *hybrid* scenario, the annotation cost depends on a mix of HA and WSA annotations. More WSA labels are generally rejected when using max-entropy sampling compared to random sampling in our *hybrid* scenario. The savings in Table 1 demonstrate that we can save annotation costs of 43 % with max-entropy sampling and 66 % with random sampling compared to the baseline cost of 5370 in the *full-human* scenario. However, we can see that random sampling degrades test accuracy. We attribute this to the fact that we actively select instances where the classification model is most uncertain in each batch. These also seem to be instances where the annotator model is uncertain and, thus, we more frequently reject the error-prone WSA. Additionally, we can observe a decreasing slope of the green cost curve with max-entropy sampling in our *hybrid* scenario on the left side of Figure 2. This seems intuitive since the

high-entropy instances from the unlabeled pool also diminish with cycle iterations. Therefore, a bigger labeled pool as the pre-defined stopping criterion could lead to only a slight increase in annotation cost and more strongly emphasize the benefits of our approach. The slope of the purple cost curve further highlights this assumption as it is monotonously increasing with random sampling, where we draw instances without considering the uncertainty of the classification model.

When looking at the accuracy in Figure 2, we observe that the model performance with max-entropy sampling is consistently superior to random sampling in each annotation scenario. Table 1 supports this observation and shows a performance increase of up to 3 % in accuracy with AL. Accordingly, the classification model's accuracy grows more rapidly in each cycle iteration, and it reaches the highest test accuracy with max-entropy sampling in the *hybrid* and *full-human* annotation scenarios. This demonstrates how AL techniques enable us to obtain a better classification accuracy with the same number of labeled instances compared to random sampling. The worst classification accuracy is obtained by random sampling in the *full-WSA* scenario. Accordingly, the results deteriorate for both selection strategies when only the error-prone WSA provides the class labels. Even though we can obtain savings of 86 % in the *full-WSA* scenario, the accuracy of the BBM (rule-based system) limits the achievable test accuracy of the classification model. This emphasizes the importance of ground-truth class labels from HAs and, thus, strengthens our combined approach in the *hybrid* scenario. As we expect, the classification model provides the best accuracy in the *full-human* scenario with max-entropy sampling as the traditional AL approach. However, our approach in the *hybrid* scenario with max-entropy sampling delivers nearly identical test accuracy while reducing the annotation cost by 43 %, as seen by savings in Table 1. Our results show that while costly HAs are important, we can also leverage a BBM as an additional information source. These observations let us conclude that our combination of AL and WS greatly reduces the annotation cost with only a marginal performance loss compared to traditional AL.

4.3 Experiments on AL with WS and TL

In this section, we conduct experiments with our complete proposed approach to tackle the second research question. In addition to WS and AL, we leverage all of the available unlabeled data to train a language model, which serves as a sequential TL approach. We focus on the *hybrid* annotation scenario with and without pre-training. So, we assess the influence of using all available information and knowledge sources on the model performance and annotation cost.

Question 2. How far can the inclusion of TL to leverage unlabeled internal and external data as knowledge sources empower the combination of AL-WS and, thus, further reduce the annotation cost and improve the model performance?

Findings. Figure 3 shows the test accuracy and annotation cost for the aforementioned sampling strategies in the *hybrid* scenario with and without pre-training.

Table 2. Mean results (± standard error) of accuracy, annotation cost, and savings of the AL cycle in different annotation scenarios with and without TL

Sampling	Scenario	Accuracy(↑)	Cost(↓)	Savings(↑)
random	full-human	0.879±0.002	5370	0
	hybrid	0.876±0.002	1819±51	0.66
	full-wsa	0.840±0.003	762	0.86
max-entropy	full-human	0.894±0.003	5370	0
	hybrid	0.893±0.001	2652±40	0.51
	full-wsa	0.847±0.002	762	0.86

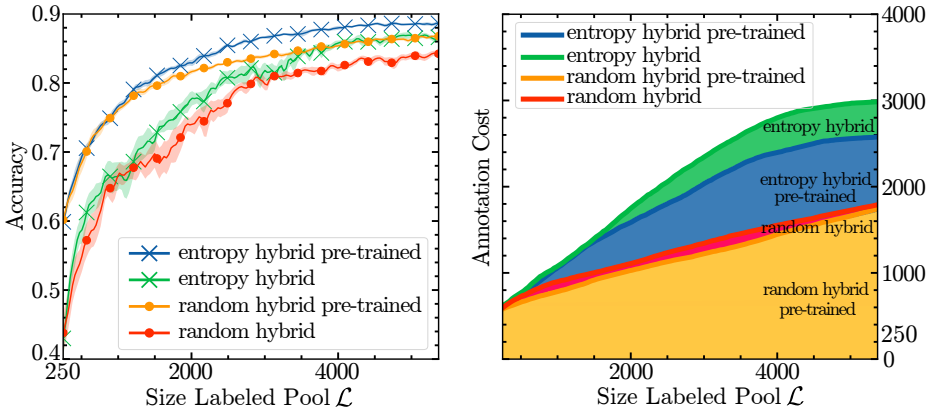


Fig. 3. Test accuracy and annotation cost with increasing size of the labeled data pool in the AL-WS cycle with and without TL.

In Table 2, we summarize the final results in all annotation scenarios with pre-training. We can see in Figure 3 that utilizing pre-trained weights gives the classification model a clear head start in performance. After initial training in the *hybrid* scenario, the model already reaches an accuracy of 60% for random (orange curve) and max-entropy (blue curve) sampling. This increase represents a 20% improvement over the green and red curves without pre-training. The advantage generally decreases with more training data but remains fundamentally intact and demonstrates the benefits of adding TL to our WS-AL approach, as also demonstrated in Table 2. We obtain the best results with the fastest accuracy increase in each iteration with pre-training and maximum-entropy sampling (blue curve). However, with the increasing size of the labeled data set, the accuracy of max-entropy sampling without pre-training adjusts to the same level of random sampling with pre-training. This means that max-entropy sampling has the same effect on the final model accuracy as leveraging the knowledge extractable from 7.7 million transactions and shows the general advantage of AL as the backbone of our approach. Table 2 further highlights the increase in accuracy in all annotation scenarios with TL compared to Table 1. Additionally,

the curves' trajectories with pre-training are more consistent with much less performance variance across the experiments' seeds.

On the right sight in Figure 3, we can see that the annotation cost with pre-training for max-entropy sampling is lower than without pre-training. Again, random sampling leads to lower annotation costs and poorer accuracy and confirms the benefits of using AL from the results above. Table 2 also highlights the improved savings of 51 % with the addition of TL compared to the baseline cost of 5370 with an 8 % increase relative to AL-WS. Moreover, we assume that the transferred knowledge improves the classification model's and the annotator model's certainty estimations. This means that pre-trained weights enable us to more efficiently select the most-useful instances and the low-cost annotations of the WSA. The results demonstrate the benefits of enhancing our AL-WS approach with TL by also leveraging available unlabeled data as a knowledge source with a pre-trained model. With our AL-WS-TL approach, we can improve the overall test accuracy of the classification model while further reducing the annotation cost.

5 Conclusion and Future Work

This work presented a novel approach to extending AL with WS and TL to reduce the annotation cost by leveraging multiple information and knowledge sources. We treated an established BBM (e.g., a rule-based system) as a weakly-supervised annotator that provides error-prone class labels inexpensively. This assumption made it possible to estimate the performance of this information source with an annotator model to decide whether a costly human annotation in an AL cycle is required. In a use case, we have successfully shown that enhancing AL with WS reduces annotation cost by 43% and leads to an almost identical model performance compared to traditional AL. Moreover, we leveraged unlabeled internal and external data as knowledge sources by fine-tuning a pre-trained language model on all available unlabeled data in an unsupervised manner. We then transferred this knowledge to expand our AL-WS cycle with TL. This enabled us to reduce the annotation cost by 51 % and improve the overall model performance compared to the AL-WS approach.

Since we applied our proposed approach for a shallow NN, we plan to move towards deep AL and the related problems in an application-oriented setting. To provide an accurate probabilistic estimation for the selection of instances, we aim to investigate the uncertainty estimates [15] of our classification and annotator models and calibrate them with methods such as temperature scaling [9] or scaling-binning [21]. Since we greedily acquired a batch of instances without batch-awareness, we intend to use a more complex selection strategy, such as BALD [6,19]. Moreover, we aim to enhance and further investigate the annotator model to measure the label quality of other information sources in the annotation process, such as the HA. Accordingly, we can move towards modern AL settings, where we also consider the HA as error-prone and can determine a more complex

cost scheme [11]. This could also be done in a multi-task learning setting by embedding the annotator model directly into the classification model.

Acknowledgments. This work results from the project INFINA, funded by Wirtschafts- und Infrastrukturbank Hessen under the Operational Program for the Promotion of Investments in Growth and Employment in Hessen which is financed by the European Regional Development Fund (ERDF).

References

1. Bach, S.H., Rodriguez, D., Liu, Y., Luo, C., Shao, H., Xia, C., Sen, S., Ratner, A., Hancock, B., Alborzi, H., Kuchhal, R., Ré, C., Malkin, R.: Snorkel DryBell: A Case Study in Deploying Weak Supervision at Industrial Scale. In: Proceedings of the 2019 International Conference on Management of Data. pp. 362–375 (2019). https://doi.org/10.1145/3299869.3314036
2. Biegel, S., El-Khatib, R., Oliveira, L.O.V.B., Baak, M., Aben, N.: Active weasul: Improving weak supervision with active learning. CoRR (2021). https://doi.org/10.48550/arXiv.2104.14847
3. Boecking, B., Neiswanger, W., Xing, E., Dubrawski, A.: Interactive weak supervision: Learning useful heuristics for data labeling. ICLR (2021)
4. Budd, S., Robinson, E.C., Kainz, B.: A Survey on Active Learning and Human-in-the-Loop Deep Learning for Medical Image Analysis. Medical Image Analysis **71**, 102062 (2021). https://doi.org/10.1016/j.media.2021.102062
5. Dunnmon, J.A., Ratner, A.J., Saab, K., Khandwala, N., Markert, M., Sagreiya, H., Goldman, R., Lee-Messer, C., Lungren, M.P., Rubin, D.L., Ré, C.: Cross-Modal Data Programming Enables Rapid Medical Machine Learning. Patterns **1**(2), 100019 (2020). https://doi.org/10.1016/j.patter.2020.100019
6. Gal, Y., Islam, R., Ghahramani, Z.: Deep bayesian active learning with image data. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70. pp. 1183—1192. ICML (2017)
7. Gonsior, J., Thiele, M., Lehner, W.: WeakAL: Combining Active Learning and Weak Supervision. In: Appice, A., Tsoumakas, G., Manolopoulos, Y., Matwin, S. (eds.) Discovery Science. pp. 34–49. Lecture Notes in Computer Science, Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-61527-7_3
8. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning word vectors for 157 languages. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018) (2018)
9. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning. pp. 1321–1330. ICML (2017)
10. Hanika, T., Herde, M., Kuhn, J., Leimeister, J.M., Lukowicz, P., Oeste-Reiß, S., Schmidt, A., Sick, B., Stumme, G., Tomforde, S., Zweig, K.A.: Collaborative Interactive Learning – A clarification of terms and a differentiation from other research fields. CoRR (2019). https://doi.org/10.48550/arXiv.1905.07264
11. Herde, M., Huseljic, D., Sick, B., Calma, A.: A survey on cost types, interaction schemes, and annotator performance models in selection algorithms for active learning in classification. CoRR (2021). https://doi.org/10.48550/arXiv.2109.11301

