



UvA-DARE (Digital Academic Repository)

Learning Objective-Specific Active Learning Strategies with Attentive Neural Processes

Bakker, T.; van Hoof, H.; Welling, M.

DOI

[10.48550/arXiv.2309.05477](https://doi.org/10.48550/arXiv.2309.05477)
[10.1007/978-3-031-43412-9_1](https://doi.org/10.1007/978-3-031-43412-9_1)

Publication date

2023

Document Version

Final published version

Published in

Machine Learning and Knowledge Discovery in Databases : Research Track

License

Article 25fa Dutch Copyright Act (<https://www.openaccess.nl/en/in-the-netherlands/you-share-we-take-care>)

[Link to publication](https://doi.org/10.48550/arXiv.2309.05477)

Citation for published version (APA):

Bakker, T., van Hoof, H., & Welling, M. (2023). Learning Objective-Specific Active Learning Strategies with Attentive Neural Processes. In D. Koutra, C. Plant, M. Gomez Rodriguez, E. Baralis, & F. Bonchi (Eds.), *Machine Learning and Knowledge Discovery in Databases : Research Track : European Conference, ECML PKDD 2023, Turin, Italy, September 18–22, 2023 : proceedings* (Vol. 1, pp. 3-19). (Lecture Notes in Computer Science; Vol. 14169), (Lecture Notes in Artificial Intelligence). Springer. <https://doi.org/10.48550/arXiv.2309.05477>, https://doi.org/10.1007/978-3-031-43412-9_1

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (<https://dare.uva.nl>)



Learning Objective-Specific Active Learning Strategies with Attentive Neural Processes

Tim Bakker¹(✉), Herke van Hoof¹, and Max Welling^{1,2,3,4}

¹ AMLab, University of Amsterdam (UvA), Amsterdam, The Netherlands
{t.b.bakker,h.c.vanhoof,m.welling}@uva.nl

² Canadian Institute for Advanced Research (CIFAR), Toronto, Canada

³ European Lab for Learning and Intelligent Systems (ELLIS), Amsterdam, The Netherlands

⁴ Microsoft Research (MSR), Amsterdam, The Netherlands

Abstract. Pool-based active learning (AL) is a promising technology for increasing data-efficiency of machine learning models. However, surveys show that performance of recent AL methods is very sensitive to the choice of dataset and training setting, making them unsuitable for general application. In order to tackle this problem, the field Learning Active Learning (LAL) suggests to learn the active learning strategy itself, allowing it to adapt to the given setting. In this work, we propose a novel LAL method for classification that exploits symmetry and independence properties of the active learning problem with an Attentive Conditional Neural Process model. Our approach is based on learning from a myopic oracle, which gives our model the ability to adapt to non-standard objectives, such as those that do not equally weight the error on all data points. We experimentally verify that our Neural Process model outperforms a variety of baselines in these settings. Finally, our experiments show that our model exhibits a tendency towards improved stability to changing datasets. However, performance is sensitive to choice of classifier and more work is necessary to reduce the performance the gap with the myopic oracle and to improve scalability. We present our work as a proof-of-concept for LAL on nonstandard objectives and hope our analysis and modelling considerations inspire future LAL work.

Keywords: Active Learning · Deep Learning · Neural Process

1 Introduction

Supervised machine learning models rely on large amounts of representative annotated data and the cost of gathering sufficient data can quickly become prohibitive. Active learning (AL) attempts to mitigate this problem through clever selection of data points to be annotated, thereby reducing total data requirements. To achieve this, AL exploits available information about the dataset and/or supervised task model (e.g. an image classifier) to select data points whose labels are expected to

lead to the greatest increase in task model performance. Most classical AL strategies are hand-designed heuristics, based on researcher intuition or theoretical arguments [49]. Recently, much work has been focused on scaling AL to deep learning (DL) settings, which are even more data-hungry [47]. Such works for instance combine heuristics with representations learned by neural networks [8, 9, 20, 51]), focus specifically on batch acquisition [2, 7, 44, 48, 50], or adapt Bayesian Active Learning by Disagreement (BALD) [17, 26, 28, 32, 33, 40, 52]. Despite these developments, it has been observed that modern AL strategies can vary wildly in performance depending on data setting and that there is no single strategy that consistently performs best [3, 12, 15, 45, 47, 55]. This observation has spurred the development of Learning Active Learning (LAL) methods, which attempt to directly learn an active learning strategy on some data. The goal is to either learn a method that is specifically adapted to the data setting at hand [23, 27, 34], or to learn a strategy that performs well for various data settings [21, 35, 38, 41]. Such methods have the potential of adapting to additional properties of the task as well, such as nonstandard objectives. A prominent real-life example of such objectives appears in imbalanced data settings, where rare classes are typically more important than their standard contribution to the loss or accuracy suggests. Current active learning surveys generally focus on balanced data settings; few large-scale empirical studies exist for alternative objectives, such as imbalanced data and AL methods designed to work with imbalanced data. In this paper, we propose a novel Learning Active Learning (LAL) method for pool-based active learning. The model learns from a myopic oracle, which gives it the ability to adapt to objectives besides standard classification accuracy. We validate our model in imbalanced data settings, where we show that 1) existing AL methods underperform, and 2) the myopic oracle provides a strong signal for learning. Our contributions are as follows:¹

1. We show that a wide range of current pool-based AL methods do not outperform uniformly random acquisition on average across multiple deep learning image classification benchmarks. The tested methods generally perform worse on imbalanced data settings than on balanced data settings, suggesting that current AL methods may be under-optimised for the former.
2. We present experiments with a myopic oracle that show large performance gains over standard AL methods on simple benchmarks. We observe that these gains are larger for imbalanced data settings, suggesting the oracle exploits specific highly-informative samples during acquisition.
3. We propose a novel LAL method based on Attentive Conditional Neural Processes that learn from the myopic oracle. The model naturally exploits symmetries and independence properties of the active learning problem. In contrast to many existing LAL methods, it is not restricted to heuristics and requires no additional data and/or feature engineering.

¹ Experiment code can be found at: <https://github.com/Timsey/npal>.

2 Related Work

The field of active learning has a rich history going back decades, with the current taxonomy of methods founded on the extensive survey by [49]. In this work, we focus on *pool-based* active learning, where a ‘pool’ of unlabelled data points is available, and the goal is to select one or more of these to label (i.e. ‘acquire’ the label). Here we focus on some relevant works, and refer to the supplementary material², for additional discussion.

The aforementioned survey discusses a number of classical pool-based active learning methods, the most notable among which is Uncertainty Sampling. Here label acquisition is determined by the uncertainty of the classifier. How this uncertainty is measured determines the flavour of Uncertainty Sampling: Entropy selects the points that have maximum predictive entropy, Least Confident acquires the sample on which the task model is least confident in its prediction, and Margin selects the data point with the smallest difference in predicted probability for the first and second most likely class. CoreSet [48] instead take a fully geometric approach to active learning by formulating it as a Core-Set selection problem. Acquisition proceeds through optimising annotated data coverage in some representation space. The authors provide a greedy approximation to their algorithm, called k-Center Greedy, which shows competitive performance while being cheaper to compute. Learning Loss [54] adds a loss prediction module to the base task model, motivated by the idea that difficult-to-classify samples are promising acquisition candidates. This module has the goal of predicting the task model’s loss on any given data point and is jointly trained with the task model. Unlabelled samples with the highest predicted loss are then acquired after training.

One potential goal in doing active learning is to select an annotated dataset that represents the true data distribution as well as possible. Based on this idea, Discriminative Active Learning (DAL) [20] learns a classifier (discriminator) to distinguish labelled and unlabelled data based on a representation learned by the task model. Acquisition proceeds by annotating the points that the classifier predicts are most likely to be part of the current unlabelled data pool. Variational Adversarial Active Learning (VAAL) [51] builds on this idea by setting up a two-play mini-max game where a Discriminator network classifies data points as belonging to the labelled or unlabelled set, based on a representation learned by a Variational AutoEncoder (VAE). The VAE is incentivised to fool the discriminator, such that the resulting discriminator probabilities encode similarity between any data point and the currently annotated set. Acquisition then occurs by choosing the least similar points. [8] is a recent Convolutional Graph Neural Network (GCN) method that represents data points as nodes in a graph instead. It too is trained to distinguish labelled and unlabelled datapoints; after training the point with the highest uncertainty according to the GCN is selected for labelling. By representing the full dataset as a graph, this method can encode