12. Hino, H.: Active learning: Problem settings and recent developments. CoRR (2020). https://doi.org/10.48550/arXiv.2012.04225
13. Holzinger, A., Plass, M., Kickmeier-Rust, M., Holzinger, K., Crişan, G.C., Pintea, C.M., Palade, V.: Interactive machine learning: Experimental evidence for the human in the algorithmic loop: A case study on Ant Colony Optimization. Applied Intelligence **49**(7), 2401–2414 (2019). https://doi.org/10.1007/s10489-018-1361-5
14. Huang, S.J., Zhao, J.W., Liu, Z.Y.: Cost-Effective Training of Deep CNNs with Active Model Adaptation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1580–1588. ACM (2018). https://doi.org/10.1145/3219819.3220026
15. Huseljic, D., Sick, B., Herde, M., Kottke, D.: Separation of aleatoric and epistemic uncertainty in deterministic deep neural networks. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 9172–9179 (2021). https://doi.org/10.1109/ICPR48806.2021.9412616
16. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of Tricks for Efficient Text Classification. CoRR (2016). https://doi.org/10.48550/arXiv.1607.01759
17. Kale, D., Liu, Y.: Accelerating Active Learning with Transfer Learning. In: 2013 IEEE 13th International Conference on Data Mining. pp. 1085–1090 (2013). https://doi.org/10.1109/ICDM.2013.160
18. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. CoRR (2017). https://doi.org/10.48550/arXiv.1412.6980
19. Kirsch, A., van Amersfoort, J., Gal, Y.: BatchBALD: Efficient and diverse batch acquisition for deep bayesian active learning. In: Advances in Neural Information Processing Systems (2019)
20. Kottke, D., Schellinger, J., Huseljic, D., Sick, B.: Limitations of Assessing Active Learning Performance at Runtime. CoRR (2019). https://doi.org/10.48550/arXiv.1901.10338
21. Kumar, A., Liang, P., Ma, T.: Verified uncertainty calibration. In: Advances in Neural Information Processing Systems (NeurIPS) (2019)
22. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal Loss for Dense Object Detection. CoRR (2018). https://doi.org/10.48550/arXiv.1708.02002
23. Liu, H., Gegov, A., Cocea, M.: Rule-based systems: A granular computing perspective. Granular Computing **1**(4), 259–274 (2016). https://doi.org/10.1007/s41066-016-0021-6
24. Nashaat, M., Ghosh, A., Miller, J., Quader, S., Marston, C., Puget, J.F.: Hybridization of Active Learning and Data Programming for Labeling Large Industrial Datasets. In: 2018 IEEE International Conference on Big Data (Big Data). pp. 46–55 (2018). https://doi.org/10.1109/BigData.2018.8622459
25. Paleyes, A., Urma, R.G., Lawrence, N.D.: Challenges in Deploying Machine Learning: A Survey of Case Studies (2021)
26. Pan, S.J., Yang, Q.: A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering **22**(10), 1345–1359 (2010). https://doi.org/10.1109/TKDE.2009.191
27. Peng, Z., Zhang, W., Han, N., Fang, X., Kang, P., Teng, L.: Active Transfer Learning. IEEE Transactions on Circuits and Systems for Video Technology **30**(4), 1022–1036 (2020). https://doi.org/10.1109/TCSVT.2019.2900467
28. Perez, F., Lebret, R., Aberer, K.: Weakly Supervised Active Learning with Cluster Annotation. CoRR (2019). https://doi.org/10.48550/arXiv.1812.11780
29. Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., Huang, X.: Pre-trained Models for Natural Language Processing: A Survey. Science China Technological Sciences **63**(10), 1872–1897 (2020). https://doi.org/10.1007/s11431-020-1647-3

30. Ratner, A., Bach, S.H., Ehrenberg, H., Fries, J., Wu, S., Ré, C.: Snorkel: Rapid training data creation with weak supervision. The VLDB Journal **29**(2), 709–730 (2020). https://doi.org/10.1007/s00778-019-00552-1
31. Ren, P., Xiao, Y., Chang, X., Huang, P.Y., Li, Z., Gupta, B.B., Chen, X., Wang, X.: A survey of deep active learning. ACM Comput. Surv. **54**(9) (2021). https://doi.org/10.1145/3472291
32. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2010)
33. Siméoni, O., Budnik, M., Avrithis, Y., Gravier, G.: Rethinking deep active learning: Using unlabeled data at model training. In: International Conference on Pattern Recognition (ICPR). pp. 1220–1227 (2021). https://doi.org/10.1109/ICPR48806.2021.9412716

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.

© 2022 for this paper by its authors. Use permitted under CC BY 4.0.

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.

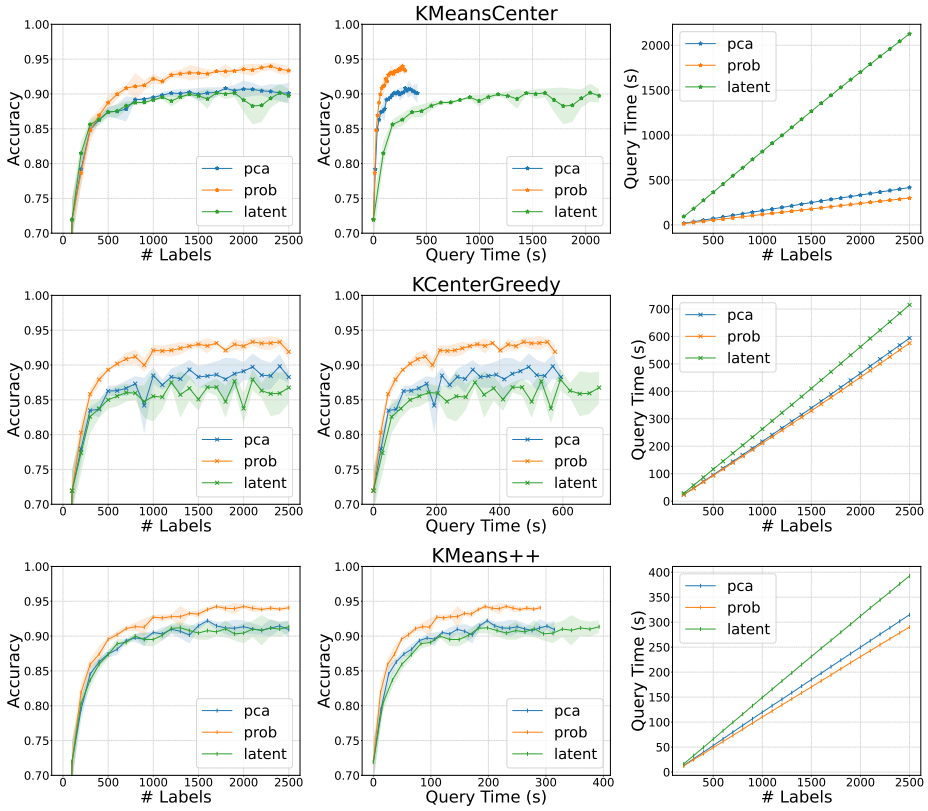


Fig. 1. Comparison of the different techniques for three different acquisition functions. The first column shows the accuracy w.r.t. the number of labels, the second column accuracy vs. cumulative query time, and the last column the cumulative query time vs. the number of acquired labels.

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)

A Practical Evaluation of Active Learning Approaches for Object Detection

Jan Schneegans[1], Maarten Bieshaar[2], and Bernhard Sick[1]

[1] Intelligent Embedded Systems, University of Kassel, Kassel, Germany
{jschneegans, bsick}@uni-kassel.de
[2] Robert Bosch GmbH, Corporate Research, Hildesheim, Germany
maarten.bieshaar@de.bosch.com

Abstract. The supervised training of deep learning models typically requires vast amounts of annotated data. With active learning, the annotation process can be made much more efficient by intelligently selecting the most valuable batches of samples to annotate and train on. Those samples are selected based on their utility regarding the training algorithm. In this work, we examine a wide range of such selection criteria for the task of object detection as performed by the widely applied Faster R-CNN model. We focus on the large and diverse BDD100K autonomous driving dataset, paying special attention to evaluate the model's performance regarding the dataset's meta information. Furthermore, we distinguish between approaches that select samples based on aleatoric or epistemic uncertainty. A selection of evaluation measures that cover specific error sources and the overall model performance suggests that there is little difference between the individual active learning approaches, even in regards to their specialized focus on different model parts and the object detection tasks of localization and classification. We conclude with a detailed discussion of the implied mechanisms regarding the active learning approaches that seem to affect model performances.

1 Introduction

Data annotation is costly both in time and resources (human and computational). The theoretical advantages of using active learning lie in a more efficient data annotation process by intelligently selecting a subset of samples that is thought to be most useful to train the machine learning model on. In this work, we perform a practical examination of active learning strategies, gaining insights into why certain approaches perform better than others regarding the task of 2D object detection. This is done on the very large BDD100K [1] dataset, which is one of the most diverse autonomous driving datasets in terms of scenarios, weather, uncommon objects, and other attributes, applying the popular Faster R-CNN model [2]. We describe the active learning process as a cycle of iteratively selecting a batch of samples to be annotated and training the Faster R-CNN model on the annotated portion of the data, cf. Figure 1. One batch consists of a set of images, each image containing one or more objects. The selection process consists of three parts: i) an utility function which estimates the

© 2022 for this paper by its authors. Use permitted under CC BY 4.0.

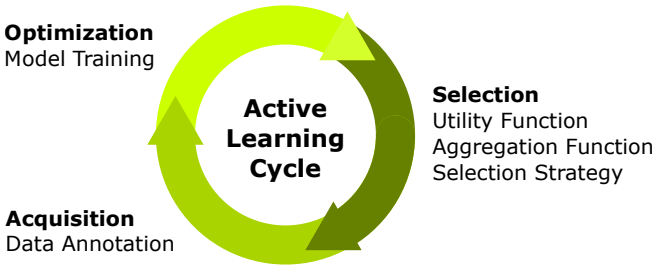


Fig. 1. The typical active learning cycle consisting of data selection, acquisition of annotations, and model training.

usefulness of each object, ii) an aggregation functions, which aggregates the usefulness for a complete image, and iii) a selection strategy, which selects the set of images deemed most useful. The examined variety of active learning strategies are based on the sub-tasks performed by the Faster R-CNN model, namely the separation of the annotated objects and the background, and the precise classification and localization of those objects. Furthermore, we consider and compare utility functions based on aleatoric and epistemic measures of uncertainty facilitated by Monte-Carlo dropout [3]. Multiple aggregation functions, e.g., mean and quantiles, are tested for each utility function to summarize the utilities over a whole image. We restrict the selection strategy to simply select the top k samples, i.e., images, with the highest utilities as aggregated from the individual object utilities for each image.

Our goal is to evaluate the practicality and benefits of utilizing active learning strategies in the training of large object detection models and to provide practical insights into the design of the corresponding machine learning pipeline. We present a novel utility function facilitated by the box predictions and their intersection-over-union (IOU) and experiment with approaches based on the different sub-tasks executed by the object detection model, i.e. utilizing the objectness, class, and box predictions of the model. We see a lack of a thorough, realistic and foremost practical evaluation of various active learning approaches for the task of object detection. In this work, we aim to close this gap and compare the actual annotation cost of each active learning approach and discuss the practicality and performance given the required computational effort. Moreover, we can get further insights into why certain active learning approaches perform better than others by examining the selection of meta-attributes, e.g. weather, time-of-day, scene, etc., of the BDD100K dataset.