² Supplementary material can be found at: https://github.com/Timsey/npal/blob/main/full_paper.pdf.

relevant correlations between data points explicitly. [9] extend this method by using Visual Transformers to learn the graph representation. Although research into active learning methods continues, it has been widely observed that AL strategies performance varies heavily depending on data setting and that there is no single strategy that consistently performs best. Such studies typically focus on balanced data settings [3, 12, 15, 45, 47, 55].

Active Learning for Imbalanced Data: Compared to the wealth of research on active learning, little work has been done on AL for imbalanced datasets specifically. This further motivates imbalanced data settings as relevant nonstandard objectives for active learning. Existing work in this area typically incorporates explicit class-balancing strategies or additional exploration towards difficult examples. Hybrid Active Learning (HAL) [30] is built on the idea that rare samples may differentiate themselves in feature space. HAL trades off geometry-based exploration (e.g. some average distance to the currently annotated data) with informativeness-based exploitation (e.g. as in Uncertainty Sampling). Class-Balanced Active Learning (CBAL) [5] combines entropy sampling with a regulariser that assigns high values to rare points. This regulariser is the difference between a desired class-histogram (i.e. fully balanced classes) and the sum of softmax values of currently sampled points. This intuitively will have the effect of selecting rare points more often. [10] derives an active learning strategy based on selecting the example with the highest estimated probability of misclassification through Bayes’ theorem and various approximate distributions learned by VAE. [1] describe a two-step approach that uses the data’s class imbalance profile to switch from classical AL to a class-balancing acquisition function that favours pool points close (in embedding space) to the rarest class in the annotated data. [4] suggests that doing active learning using the variation ratio of a model ensemble may help counteract imbalance in the data.

Learning Active Learning: With the observation that existing AL methods do not consistently perform well across data settings, interest in learning pool-based active learning has risen. The seminal paper by [27] formulates Active Learning By Learning (ALBL) as a multi-armed bandit problem, where the arms are different AL heuristics. The goal is to learn to select the best heuristic for each acquisition round. [23] learns to fine-tune existing AL heuristics using a Bayesian acquisition net trained with the REINFORCE algorithm. [38] instead learn to imitate actions performed by an approximate oracle. Relatedly, [21] reduce the imitation learning goal to a learning-to-rank problem. They meta-train on synthetic data and show this generalises to other datasets. [34] formulates learning active learning as a regression problem. Similarly to our proposed method, they train a model to predict the reduction in generalisation error expected upon adding a label to the dataset. However, their method requires handcrafted global features representing the classification state and annotated dataset as input to their regressor. In contrast, our method implicitly learns the required features from the raw data, allowing for more complex relationships and simplifying engineering choices. Finally, both [41] and [35] perform meta-learning

over various binary classification datasets. The former employs a meta-network that encodes dataset and classifier states into parameters for a policy, which is reinforcement learned by the REINFORCE algorithm. The latter employs reinforcement learning with a Deep Q-Network and eschews the meta-network. These methods are either still restricted to heuristics [23,27], or require gathering additional representative or synthetic datasets for training [21,35,38,41,46], as well as dataset-independent features.

3 A Study on Existing Active Learning Methods

In pool-based active learning, we are given a labeled (classification) dataset $\mathcal{D}_{annot} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=0}^M$ of size M , where i indexes the data points, $\mathbf{x}_i \in \mathbb{R}^K$ are feature vectors of size K , and $\mathbf{y}_i \in \{0, 1\}^C$ is a (one-hot) label on C total classes. We are further given an unlabelled dataset $\mathcal{D}_{pool} = \{\mathbf{x}_j\}_{j=0}^N$ of size N and are tasked with selecting candidates \mathbf{x}_j from \mathcal{D}_{pool} to annotate: i.e. select the index j , obtain the label \mathbf{y}_j , and subsequently add $(\mathbf{x}_j, \mathbf{y}_j)$ to \mathcal{D}_{annot} . The goal of this procedure is to iteratively improve a task model, e.g. a classifier, trained on the annotated data \mathcal{D}_{annot} . Improvement is typically measured by some performance metric, e.g. the accuracy on some test dataset \mathcal{D}_{test} . Most existing AL methods depend on combinations of heuristics and representation learning for selecting the index j . The implicit expectation is that the selections such heuristics make are also highly performant according to the chosen performance metric. Here we explore whether this assumption holds in modern deep active learning.

Data: To explore the performance of existing heuristic-based AL strategies, we perform active learning on four standard ten-class image classification benchmark datasets: MNIST [11], FashionMNIST [53], SVHN [39], and CIFAR-10 [37]. We use a standard ResNet18 convolutional neural network [24] as the base classifier. We consider three objective settings for each benchmark: Balanced, Imbalanced, and Imbalanced weighted. In imbalanced settings, half the classes are undersampled by a factor 10. Evaluation is performed with a balanced accuracy metric, where instances from undersampled classes are upweighted such that all classes have the same importance. Imbalance weighted additionally takes these weights into account during training. This mimics objectives in typical imbalanced data applications, where rare class instances are often considered more important than common ones [29]. Following [8], we initialise active learning with an annotated dataset \mathcal{D}_{annot} of 1000 data points that follow the specified class ratios; the remaining point also follow these class ratios and are left as the pool dataset \mathcal{D}_{pool} . Every acquisition step we batch annotate 1000 points using the specified AL strategy, for a total of ten steps. After each step, we retrain the classifier from scratch. See the supplementary material for further implementation details.

AL Strategies: First, we consider the three classical uncertainty sampling strategies [49]: ENTROPY, MARGIN and LSTCONF (least-confident). Second, we include

the purely geometric approach of [48]: KCGRDY (K-center greedy). Third, active learning through Learning Loss Module: LLOSS [54]. Fourth, Variational Adversarial Active Learning VAAL [51]; a discriminator method based on VAE-learned representations. Fifth, two variations on the same convolutional graph neural network method – UNCGCN and COREGCN [8] – that employ a jointly learned discriminator and graph embedding; unlike VAAL, this approach can explicitly model inter-datapoint correlations. Sixth, we employ HAL [30] and CBAL [5] as baselines specifically developed for active learning in imbalanced data settings. HAL is further split into HALUNI and HALGAU, depending on the exploration scheme (uniform or Gaussian). Finally, RANDOM is the uniformly random sampling baseline, corresponding to no active learning.

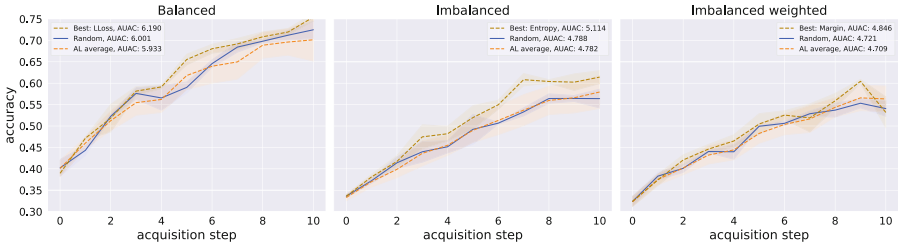


Fig. 1. Random vs. best and average of remaining AL strategies for CIFAR-10 dataset and ResNet18 classifier, 1000 acquisitions per step, and 1000 initial labels. Shaded region represents standard deviation over three seeds.

Results: In Fig. 1 we plot CIFAR-10 test accuracy as a function of acquisition step for RANDOM, the best performing AL method, and the average of all AL methods (excluding RANDOM). AUAC is the Area Under the Acquisition Curve, which is computed as the area under the curves of Fig. 1. It measures performance of the whole AL trajectory. We observe that the average active learning strategy does not perform significantly better than RANDOM in any setting. The best performing AL strategy (by AUAC) does outperform RANDOM. These results suggest that AL can be useful, but only if an appropriate strategy is found for the data at hand; a mismatched strategy can lead to performance worse than uniformly random labelling. Note that there is no consistent best performer among the AL methods. This variation in (relative) performance across benchmarks has been previously observed in the literature [3, 12, 15, 45, 47, 55]. We refer to the supplementary material for implementation details and additional results. We further argue in the supplementary that the tested AL methods generally perform worse on the imbalanced objective settings than on the balanced settings, suggesting that current AL methods may be under-optimised for the former.