In the following Section 2, we give an overview of active learning approaches as a whole and those specifically aimed at the task of object detection. Section 3 introduces our methodology, including the model setup and a thorough introduction of the examined active learning strategies. In Section 4, we provide further information regarding the dataset, experimental design, and evaluation measures. Section 5 discusses the experimental results, after which we summarize our findings and presents directions towards future work in Section 6.

2 Related Work

Active learning methods deal with the selection of data samples for annotation and subsequent model training. Much published work on the topic of active learning is concerned with proposing specific utility functions or selection criteria. Often these are based on a Bayesian Neural Network approach or Monte-Carlo sampling approaches through dropout [4]. Popular examples are uncertainty sampling [5], and entropy based ones [6], e.g. BALD [7] and Batch-BALD [8]. Siddahnt et al. [9] present a large-scale empirical study on deep active learning approaches, concluding that BALD can significantly outperform other approaches, using uncertainty estimates provided either by Dropout or Bayes-by-Backprop. Most techniques are made for classification tasks and only recently the spectrum of approaches was widened to encompass regression tasks [10,11,12,13]. For a survey on further aspects to consider in active learning, e.g. cost types and annotator performance, see [14]. As we do not consider any temporal information in the object detection tasks, we do not consider stream-based active learning methods, but instead focus on a variety of pool-based utility functions building on uncertainty estimation. Methods for query-synthesis, i.e. the generation of novel sample to annotate, are also beyond the scope of this work.

Advancing active learning methodologies towards more complex prediction tasks, e.g., object detection and localization, requires more sophisticated active learning approaches. Brust et al. [15] select images in an uncertainty-based approach using bounding box and class metrics. In [16], the uncertainties of both classification and bounding box predictions are utilized, as well. Roy et al. [17] use a query by committee approach and the disagreement between the convolutional layers in the object detector backbone. [18] investigates continual learning aspects of an ensemble-based method incorporating both classification and localization aspects for 2D and 3D object detection. Multiple Instance Active Learning for Object Detection [19] adapts an adversarial training procedure to select informative images for detector training by observing instance-level uncertainty, although, this approach implicitly assumes that there is a dominating object in each image, hence it can attach a single label to each images (as in image classification). Haussmann et al. [20] evaluate the use of active learning on a large scale object detection dataset for autonomous driving, although, with a different choice of models, active learning strategies, and dataset.

The Faster R-CNN model is one of the most widely used object detection model due its good performance and many readily available implementations. Since the original publication of the Faster R-CNN model many improvements to its architectural design were proposed [21]. Since most of these approaches add more complexity to the models with minuscule performance improvements, we only utilize an additional feature pyramid network [22], whose multi-scale feature maps will take part in our active learning approaches. Aghdam et al. [23] perform active learning for object detection by aggregating different pixel-level scores on the output of a convolutional neural network, which bears resemblance to our application of utility functions on the objectness maps predicted by the region proposal network inside the Faster R-CNN.

3 Methodology

This section describes the required preliminaries and individual parts of the applied machine learning model and introduces the examined active learning approaches. First, we briefly describe the applied object detection model, i.e. a modified Faster R-CNN, which we augment with dropout layers to perform uncertainty estimations, i.e. Monte-Carlo Dropout. Then the active learning approaches, consisting of individual utility functions, aggregation functions, and selection strategies are introduced.

3.1 Faster R-CNN

The Faster R-CNN model is one of the most widely used object detection models due to it's reliable performance and readily available implementations; but due to it's two-stage approach it is also one of the slowest. Accordingly, incorporating active learning in the training pipeline is a natural match to reduce training times. The Faster R-CNN consists of three main parts: a ResNet [24] *backbone*, a *region proposal network* (RPN), and the *classification and regression heads*. Fig. 2 shows the three components of the model, the augmentation via the dropout layers, as well as exemplary predictions for each sub-task utilized in the active learning approaches.

The backbone used for features extraction consists of a ResNet50 as implemented by the torchvision framework [25], followed by a feature pyramid network (FPN) [22] to better handle objects of different scales. The FPN extracts features at five different scales and three aspect ratios (1:1, 1:2, and 2:1), which are all input to the region proposal network.

The region proposal network consists of convolutional layers and performs a foreground and background classification and an initial rough localization of potential objects. Due to the use of the FPN, this is performed at five scales (32^2, 64^2, 128^2, 256^2, and 512^2 pixels) and three aspect ratios, resulting in a total of 15 *objectness* maps containing pixel wise binary classifications. The objectness is treated as one of three model outputs, which are further utilized in the active learning approaches.

Based on the binary classification performed by the RPN, a set of highest scoring object proposals is selected. Together with the features extracted by the backbone the selected object proposals are subsequently processed in separate heads for the final object classification and localization, i.e. box prediction. Those are the second and third model outputs on which the active learning approaches are applied.

The dimensions of the last few layers of the Faster R-CNN need to be adjusted to the specific learning problem posed by the dataset, i.e. the thirteen object classes considered. The dimensions of the final fully connected layers are adjusted accordingly. We start each experiment on a pretrained Faster R-CNN model on the COCO [26] object detection dataset to facilitate faster learning.

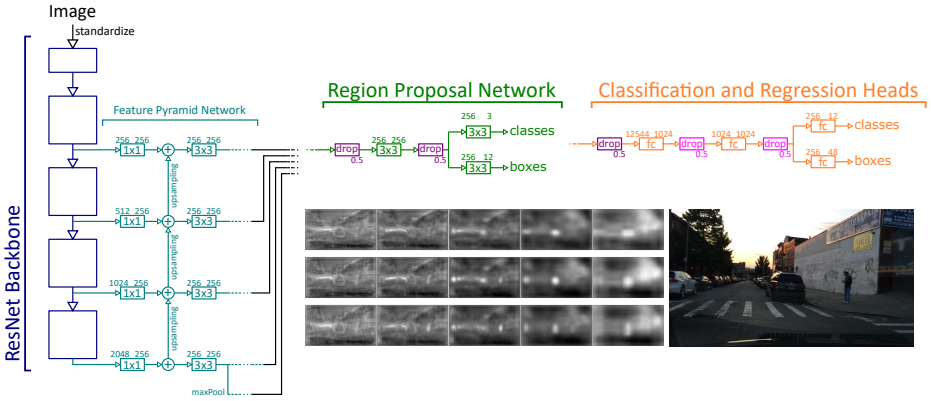


Fig. 2. Architecture of the Faster R-CNN model. The added dropout layers required to produce uncertainty estimates via Monte-Carlo dropout are shown in purple. The RPN is applied to all five scale dimensions output by the FPN individually. Exemplary predictions of the objectness (rows: aspect ratio, columns: scales), class and box predictions are shown. (We will colormap the objectness maps and depict matching model predictions in the final version)

Uncertainty Estimation Most active learning approaches are based on uncertainty estimates provided by a probabilistic model or sampled from an augmented model [27]. Typically a distinction between aleatoric and epistemic uncertainty is done [28,29]. Aleatoric uncertainty measures the uncertainty inherent in the data, produced by, e.g. noise, or in regards to the application it might also encompass sources of unpredictability such as motion blur or dirty lenses. Epistemic uncertainty measures the uncertainty of the model itself about its predictions and is typically harder to compute. To perform active learning this second kind of uncertainty is more useful, because one desires to select those samples, which the model struggles with, given the assumption that those samples provide the most benefit during training. The (pseudo-) probabilities produced by a neural network do not capture the epistemic uncertainty [28], therefore, the model architecture needs to be extended via an appropriate uncertainty estimation technique. We will utilize Monte-Carlo dropout as proposed in [4] and add respective dropout layers to the Faster R-CNN. More specifically they are added to the convolutional layers of the RPN and the classification and regression heads. Dropout refers to the CNN specific 2D variant that zeros-out entire channels, or in abstraction complete features. The model output can thus be sampled via multiple forward passes to estimate the epistemic uncertainty about the predictions. We draw 10 samples in each forward pass to maintain a reasonable inference time during the active learning cycles. The dropout layers are also kept active in those approaches that do not rely on the epistemic uncertainty estimates to avoid biasing the results, because we observed slightly lower performance while utilizing dropout, and we want to investigate the performance differences based on the utility functions and not due to adding dropout.

3.2 Active Learning Approaches for Object Detection

The term active learning encompasses strategies to select a subset of a given dataset with the goal of reducing costs in data annotation and model training. It does so in an iterative process of selecting and annotating data, and training on the available annotated data. We term one of those iterations as a *cycle*. Since we are working with the already annotated BDD100K dataset the annotation process is simulated by taking the annotations of the selected data into account.

For the application of object detection an active learning strategy consists of three main parts: a *utility function* that estimates the utility of an object or image, an *aggregation function* that aggregates the utilities of all objects in an image, and a *selection strategy* that selects the k most useful images.

Accordingly, one cycle consists of the model predicting object locations and classes, the application of the utility function, the application of the aggregations function, the application of the selection strategy, annotation of the selected data, i.e. moving data from the unlabeled dataset to the labeled dataset, retraining of the model based on the labeled dataset, and checking of a stopping criteria.

The initial condition (zeroth cycle) consist of an unlabeled dataset and the newly initialized (pretrained) model. Stopping criteria can be based on the amount of data that can be annotated, which might be restricted by available resources, e.g. financial budget, or based on the model performance, e.g., when a desired performance is reached or when the training saturates. Since we do not explicitly consider a fixed budget in the utility functions, we simply set the number of active learning cycles to 30 (based on the model convergence during the experiments) and compare different approaches based on the model performances over those iterations. The design of cost-sensitive utility functions, which explicitly consider the sample costs during estimation of their utility is still an open research-topic.

Utility Functions Given the model predictions about the object classes and locations, a utility function ascribes a usefulness to each object in the unlabeled dataset. Additionally, in case of the objectness predictions for an image, i.e. the feature maps showing the foreground-background classifications performed by the RPN, the utility of the entire image can be estimated directly. We further consider an approach utilizing all three predictions, i.e. objectness, class and location, combining multiple utility functions. The approaches based on the object classes consist of the following measures:

The normalized entropy provides a measure of the lack of model confidence based on the class predictions. It is obtained through normalization of the Shannon entropy [6] H over the maximum possible entropy $log(K)$, which is reached by a uniform distribution. Formally it is defined as

$$\eta(\hat{\mathbf{p}}) = \frac{H}{H_{max}} = -\sum_{k=1}^{K} \frac{\hat{p}_k \log(\hat{p}_k)}{\log(K)}, \tag{1}$$

where $\hat{\mathbf{p}}$ are the class predictions and K the number of classes. The predicted class probabilities $\hat{p}_k = \sigma_K(\mathbf{x})_c$ are given by applying the softmax function to the logits, i.e. the classification output of the model. The *normalized entropy* produces a high value when there is a strong disagreement between the different classes, i.e., when the distribution over the predicted classes approaches uniformity. Contrary, this entropy measure will be low, when there is a single class holding most of the distribution's mass, i.e., when the model in confident.

BALD [7], is also an entropy based measure. It aims to select samples that are expected to maximize the information gained about the model parameters [3]. BALD specifically utilizes the epistemic uncertainty of the model by sampling the model output via the applied Monte-Carlo dropout technique. The sampled predictions are clustered according to their location via an agglomerative clustering based on a distance threshold of 0.5 regarding their box IOU. The BALD utility function can then be applied to each cluster, i.e. each predicted object. We utilize a modified version of the BALD utility function that is normalized facilitating an unbiased combination of utility functions. Given a dataset $\mathcal{D}$ and model $\mathcal{M}$ with parameters ω_t as one of T random dropout configurations, the computationally tractable approximation of the utility function used during the experiments is formally given by

$$\alpha(\mathbf{x}|w) = \frac{1}{log(K)} \left(- \sum_k \left[\left(\frac{1}{T} \sum_t \hat{p}_k^t \right) log \left(\frac{1}{T} \sum_t \hat{p}_k^t \right) \right] + \frac{1}{T} \sum_{k,t} \hat{p}_k^t \, log(\hat{p}_k^t) \right) \tag{2}$$

where $\hat{p}_k^t$ is the probability of input $\mathbf{x}$ predicted by the model with parameters ω_t to take on class k, i.e., $\hat{p}_k^t = \sigma_K(\mathcal{M}_{\omega_t}(\mathbf{x}))_k$. p_k^t is given by the class predictions of the cluster. As with the entropy measure above, we normalize the BALD equation by the maximal possible entropy $log(K)$.

We will also apply both the normalized entropy and BALD utility functions to the objectness maps produced by the RPN.

As a utility function based on the object box regression, we propose a novel measure based on the Intersection-Over-Union (IOU). Similar to BALD we first need to cluster the proposed boxes per object before we calculate the IOU of each box within a cluster to the cluster mean, i.e., the mean box of the cluster. Subsequently those IOU values are averaged. Because we require the utility function to signify higher uncertainty with higher values we invert the expression by calculating 1 - the mean IOU. The *expected IOU* (eIOU) is thus formalized as

$$\bar{\mathbf{b}}_c = \frac{1}{|\mathcal{B}_c|} \sum_{\mathbf{b} \in \mathcal{B}_c} \mathbf{b} \tag{3}$$

$$\text{eIOU}(\mathcal{B}_c) = 1 - \frac{1}{|\mathcal{B}_c|} \sum_{\mathbf{b} \in \mathcal{B}_c} \text{IOU}(\mathbf{b}, \bar{\mathbf{b}}_c), \tag{4}$$

where $\mathcal{B}_c$ is the set of predicted boxes per cluster c, and $\bar{\mathbf{b}}_c$ the mean of the cluster. The expected IOU is normalized by definition, due to the IOU being normalized; this is advantageous compared to using the total or generalized variance of a cluster, because it is not influenced by the position and size of the object proposals. For the utility function should not be biased by those properties.

In order to evaluate a utility function utilizing all three model outputs, i.e., objectness, classes and boxes, we define a combined measure consisting of the normalized BALD approach applied to the objectness and the class, together with the eIOU.

The presented selection of utility functions comprise the most general and widely applied approaches based on estimated model uncertainties with the addition of a similarly inspired box-based version, i.e. the expected IOU. Having defined the utility functions, we can measure the utility per predicted object, or pixel in case of the objectness maps.

Aggregation Functions An aggregation function summarizes the output produced by a utility function to describe the utility of a complete sample, i.e. an entire image.

Intuitively the mean over the utility function output provides a measure of the average utility in annotating a certain sample. The median is not a good option as it is not influenced by outliers, e.g. objects the model is especially uncertain about, but we want the utility measure to be explicitly influenced by those parts of the image, assuming that these outliers are particularly interesting and useful.

Accordingly, applying the max function gives further priority to especially high values of the utilities produced by the utility function.

Although, simply applying the max aggregation function on the utility functions applied to the objectness maps would not work well due to fact that almost always the objectness maps contains maximum values of 1, thus every image would be ascribed the same maximum utility. To solve this issue we ignore those very high values by only considering the 95th and 99th percentiles.

Selection Strategies The selection strategy decides which of the samples from the unlabeled dataset are selected for annotation, given the aggregated utilities inferred through the application of a utility and aggregation function. We want to train the model on those samples that are deemed most useful, naturally, the selection strategy will simply consist of the max function over all unlabeled samples, selecting those samples with the maximum utility as aggregate per image. Depending on the utilized utility functions those samples are also the ones the model is most uncertain about.

4 Evaluation Methodology

This section details preliminary information regarding the dataset, the experimental setup, and the applied evaluation measures necessary to investigate the object detection as well as the active learning performances.

Dataset The experiments are performed on the BDD100K dataset, which is is one of the largest object detection datasets in the autonomous driving domain. It contains a variety of scenarios, sceneries, and annotated objects. Due to varying conditions such as time-of-day, weather, as well as noise, motion blur, and lens flares, the dataset poses a challenge towards current machine learning models. This naturally befits the use of active learning techniques to select different and useful samples, with the goal of reducing both annotation costs and training time.

There are five object categories with overall 13 classes: bike: bicycle, motorcycle; person: pedestrian, rider; vehicle: bus, car, truck; distractor: other person, other vehicle, trailer, train; signal: traffic light, traffic sign.

We perform experiments on all of the 13 classes as well as an easier subset of three classes summarized by bike, person, and vehicle, which supported the results on the larger set of classes. Additionally, we investigate the connection between the available meta-attributes with the active learning approaches to see if the approaches display certain preferences in selecting data samples in regards to these attributes: weather: clear, foggy, overcast, partly cloudy, rainy, snowy, undefined; scene: city street, gas stations, highway, parking lot, residential, tunnel, undefined; time-of-day: dawn/dusk, daytime, night, undefined; occluded: False, True; truncated: False, True.

Experimental Design Our goal is to evaluate the individual active learning approaches, consisting of combinations of the introduced acquisitions functions, aggregation functions, and selection strategy. For each of those combination we train a Faster R-CNN model for 30 active learning cycles. The pretrained model gets trained from scratch in each cycle as to not overfit on the data annotated the earliest, which we observed upon initial experimentation. Each experiment is performed twice, to make sure the experimental results and discussion thereof are reliable.

The BDD100K dataset contains 100.000 images, although, the annotations of the original test set are not available so we took the last 10.000 images from the training set to form an annotated test set. The splits are illustrated in Figure 3 with the original validation set, containing 10k images.

¡Through preliminary experiments the main hyper-parameters were determined. Those include: learning rate = 1e-5, batch size = 20, epochs = 10 (per cycle), and Mish activation functions. We utilize the Ranger optimizer [30] for faster convergence, which incorporates AdaBelief, RAdam, Lookahead, and Gradient Centralization. An acquisition size of 512, i.e. the number of acquired samples after each cycle, was determined to balance a reasonable annotation cost

Fig. 3. Training, test, and validation splits of the BDD100K dataset.

and training progress on the growing annotated dataset. This means the final models (at cycle 30) were trained on $30 * 512 = 15360$, which corresponds to 25.6% of the training data.

During training we utilize image augmentation by alteration of the brightness and contrast, or by adding Gaussian noise. The augmentations are applied with a random probability, order, and intensity (within previously determined bounds).

Performance Measures We distinguish between two kinds of measures that evaluate the object detection performance and further aid the investigation of the active learning approaches, respectively.

The most commonly applied evaluation measure for the task of object detection is the *mean Average Precision* (mAP). It describes the area under the precision-recall curve derived from the statistics of the model predictions. Since the mAP score only provides a single number, we additionally apply separate measures to evaluate each of 6 possible kinds of errors: class error, location error, class and location error, duplicate predictions, background prediction, missed objects, as proposed by [31]. A predicted box is considered correct if its IOU with the ground-truth box is higher than 0.5. To be able to compare class and location errors independently, an additional lower IOU threshold is needed, so that if a box is in the wrong location the classes can still be reasonably compared. This lower IOU threshold is set to 0.1, as suggested by [31]. Arguably, predictions recognized as class or location errors could also be counted as duplicate predictions if they can be matched to a ground-truth box, but since we want to count every prediction only once, only otherwise correct predictions count towards duplicate errors. By investigating the individual error sources, we can for example examine if approaches based on the predicted classes produce less classification errors, or if approaches based on the box predictions perform better in the localization sub-task.

To evaluate the performance of the active learning strategies, we are not only interested in the final model performances after 30 cycles, but also in the annotation costs (as measured by the number of annotated objects), which are often neglected in the current literature, and in the learning behavior over all active learning cycles. Notably, while the same number of samples, i.e., images, is selected in each cycle, different amounts of objects in the selected images lead to different annotation costs of the active learning approaches. Splitting the performance evaluation according to the available attributes is very useful to investigate whether a model performs better on samples that are considered more difficult, e.g., when time-of-day is night. Another assumption often made is that difficult samples are most useful to train on and that active learning approaches based on uncertainty measures are supposed to select those difficult samples. To investigate if this is the case, we sort all samples by the average mAP score over all models and compare them with the selection by the models.

5 Results

This section presents the results of the experiments and discusses the various insights into the applied active learning approaches. This encompasses three main parts: the learning behavior, the final model performances, and the acquisition characteristics.

The different approaches are each abbreviated by three letters, indicating: the type of sub-task predictions used for the active learning approach, the utility function, and the aggregation function. For example, `cbm` is the approach comprised of the BALD utility function applied to the predicted object classes and aggregated by the mean aggregation function. See all abbreviations in Table 1. `rdm` denotes random sampling, and `all` stands for the approach that utilizes all three kinds of predictions, applying the normalized BALD utility function to the class and objectness predictions, and the eIOU to the boxes.

Table 1. Active learning approaches abbreviations.

prediction		utility		aggregation	
c	class	e	entropy	m	mean
b	box	b	BALD	x	max
o	objectness	i	eIOU	5	95-percentile
a	all			9	99-percentile

We first compare the various active learning approaches by considering their performance on the test set. Fig. 4 shows the mAP performance for each cycle, averaged over both random seeds. Remember that each cycle adds 512 images to the labeled portion of the training set and in each cycle the models are trained anew. With this in mind we can see that training on more data does improve the model performances consistently for all approaches. Although, the convergent behavior is similar for all approaches, i.e. no approach provides considerably faster learning based on the selected data, and they all end up with around the same performance after 30 cycles. We performed the same set of experiments with a reduced set of annotation containing only three classes (vehicle, pedestrian, and cyclist), resulting in very much the same performance and learning behavior.

Second, we compare the performance of the trained models after 30 cycles. Fig. 5 shows the performance of each approach sorted by the mAP score on the test set, and the different error sources according to the TIDE measures [31]. Again, the numbers represent the average over both random seeds. Generally, we observe no significant differences between the active learning approaches. Most perform slightly better than random sampling. The approaches based on the objectness seem to perform best, for which the reason will be explored in the next subsection. There is no clear winner between the utility functions or the aggregation function. The performance of models trained via utility functions based on class or box predictions, do not show a clear correlation for their specific sub-tasks, as shown by the detailed evaluation of the different kinds of errors.