4 Myopic Oracle Active Learning

Given the results of the previous section, we may wonder if stronger AL strategies can be found. In particular, it would be valuable to develop strategies that perform well out of the box on many different settings. To this end, the field of Learning Active Learning (LAL) has emerged. The motivating idea is that information about the problem setting should be used for constructing the AL strategy: LAL-methods attempt to do this through learning. What is learned can vary from a choice between existing heuristics [27], to a fine-tuning of such heuristics [23], to a labelling policy that tries to generalise over datasets [21, 35, 38, 41], to the direct improvement to the underlying classifier upon annotating a datapoint in the given dataset [34]. Ideally, the learned AL strategy should not be constrained to be close to human heuristics, as there is no guarantee that optimal strategies can be represented as such. Additionally, we will only require the availability of a single dataset to train an AL strategy, since finding additional datasets representative of the problem setting at hand is often not feasible in real-world applications. That leaves us with strategies similar to those in e.g. [34], where the AL strategy tries to learn a function mapping the features of an unlabelled datapoint to the expected improvement of the classifier after retraining with that datapoint labelled. Before attempting to train such a strategy, we should quantify whether such a method – if properly learned – actually improves much over existing methods. To this end, we introduce the myopic oracle strategy – denoted ORACLE in the below – which computes the actual classifier improvement on the test data for an unlabelled datapoint \mathbf{x}_j in \mathcal{D}_{pool} , by treating the corresponding label \mathbf{y}_j as known and retraining the classifier with this additional label. This improvement is stored, the classifier is reset, and the process is repeated for every datapoint in \mathcal{D}_{pool} . Pseudocode for obtaining improvement scores with the ORACLE is presented in the supplementary. ORACLE then selects the datapoint $(\mathbf{x}^*, \mathbf{y}^*)$ corresponding to the largest classifier improvement and this point is added to the annotated dataset \mathcal{D}_{annot} . This oracle uses information that is typically unavailable during the AL process, namely the true labels \mathbf{y}_j and the exact classifier improvements on the test set. The oracle is myopic, as it greedily acquires the best datapoint every acquisition step, rather than planning ahead: looking ahead t acquisition steps requires retraining the classifier $\binom{|\mathcal{D}_{pool}|}{t}$ times, which is infeasible.

Classifiers: Even for $t = 1$, the myopic oracle strategy requires retraining the underlying classifier $|\mathcal{D}_{pool}|$ times every acquisition step, which is computationally intractable for neural network classifiers. For this reason, our experiments in this setting use simpler classification models. We run experiments with logistic regression classifiers and provide additional experiments with support vector machine (SVM) classifiers in the supplementary material. These are both quick-to-train models that have a long history of being used in AL research [16, 36, 49, 55], including within the subfield of LAL [27, 34, 35, 46]. For both classifiers, we employ the default scikit-learn implementations [43], with class-weighting when specified.

Table 1. AL strategy AUAC and final-step test accuracy on UCI waveform dataset with logistic regression classifier, 1 acquisition per step, and 100 initial labels. Averages and standard deviations are computed over nine seeds.