To investigate which kind of images each active learning approach acquires, we compare the selection for each model and for each cycle for the following object- and image-level attributes: class label, box size (5 bins, logarithmic),

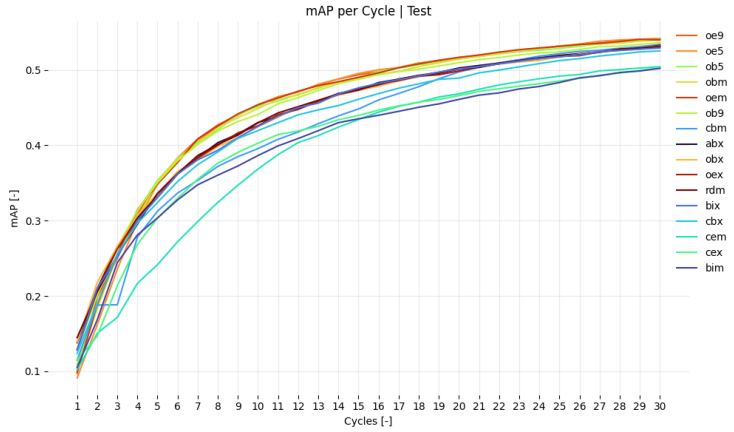


Fig. 4. Learning curves showing the mAP performances per cycle on the test set. All approaches show similar learning behavior, although, *cem*, *cex*, and *bim* perform slightly worse than the other approaches.

and the attributes presented by the BDD100K dataset. Additionally, we will assert if the approaches select especially difficult samples, as measured by the average mAP score of the models.

Fig. 6 shows the distribution of attributes in the training set (black dashed) from which the active learning approaches iteratively select a subset to train on. The attribute distributions of the selected subsets are shown for each cycle, whereby lower cycles are blue and higher cycles are red. One would assume, that the approaches should select a higher proportion of attributes that occur less often in the dataset, given the assumption that those samples are probably more difficult for the models. If this were the case, one should see a balancing between the individual instances of the attributes, respectively.

Regarding the label distributions, the approaches mostly adhere to the training distributions, with the exception of *bim* and *cbm*, which even exaggerate the label imbalances by selecting many images with cars in them. The approaches based on the objectness maps show more promising behavior by selecting less cars and more pedestrians, with the exception of the approaches utilizing the max aggregation function. The reason being, that they practically reduce to random random sampling due to the effect explained in Sec. 3.2. The box size selection is very consistent between every approach, oversampling small boxes. The weather selection looks very similar to the label attribute, with the *cem* approach also oversampling the most prominent weather instance (*clear*). Interestingly, the scene attribute shows an inverse behavior compared to the label and weather distributions. *bim* and *cbm* select more images showing *residential* and fewer showing *city street*. *cem* and *cex* have a similar preference towards *highway* scenes. Contrary to the label selection, the objectness based approaches oversample *city street* scenes and avoid *highway* scenes; we will discover why when we look at the number of acquired objects. Regarding the time-of-day, most ap-

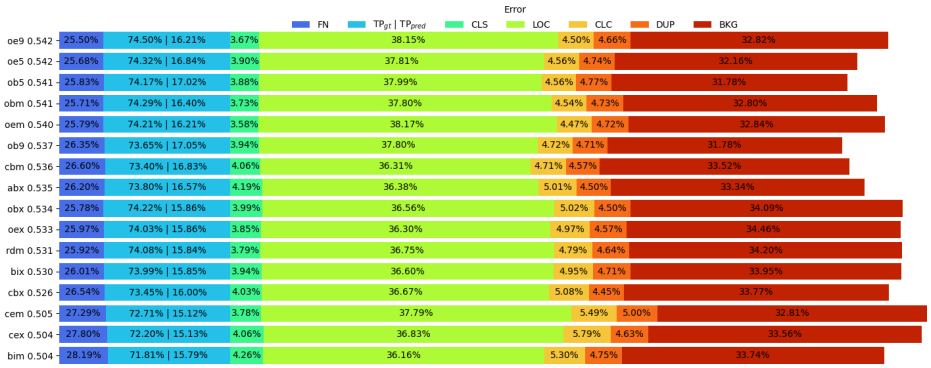


Fig. 5. Model performance after 30 cycles, sorted by the mAP score on the left. FN and TP_{gt} are in proportion to the number of ground truth, all else in proportion to the number of predictions. There is no score threshold applied to the predictions, which is why the percentage TP_{pred} seems relatively low. FN: false negatives, TP: true positives, CLS: class error, LOC: location error, CLS: class and location error, DUP: duplicate predictions, BKG: false positives/background predictions.

proaches exaggerate the day- and night-time imbalance in the distribution by selecting primarily day-time images, while some seem to prefer night-time images (*bim*, *cem*). The distributions of the occlusion and truncation attributes show no significant behavior, except for some approaches, apparent in the figure.

Contrary to the assumption, we generally observe little balancing and the attribute distributions of the selected images mostly vary around the training data distribution. We make the general observation that if there are deviations from the training distributions, they are more pronounced in the early cycles (blue), and the selection distributions are closing in on the training distributions towards later cycles (red), often matching them in the final cycles. For the attribute instances with very few sample we barely see any selection; enabling the approaches to oversample during sample selection might help in those cases. The attribute distributions of the data selected by random sampling (*rdm*, last row) is consistent with the overall data distribution, as expected.

Another kind of attribute often implicitly talked about is the difficulty of the samples, based on the assumption that more difficult samples are particularly useful for model training and should thus be selected by active learning approaches; especially by uncertainty based ones if we relate uncertainty to difficulty. To check this assumption we sorted all images in the training set according to their average mAP score over all models to estimate their difficulty.

Fig. 7 shows the four kinds of behaviors observed when we look at the image selections in each cycle over the mAP score. To create the depiction all 60k images in the training set, sorted by their average mAP score, were binned into 128 bins, shown along the x-axis (low to high mAP, from left to right). This includes the selection from both random seeds. The rows depict the cycles (early to late, from bottom to top). The approaches based on the class predictions and the BALD utility function, as well as the proposed box based eIOU approach,

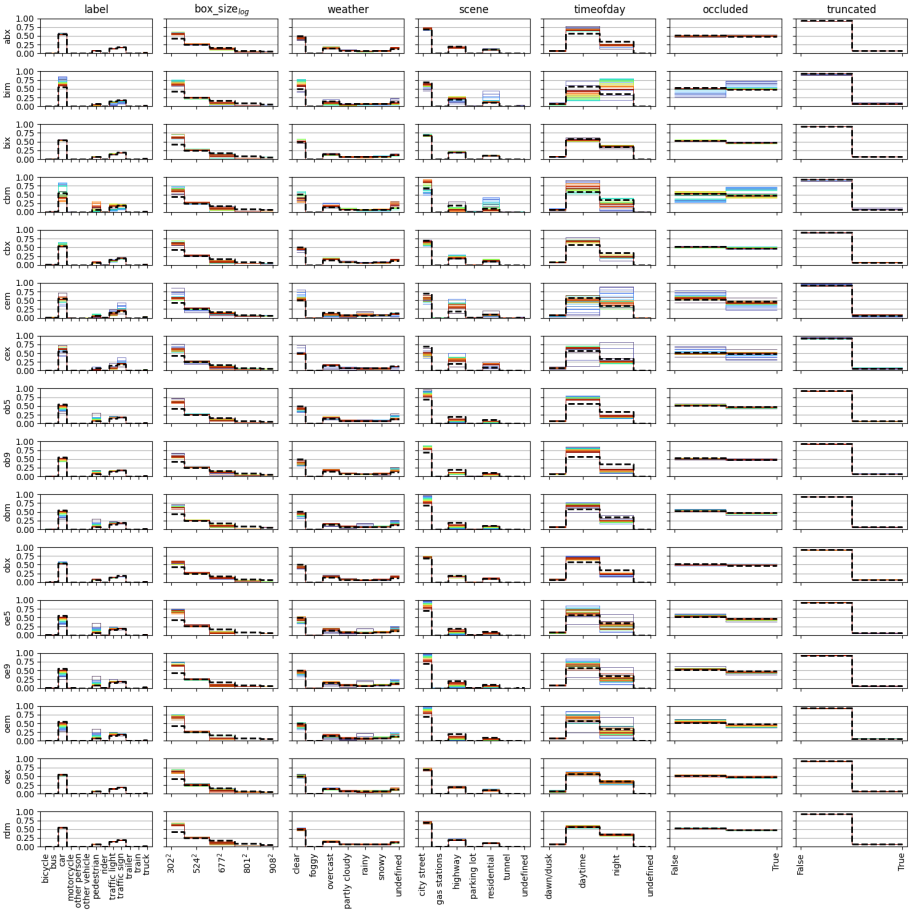


Fig. 6. Distributions of selected object and class level attributes compared to the training set distributions (black, dashed). For each active learning approach and attribute the distributions for all cycles and both random seeds are shown. One cycles consists of the attribute distribution of the 512 acquired images or the objects contained therein. Early to late cycles are colored blue to red.

select difficult examples in the earlier cycles, which then diffuses towards the region of average difficulty after the first few cycles. *cbx* maintained a slightly stronger preference towards high mAP scores throughout all cycles class-based approaches utilizing the normalized entropy utility function have a very strong tendency to sample very easy or very difficult images, as estimated by the mAP score. Here the focus tapers off towards later cycles as well. The approaches based on the objectness maps show a more independent distributions over the mAP score, with a slight tendency to not sample very difficult images. There is barely any variation over the cycles. Lastly, some approaches show similar random behavior as random sampling. Notably, all of those approaches use the

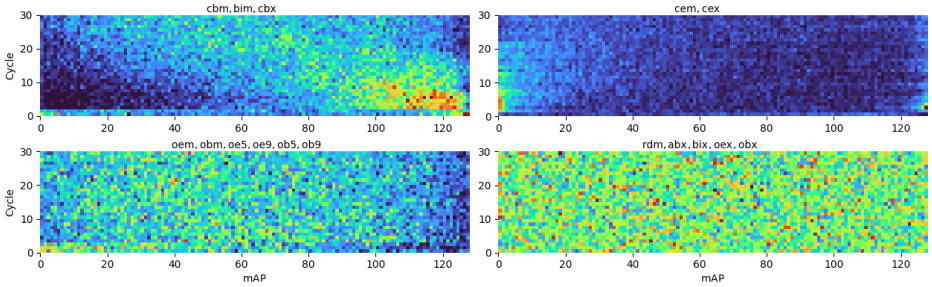


Fig. 7. The four kinds of selection behaviors expressed by the active learning approaches, in regards to the difficulty of the available images, estimated by the average mAP score of each image. It shows a two-dimensional histogram indicating the number of images in that region of the sorted mAP score. The mAP score is sorted from left to right, i.e. less difficult to most difficult.

max aggregation function. As already mentioned before, these approaches assign the same maximum utility to most images, leading to a random selection.

Finally, we compare the performance of each approach with the annotation costs of their acquired images. For this shows the actual practical applicability of the approaches. The performance is estimated by the mAP score of the trained models after 30 cycles. The annotation costs are estimated by the number of objects contained in the acquired images, because annotation companies usually calculate the annotation effort and costs per object. Fig. 8 shows both performance and the annotation costs for each approach, relative to random sampling. The approaches are sorted by their performance.

We observe a strong correlation between annotations costs, i.e. the number of objects, with the model performance. If we allow for a small decrease in performance compared to random sampling, the annotation costs can be reduced immensely, depending on the approach. In contrast, to achieve a slight increase in performance the annotation costs rise by about 40%. Notably, we see that the objectness based approaches consistently acquire images with more objects.

This is consistent with our previous observations that the objectness based approaches preferably select *city street* scenes with many *pedestrian* labels compared to the usually high number of *car* labels, during *daytime*. Otherwise, the results do not seem to correlate with the results depicted in Fig. 6 or Fig. 7. If we compare the annotation costs with the more detailed TIDE scores, we notice that the model performance correlates well with the overall amount of errors.

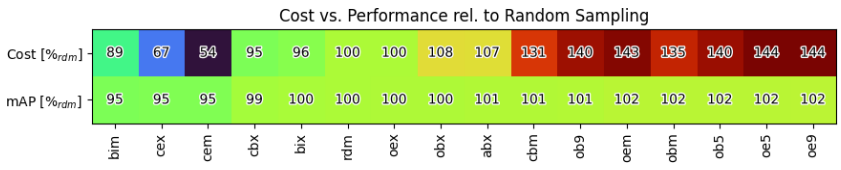


Fig. 8. cost vs performance rel. to random sampling

Overall, a high number of objects is beneficial to the model performance, and the objectness based approaches represent a proxy to find images containing many objects. We attribute it to the observation that the objectness maps produce high values around object edges, because there the uncertainty, whether a pixel in the objectness map corresponds to and object, is very high. Accordingly, the more objects are in the image, the more edges there will be, and the higher the aggregated utilities are. This naturally leads to the question, whether simply approaches like an edge detector could provide a good basis for active learning strategies, reducing the computationally effort by a large margin. To further note: random sampling takes drastically less compute effort, because one does not need to estimate the uncertainties, e.g. by sampling a model multiple times.

6 Conclusion

We applied a variety of active learning approaches to the task of object detection on a large and varied autonomous driving dataset. The approaches, comprising combinations of multiple utility functions and aggregations functions, utilized different kinds of model predictions based on the sub-tasks performed by the Faster R-CNN model. Overall, the approaches performed similarly, but showed some differences in how they functioned. An investigation into the attributes of the selected images lead to the observation that the objectness based approaches perform an elaborate proxy-task to estimate the number of objects per image. A main insight is the clear correlation between number of objects in the selected images and performance of the models trained on them.

It remains questionable if the uncertainty based approaches evaluated in this work justify the added complexity in the implementation and computational costs, compared to random sampling. Therefore, active learning approaches must further strive to be applicable to complex, real world datasets and difficult learning tasks such as object detection. Although, we discovered a promising direction of utilizing more primitive and efficient proxy-tasks, e.g. estimating the number of object per image, to base the active learning approaches on.

The assumption that, for example, night-time images are more difficult and should thus be selected by active learning approaches could not be confirmed. Which either means that the assumption is not true, which could be further verified by looking at the error scores of individual images with the respective attributes, or that the approaches simply do not select samples according to the assumption.

The experiments can be extended to include more datasets and a wider range of active learning approaches, since in the time past since the conduction of the experiments more utility functions and active learning strategies were proposed. Likewise, the application domain as well as the object detection task further warrant the additional use of other sensor modalities, e.g. Lidar or Radar. We also did not consider temporal information, e.g. video data, for stream based active learning.

7 Code Availability

The code base of this work is available to reproduce, verify, or extend the experiments conducted for this work under `https://git.ies.uni-kassel.de/public_code/a_practical_evaluation_of_active_learning_approaches_for_object_detection`.

8 Acknowledgment

This work results from the project KI Data Tooling (19A20001O) funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK).

References

1. F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2633–2642, 2020.
2. S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 2015.
3. Y. Gal, R. Islam, and Z. Ghahramani, "Deep Bayesian active learning with image data," in *34th International Conference on Machine Learning, ICML 2017*, vol. 3, 2017, pp. 1923–1932.
4. Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Appendix," in *33rd International Conference on Machine Learning, ICML 2016*, vol. 3, 2016, pp. 1661–1680.
5. D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994*, 1994, pp. 3–12.
6. C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 6, 10, pp. 379–423, 623–656, 1948.
7. N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, "Bayesian Active Learning for Classification and Preference Learning," *ArXiv*, vol. abs/1112.5, 2011.
8. A. Kirsch, J. van Amersfoort, and Y. Gal, "BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
9. A. Siddhant and Z. C. Lipton, "Deep Bayesian Active Learning for Natural Language Processing: Results of a Large-Scale Empirical Study," *ArXiv*, vol. abs/1808.0, 2018.
10. D. Wu, C. T. Lin, and J. Huang, "Active learning for regression using greedy sampling," *Information Sciences*, vol. 474, pp. 90–105, 2019.
11. D. Wu, "Pool-Based Sequential Active Learning for Regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, pp. 1348–1359, 2019.
12. C. Käding, E. Rodner, A. Freytag, O. Mothes, B. Barz, and J. Denzler, "Active learning for regression tasks with expected model output changes," in *British Machine Vision Conference 2018, BMVC 2018*, 2019.

13. J. Goetz, A. Tewari, and P. Zimmerman, "Active Learning for Non-Parametric Regression Using Purely Random Trees," in *NeurIPS*, 2018.
14. M. Herde, D. Huseljic, B. Sick, and A. Calma, "A survey on cost types, interaction schemes, and annotator performance models in selection algorithms for active learning in classification," *IEEE Access*, vol. 9, pp. 166 970–166 989, 2021.
15. C. A. Brust, C. Käding, and J. Denzler, "Active learning for deep object detection," in *VISIGRAPP 2019 - Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, vol. 5, 2019, pp. 181–190.
16. C. C. Kao, T. Y. Lee, P. Sen, and M. Y. Liu, "Localization-Aware Active Learning for Object Detection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11366 LNCS, 2019, pp. 506–522.
17. S. Roy, A. Unmesh, and V. P. Namboodiri, "Deep active learning for object detection," in *British Machine Vision Conference 2018, BMVC 2018*, 2019.
18. S. Schmidt, Q. Rao, J. Tatsch, and A. Knoll, "Advanced active learning strategies for object detection," in *Intelligent Vehicles Symposium (IV)*, Las Vegas, NV, 2020, pp. 871–876.
19. T. Yuan, F. Wan, M. Fu, J. Liu, S. Xu, X. Ji, and Q. Ye, "Multiple instance active learning for object detection," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, virtual, 2021, pp. 5330–5339.
20. E. Haussmann, M. Fenzi, K. Chitta, J. Ivanecký, H. Xu, D. Roy, A. Mittel, N. Koumchatzky, C. Farabet, and J. M. Alvarez, "Scalable Active Learning for Object Detection," in *IEEE Intelligent Vehicles Symposium (IV)*, Las Vegas, NV, USA, 2020, pp. 1430–1435.
21. Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving Into High Quality Object Detection," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, 2018.
22. T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, 2017, pp. 936–944.
23. H. Habibi Aghdam, A. Gonzales-Garcia, A. M. Lopez, and J. van de Weijer, "Active learning for deep detection neural networks," in *International Conference on Computer Vision (ICCV)*, Seoul, South Korea, 2019, pp. 3671–3679.
24. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, 2016, pp. 770–778.
25. S. Marcel and Y. Rodriguez, "Torchvision the machine-vision package of torch," in *Proceedings of the 18th ACM International Conference on Multimedia*, ser. MM '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 1485–1488.
26. T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014.
27. P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, X. Chen, and X. Wang, "A survey of deep active learning," *ACM Computing Surveys (CSUR)*, vol. 54, pp. 1 – 40, 2022.
28. D. Huseljic, B. Sick, M. Herde, and D. Kottke, "Separation of aleatoric and epistemic uncertainty in deterministic deep neural networks," in *25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 9172–9179.

29. A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Conference on Neural Information Processing Systems (NIPS)*, Long Beach, CA, 2017, pp. 5574–5584.
30. L. Wright, "Ranger-deep-learning-optimizer," 2020.
31. D. Bolya, S. Foley, J. Hays, and J. Hoffman, "Tide: A general toolbox for identifying object detection errors," *ArXiv*, vol. abs/2008.08115, 2020.

Certifiable Active Class Selection in Multi-Class Classification

Martin Senz[0000−0002−9377−3939], Mirko Bunse[0000−0002−5515−6278], and Katharina Morik[0000−0003−1153−5986]

TU Dortmund University, Artificial Intelligence Group, D-44227 Dortmund, Germany
{martin.senz, mirko.bunse, katharina.morik}@tu-dortmund.de

Abstract. Active class selection (ACS) requires the developer of a classifier to actively choose the class proportions of the training data. This freedom of choice puts the trust in the trained classifier at risk if the true class proportions, which occur during deployment, are subject to uncertainties. This issue has recently motivated a *certificate* for ACS-trained classifiers, which builds trust by proving that a classifier is sufficiently correct within a specific set of class proportions and with a high probability. However, this certificate was only developed in the context of binary classification. In this paper, we employ Hölder's inequality to extend the binary ACS certificate to multi-class settings. We demonstrate that our extension indeed provides correct and tight upper bounds of the classifier's error. We conclude with several directions for future work.

Keywords: Active class selection · Prior probability shift · Multi-class classification · Model certification · Learning theory · Validation.

1 Introduction

The proceeding deployment of machine learning models in real-world applications increases the importance of validating these models thoroughly. Ideally, the robustness of these models against distribution shifts [5] is *certified* in the sense of being formally proven or extensively tested [3].

In active class selection [4], a class-conditional data generator is repeatedly asked to produce feature vectors for arbitrarily chosen classes. In this setting, the developer of a classifier must actively decide for the class proportions in which the training data set is produced. While this freedom can reduce the data acquisition cost while improving classification performance, it also puts the trust in the trained classifier at risk: what if the class proportions, which occur during deployment, are not precisely known or are even subject to changes?

These uncertainties have motivated a *certificate* for ACS-trained classifiers, which declares a set of class proportions to which a classifier is safely applicable [2]. In particular, the certified classifier is required to exhibit an ACS-induced error of at most some $\epsilon > 0$, with a probability of at least $1 - \delta$. However, this certificate was only developed in the context of binary classification; a multi-class certificate has not yet been proposed, to the best of our knowledge.

© 2022 for this paper by its authors. Use permitted under CC BY 4.0.

In this paper, we close the gap between ACS model certification and multi-class classification. In the following, we recapitulate the theoretical background of binary ACS certification in Section 2 before we develop our multi-class ACS certificate in Section 3. We validate our claims empirically in Section 4 before we conclude with Section 5.

2 Theoretical Background

The term "domain", as used in domain adaption [6], describes a probability density function over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the feature space and $\mathcal{Y}$ is the label space. In ACS, we assume that the *source domain* $\mathcal{S}$, where a machine learning model is trained, differs from the target domain $\mathcal{T}$, where the model is deployed, only in terms of the class proportions $p_\mathcal{S} \neq p_\mathcal{T}$ [2]. Such deviations, also known as target shift [8] or as prior probability shift [5], occur due to the freedom of choosing any $p_\mathcal{S}$ for the acquisition of training data. We are interested in the impact of such deviations on the classification performance with respect to $\mathcal{T}$.

Recently, a PAC learning perspective [2] on this setting has provided us with Theorem 1. This result quantifies the difference in loss values $L(h)$ between an ACS-generated training set D and the target domain $\mathcal{T}$. Only if this difference is small, we can expect to learn a classifier h from D that is accurate also with respect to $\mathcal{T}$, similar to standard PAC learning theory. The key insight of this theorem is that the relevant loss difference between D and $\mathcal{T}$ is continuously approaching the inter-domain gap $|L_\mathcal{T}(h) - L_\mathcal{S}(h)|$, which is independent of the random draw of D from $\mathcal{S}$, while the training set size m increases. In ACS, this increase happens naturally while more and more data is actively being acquired, so that the error of any ACS-trained classifier is increasingly dominated by this gap. Since the inter-domain gap is constant with respect to the random draw of the training set D, it is also independent of ϵ, δ, and m.

Theorem 1 (Identical mechanism bound [2]). *For any $\epsilon > 0$ and any fixed $h \in \mathcal{H}$, it holds with probability at least $1 - \delta$, where $\delta = 4e^{-2m\epsilon^2}$, that*

$$|L_\mathcal{T}(h) - L_\mathcal{S}(h)| - \epsilon \;\leq\; |L_\mathcal{T}(h) - L_\mathcal{D}(h)| \;\leq\; |L_\mathcal{T}(h) - L_\mathcal{S}(h)| + \epsilon.$$

This theorem can be used to certify a trained classification model h with N classes in terms of a set of safe class proportions $\mathcal{P} \subseteq [0,1]^N$. By "safe", we mean that, during the deployment of h on $\mathcal{T}$, the trained model induces, with a high probability, at most a small domain-induced error ϵ.

Definition 1 (Certified hypothesis [2]). *A hypothesis $h \in \mathcal{H}$ is certified for all class proportions in $\mathcal{P} \subseteq [0,1]^N$ if, with probability at least $1-\delta$ and $\epsilon, \delta > 0$,*

$$|L_\mathcal{T}(h) - L_\mathcal{S}(h)| \;\leq\; \epsilon \quad \forall\, \mathbf{p}_\mathcal{T} \in \mathcal{P}.$$

Let $\mathbf{p}_S$, $\mathbf{p}_\mathcal{T} \in [0,1]^N$ be vectors with components $[\mathbf{p}_\bullet]_i = \mathbb{P}_\bullet(Y = i)$, which express the probabilities of the class labels in the respective domains S and $\mathcal{T}$.

Furthermore, let $\boldsymbol{\ell}_h \in \mathbb{R}^N$ be a vector that represents the class-wise losses

$$[\boldsymbol{\ell}_h]_i \;=\; \ell_X(h,i) \;=\; \int_{\mathcal{X}} \mathbb{P}(X = \mathbf{x} \mid Y = i) \cdot \ell(h(\mathbf{x}), i)\, \mathrm{d}\mathbf{x}, \tag{1}$$

as according to some loss function ℓ. The total loss of the hypothesis h is then given by $L_\bullet(h) = \sum_{i \in \mathcal{Y}} [\mathbf{p}_\bullet]_i [\boldsymbol{\ell}_h]_i = \langle \mathbf{p}_\bullet, \boldsymbol{\ell}_h \rangle$. Consequently the inter-domain gap for classification problems can be expressed as

$$\begin{aligned} |L_\mathcal{T}(h) - L_S(h)| \;&=\; |\langle \mathbf{p}_\mathcal{T}, \boldsymbol{\ell}_h \rangle - \langle \mathbf{p}_\mathcal{S}, \boldsymbol{\ell}_h \rangle| \\ &=\; |\langle \mathbf{p}_\mathcal{T} - \mathbf{p}_\mathcal{S}, \boldsymbol{\ell}_h \rangle| \\ &=\; |\langle \mathbf{d}, \boldsymbol{\ell}_h \rangle|, \end{aligned} \tag{2}$$

where $\mathbf{d} = \mathbf{p}_\mathcal{T} - \mathbf{p}_\mathcal{S}$ is the difference between the class probabilities in the domains $\mathcal{S}$ and $\mathcal{T}$.

In order to certify classification models, it is necessary to calculate Eq. 2. However, the true class-wise losses $\boldsymbol{\ell}_h$ are unknown, and we can only estimate the empirical class-wise losses $\hat{\ell}_X(h,y) = \frac{1}{m_y} \sum_{i:y_i=y} \ell(y, h(\mathbf{x}_i))$ from a finite amount of labeled validation data. Therefore, our goal is to constrain Eq. 2 with the smallest upper bound, which holds with a high probability.

For binary classification problems, the inter-domain gap can be factorized into a product of two scalars, $\Delta p \cdot \Delta \ell_X$. Here, $\Delta p = |p_\mathcal{T} - p_\mathcal{S}| \in \mathbb{R}$ denotes the difference between class proportions and $\Delta \ell = |\ell_{Y=2}(h) - \ell_{Y=1}(h)| \in \mathbb{R}$ denotes the difference between class-wise losses. A smallest upper bound $\Delta \ell^*$, which holds with probability $1 - \delta$, can be found for the empirical estimate $\Delta \hat{\ell}$. Therefore, by Def. 1, binary classifiers can be certified as a function of ϵ and δ, where $\mathcal{P}$ is characterized by the range $[p_\mathcal{T}^{\min}, p_\mathcal{T}^{\max}]$ of class proportions [2].

3 Certification in Multi-Class Classification

To certify multi-class classifiers according to Def. 1, an estimation for the inter-domain gap with multiple classes must be found. For this purpose, we will make use of Hölders inequality [7], a fundamental inequality theorem for the study of L^p spaces. This inequality will help us in using PAC bounds for multi-class certification, similar to the certification of binary classifiers.

Theorem 2 (Hölder's inequality [7]). *Let (S, Σ, μ) be a measure space and let $p, q \in [1, \infty]$ with $1/p + 1/q = 1$, where $1/\infty = 0$. Then, for all measurable real- or complex-valued functions f and g on S,*

$$\|fg\|_1 \leq \|f\|_p \|g\|_q. \tag{3}$$

With this inequality, the inter-domain gap from Eq. 2 can be transformed to

$$|\langle \mathbf{d}, \boldsymbol{\ell}_h \rangle| \;\leq\; \begin{cases} \|\mathbf{d}\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty, & \text{for } p = 1, q = \infty \\ \|\mathbf{d}\|_2 \cdot \|\boldsymbol{\ell}_h\|_2, & \text{for } p = 2, q = 2 \\ \|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1, & \text{for } p = \infty, q = 1 \end{cases} \tag{4}$$

In the following we restrict ourselves to the consideration of the Hölder conjugate $p = \infty, q = 1$. In principle, the other conjugate forms are also applicable. However, we will see that the infinity norm on $\mathbf{d}$ provides a simple and intuitive characterization of $\mathcal{P}$.

In order to yield a certified hypothesis, as according to Def. 1, it must hold, with a probability of at least $1 - \delta$, that, for $\epsilon, \delta > 0$,

$$|L_{\mathcal{T}}(h) - L_{\mathcal{S}}(h)| \;\leq\; \|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1 \;\leq\; \epsilon \quad \forall\, \mathbf{p}_{\mathcal{T}} \in \mathcal{P}. \tag{5}$$

Like in the binary setting, only the empirical class-wise loss $\hat{\boldsymbol{\ell}}_h$ is given. Hence, a minimum upper bound $\|\boldsymbol{\ell}_h\|_1^*$ for the norm $\|\hat{\boldsymbol{\ell}}_h\|_1$, that is valid with a probability of at least $1 - \delta$, must be found. Each $\hat{\ell}(h, y)$ is associated with a positive corresponding error ϵ_y with $\delta_y = e^{-2m_y\epsilon_y^2}$. For a given probability budget of δ, we get the smallest upper bound $\|\boldsymbol{\ell}_h\|_1^* = \|\hat{\boldsymbol{\ell}}_h\|_1 + \sum_{y=1}^N \epsilon_y^*$ by minimizing $\sum_{y=1}^N \epsilon_y$ through the optimization problem

$$\min_{\epsilon_1,\dots\epsilon_N \in \mathbb{R}} \sum_{y=1}^N \epsilon_y, \quad \text{s. t.} \begin{cases} \epsilon_1, \dots \epsilon_N & \geq \tau \\ \delta - \sum_{y=1}^N \delta_y = e^{-2m_y\epsilon_y^2} & \geq 0 \end{cases}, \tag{6}$$

where strict inequalities are realized through non-strict inequalities with some sufficiently small $\tau > 0$.

Let us now describe the set $\mathcal{P}$ of safe class proportions. In extension to the requirement given in Def. 1, $\mathcal{P}$ is supposed to cover all class proportions that are valid according to the certificate. With the minimum upper bound $\|\boldsymbol{\ell}_h\|_1^*$, we can rearrange Eq. 5 to

$$\|\mathbf{d}\|_\infty \;\leq\; \frac{\epsilon}{\|\boldsymbol{\ell}_h\|_1^*} \quad \forall\, \mathbf{p}_{\mathcal{T}} \in \mathcal{P}. \tag{7}$$

By taking the infinity norm on $\mathbf{d}$, $\|\mathbf{d}\|_\infty$ reduces to the class i which has the largest absolute *label distribution shift* $|[\mathbf{p}_{\mathcal{T}}]_i - [\mathbf{p}_{\mathcal{S}}]_i| = \Delta p$. In analogy to the binary certification, the range of safe deployment proportions for a class i can be described by $[\ [\mathbf{p}_S]_i - \Delta p^*\ ,\ [\mathbf{p}_{\mathcal{S}}]_i + \Delta p^*] = [p_{\mathcal{T},i}^{\min}, p_{\mathcal{T},i}^{\max}]$. Here, $\Delta p^* = \frac{\epsilon}{\|\boldsymbol{\ell}_h\|_1^*}$ is constant for all classes and represents the largest absolute shift that a class is allowed to have to satisfy Eq. 5 with probability at least $1 - \delta$. Therefore,

$$\mathcal{P} = \left\{ \mathbf{p} \in [0,1]^N \,:\, [\mathbf{p}]_i \in [p_{\mathcal{T},i}^{\min}, p_{\mathcal{T},i}^{\max}]\ \forall\, i \in \{1, \dots, N\} \text{ and } \sum_{i=1}^N [\mathbf{p}]_i = 1 \right\} \tag{8}$$

defines the set of class proportions to which the certified classifier h is safely applicable.

Based on this approach, a variant of the certificate can be derived by modifying $\mathbf{d}$ slightly. For this modification, the negative vector components of $\mathbf{d}$ are set to zero, so that a vector $\mathbf{d}_+$ is formed. This variant is motivated by the observation that, by applying the norm to $\mathbf{d}$, the negative loss components (falsely) contribute as positives to the estimation of the error. Accordingly, $\mathbf{d}_+$ addresses

only the positive error component and allows a more tighter estimate of the inter-domain gap. However, since with $\mathbf{d}_+$ only the positive error components are considered, the range of class proportions can no longer be expressed by $[p^{\min}_{\mathcal{T},i}, p^{\max}_{\mathcal{T},i}]$ and $\mathcal{P}_{\mathbf{d}_+}$ cannot be defined by Eq. 8. As a consequence, $\mathcal{P}_{\mathbf{d}_+}$ is more difficult to characterize than $\mathcal{P}$.

4 Experiments

In the following evaluation, we show that the introduced multi-class certificate indeed represents an upper bound of the inter-domain gap. Besides the correctness of the certificate, the accuracy and tightness of the estimated upper bound are inspected. Ideally, the certificates correspond to upper bounds that are both correct and tight. To this end, we randomly subsample the data to generate different deployment class proportions $\mathbf{p}_\mathcal{T}$ while keeping $\mathbb{P}(X = \mathbf{x} \mid Y = y)$ fixed. To facilitate visualizations in two dimensions, we limit our evaluation to data sets with three classes. The implementation of our configurable experiments is available online[1].

Correctness

The certificate is correct if $\hat{L}_S + \epsilon \geq \hat{L}_\mathcal{T}$ holds, where ϵ is the predicted domain-induced error and $\hat{L}_\mathcal{T}$ is the empirical estimate of the target domain loss [2]. At this point, recognize that computing $\hat{L}_\mathcal{T}$ requires target domain data, which is typically *not* available in ACS. This unavailability raises the desire for an upper bound $\hat{L}_S + \epsilon$ of $\hat{L}_\mathcal{T}$, which allows practitioners to assess, using only ACS-generated data from $\mathcal{S}$, whether their classifier is sufficiently accurate on $\mathcal{T}$. Our certificate is designed to provide this upper bound, and the purpose of our experiments is to validate this claim.

Our experiments cover a repeated three-fold cross validation on six data sets and two learning algorithms, to represent a broad range of scenarios. In total, we have generated 216 000 certificates under the zero-one loss with $\delta = 0.05$. Among these certificates, only one failed, by producing an $\hat{L}_S + \epsilon$ that is larger than $\hat{L}_\mathcal{T}$. Due to the statistical nature of our certificates, $\delta = 0.05$ would have allowed for up to 10 800 failures. Therefore, the number of failures is much smaller than expected. This small number of failures results from the coarse bound estimation that Hölder's inequality provides.

Tightness

A fair comparison between our certificates and our empirical estimate $\hat{L}_\mathcal{T}$ requires us to take the estimation error $\epsilon_\mathcal{T}$ of the baseline, $\hat{L}_\mathcal{T}$, into account [2]. This necessity stems from the fact that $\hat{L}_\mathcal{T}$ is just an estimate from a finite amount of data. Having access to labeled target domain data would thus yield

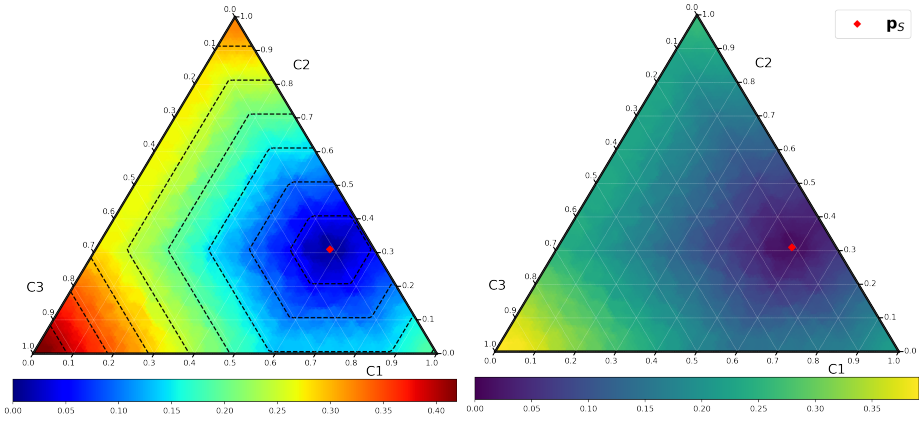
[1] https://github.com/martinsenz/MultiClassAcsCertificates

Table 1: Feasible class proportions Δp^*, according to $\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$ certificates, which are computed for a zero-one loss with $\epsilon = 0.1$ and $\delta = 0.05$.

data set	classifier	$L_S(h)$	$\mathbf{p}_S^\top$	Δp^*
optdigits	DecisionTree	0.100225	$[0.70, 0.20, 0.10]$	0.174888
optdigits	LogisticRegression	0.0955851	$[0.70, 0.20, 0.10]$	0.19131
satimage	DecisionTree	0.10647	$[0.58, 0.31, 0.11]$	0.2285
satimage	LogisticRegression	0.106242	$[0.58, 0.31, 0.11]$	0.217247
pendigits	DecisionTree	0.0467701	$[0.70, 0.20, 0.10]$	0.34621
pendigits	LogisticRegression	0.160971	$[0.70, 0.20, 0.10]$	0.137843
eye movements	DecisionTree	0.488188	$[0.35, 0.26, 0.40]$	0.0652661
eye movements	LogisticRegression	0.515892	$[0.35, 0.26, 0.40]$	0.061098
shuttle	DecisionTree	0.00521672	$[0.15, 0.79, 0.06]$	1.29197
shuttle	LogisticRegression	0.0573444	$[0.15, 0.79, 0.06]$	0.251336
connect4	DecisionTree	0.297242	$[0.65, 0.1, 0.25]$	0.0700096
connect4	LogisticRegression	0.343249	$[0.65, 0.1, 0.25]$	0.0491499

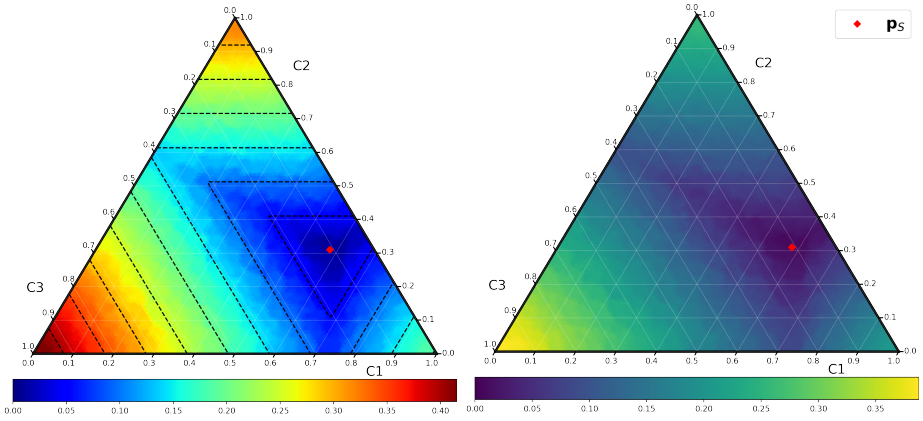
Table 2: MAD and quartiles of the absolute difference between $\hat{L}_S+\epsilon$ and $\hat{L}_\mathcal{T}+\epsilon_\mathcal{T}$.

data set	method	MAD	Q_1	Q_2	Q_3
optdigits	$\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	0.2023 ± 0.0954	0.1258	0.2015	0.2744
optdigits	$\|\mathbf{d}_+\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	0.18 ± 0.1006	0.1009	0.1607	0.2471
satimage	$\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	0.1809 ± 0.087	0.1218	0.1741	0.2324
satimage	$\|\mathbf{d}_+\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	0.1661 ± 0.0908	0.0999	0.1513	0.22
pendigits	$\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	0.1999 ± 0.1445	0.1034	0.168	0.2357
pendigits	$\|\mathbf{d}_+\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	0.1703 ± 0.1452	0.0751	0.1319	0.2034
eye movements	$\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	0.5433 ± 0.245	0.3639	0.5239	0.7331
eye movements	$\|\mathbf{d}_+\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	0.5207 ± 0.2643	0.3058	0.5006	0.7369
shuttle	$\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	0.0879 ± 0.0836	0.0315	0.0531	0.1203
shuttle	$\|\mathbf{d}_+\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	0.071 ± 0.0792	0.0223	0.0424	0.0825
connect4	$\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	0.5094 ± 0.221	0.3419	0.5167	0.6541
connect4	$\|\mathbf{d}_+\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	0.4331 ± 0.2515	0.2274	0.405	0.613



Certified domain-induced error ϵ Tightness $= |\hat{L}_S + \epsilon - L_{\mathcal{T}}|$

Fig. 1: The certified error (left) and its tightness (right), according to $\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$ on the satimage data set, using a logistic regression and the zero-one loss.



Certified domain-induced error ϵ Tightness $= |\hat{L}_S + \epsilon - L_{\mathcal{T}}|$

Fig. 2: The certified error (left) and its tightness (right), according to the variant $\|\mathbf{d}_+\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$ on the satimage data set, using a logistic regression and the zero-one loss.

an upper bound $\hat{L}_{\mathcal{T}} + \epsilon_{\mathcal{T}}$ of the true target domain error $L_{\mathcal{T}}$. We speak of a *tight* bound, if $\hat{L}_S + \epsilon \approx \hat{L}_{\mathcal{T}} + \epsilon_{\mathcal{T}}$.

For example, the prediction of the domain induced error ϵ, as according to our certificate, can be inspected in Fig. 1. The prediction by our $\mathbf{d}_+$ certificate variant is shown in Fig. 2. As we can see, the upper bound is very tight for the area around $\mathbf{p}_S$. With increasing distance from $\mathbf{p}_S$, the estimation of the upper bound becomes larger, and hence, the upper bound becomes coarser. As it is expected, the variant using the $\mathbf{d}_+$ vector provides a finer bound of the inter-domain gap in some regions. Tab. 2 summarizes the absolute deviations between $\hat{L}_S + \epsilon$ and $\hat{L}_{\mathcal{T}} + \epsilon_{\mathcal{T}}$ in terms of mean absolute deviation (MAD) and quartiles (Q_1, Q_2, Q_3).

5 Conclusion and Outlook

Using Hölder's inequality and considering PAC bounds, we have proposed an upper bound $\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$, from which certificates of model robustness in multi-class ACS can be issued. Our experiments demonstrate that this certification is correct within a probability budget δ. Moreover, safe class proportions can easily be described by the maximum allowable absolute deviation Δp^*. Thus, the certification of a multi-class ACS classifier is straightforward for the practitioner to interpret and intuitive to understand, regardless of the number of classes considered in the classification problem.

By decomposing the inter-domain gap into positive and negative error components, it is possible to find estimates that bound the domain gap even more precisely. An example is the presented $\mathbf{d}_+$ certification variant, which considers only the positive error components. In order to obtain even more precise estimates, it is further conceivable to also take the negative error components (correctly) into account. However, as already indicated by the $\mathbf{d}_+$ variant, the complexity of describing the set $\mathcal{P}$ of valid class proportions increases with the expression strength of the upper bound.

In future work, we plan to evaluate more precise estimates of this kind, as well as the other bounds that are provided by Hölder's inequality in Eq. 4. We also plan to use our multi-class certificates as a basis for theoretically justified data acquisition strategies for multi-class ACS, similar to the binary acquisition strategy that is based on binary certificates [1].

References

1. Bunse, M., Morik, K.: Active class selection with uncertain deployment class proportions. In: Workshop on Interactive Adaptive Learning. p. 70 (2021)
2. Bunse, M., Morik, K.: Certification of model robustness in active class selection. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 266–281. Springer (2021)
3. Huang, X., Kroening, D., Ruan, W., Sharp, J., Sun, Y., Thamo, E., Wu, M., Yi, X.: A survey of safety and trustworthiness of deep neural networks: Verification, testing,

adversarial attack and defence, and interpretability. Computer Science Review **37** (2020)
4. Lomasky, R., Brodley, C.E., Aernecke, M., Walt, D., Friedl, M.: Active class selection. In: European Conference on Machine Learning. pp. 640–647. Springer (2007)
5. Moreno-Torres, J.G., Raeder, T., Alaíz-Rodríguez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. Pattern Recognition **45**(1), 521–530 (2012)
6. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering **22**(10), 1345–1359 (2010)
7. Yang, W.H.: On generalized Hölder inequality. Nonlinear Analysis: Theory, Methods & Applications **16**(5), 489–498 (1991)
8. Zhang, K., Schölkopf, B., Muandet, K., Wang, Z.: Domain adaptation under target and conditional shift. In: International Conference on Machine Learning. JMLR Workshop and Conference Proceedings, vol. 28, pp. 819–827 (2013)