Strategy	Balanced		Imbalanced		Imbalanced weighted	
	AUAC	Test acc.	AUAC	Test acc.	AUAC	Test acc.
ORACLE	9.14 \pm 0.12	0.93 \pm 0.01	8.84 \pm 0.39	0.89 \pm 0.04	9.22 \pm 0.18	0.93 \pm 0.02
UNCSAMP	8.67 \pm 0.17	0.87 \pm 0.01	8.40 \pm 0.49	0.85 \pm 0.04	8.55 \pm 0.33	0.86 \pm 0.02
KCGRDY	8.68 \pm 0.28	0.87 \pm 0.03	8.29 \pm 0.49	0.84 \pm 0.04	8.58 \pm 0.37	0.86 \pm 0.03
HALUNI	8.66 \pm 0.26	0.87 \pm 0.03	8.11 \pm 0.55	0.81 \pm 0.06	8.45 \pm 0.46	0.85 \pm 0.05
HALGAU	8.68 \pm 0.23	0.87 \pm 0.02	8.16 \pm 0.54	0.82 \pm 0.05	8.48 \pm 0.45	0.85 \pm 0.04
CBAL	8.67 \pm 0.15	0.87 \pm 0.02	8.30 \pm 0.45	0.84 \pm 0.04	8.65 \pm 0.34	0.87 \pm 0.03
RANDOM	8.65 \pm 0.23	0.87 \pm 0.02	8.17 \pm 0.54	0.82 \pm 0.05	8.42 \pm 0.48	0.85 \pm 0.05
NP	8.69 \pm 0.19	0.87 \pm 0.02	8.25 \pm 0.53	0.83 \pm 0.05	8.61 \pm 0.33	0.87 \pm 0.03

Data: These simpler classifiers do not perform well on the image datasets of Sect. 3. In order to properly study the effects acquisition has on model performance, we instead use simpler datasets. A popular choice in the field of learning active learning [35, 38, 46] are binary classification datasets from the UCI data repository [13]. We use the ‘waveform’, ‘mushrooms’ and ‘adult’ datasets, since these contain sufficient samples for our experiments post-imbaling. Data is imbalanced by a factor of ten, as in the previous experiments. In all experiments we initialise the runs with 100 annotated examples and acquire one additional label in each of ten acquisition steps. We set aside 200 datapoints as test data \mathcal{D}_{test} for evaluating the classifiers; oracle scores are also computed on this test data.

AL Strategies: We first compare ORACLE with a logistic regression classifier to the same set of AL strategies we compared to in Sect. 3. However, we skip the comparisons to LLOSS, VAAL, UNCGCN, and COREGCN, since these all require neural network classifiers as their base. Additionally, the three uncertainty sampling methods ENTROPY, MARGIN, and LSTCONF reduce to the same algorithm for binary classification: we henceforth denote this method as UNCSAMP. Our goal is to work towards a general-purpose AL method that can be trained using only available data. Therefore, we do not include the discussed LAL methods in our baselines, as these methods either adapt existing heuristics or require heavy feature engineering and/or additional datasets to train.

Results: Table 1 compares the performance of the ORACLE to pre-existing AL methods on the waveform dataset for the logistic regression classifier. The NP method will be introduced and discussed in the next section. It is clear that ORACLE dominates all other AL strategies in all settings. Note that AL is only responsible for a small fraction of the total datapoints in the final step here (10 of 110), whereas in the experiments of the previous section, it was responsible for the majority of datapoints (10000 of 11000). As may be observed in

the table, such a small number of points is enough to obtain meaningful differences in scores between AL strategies. This indicates that this benchmark contains sufficient variability between strategies to observe meaningful differences in AL quality, making it an appropriate environment for learning active learning. These results suggest that the function represented by ORACLE is a strong active learner that adapts to the given objective. Moreover, we note that the performance gap between ORACLE and RANDOM – and more generally between the various AL strategies – is larger in the imbalanced settings, providing evidence that acquisition choice is more important in these settings; something ORACLE can directly exploit. We refer to the supplementary material for implementation details and additional results. In the next section, we turn our attention to an attempt at learning an approximation to the ORACLE using a Neural Process model.

5 Learning Active Learning with a Neural Process

Our approach will be to learn an approximation to ORACLE, by training a model to predict classifier improvement values for every point in \mathcal{D}_{pool} , given a context of annotated datapoints and classifier state. However, we cannot train on the true myopic oracle values, as this requires pool data labels and test data that we do not have access to at training time. Instead, we opt to simulate active learning scenarios by subsampling \mathcal{D}_{annot} . For these simulated settings we can compute the improvement values that provide the training signal. Our approach will perform the following procedure at every step of the acquisition process:

1. Simulate many active learning scenarios by subsampling \mathcal{D}_{annot} into N_{sim} pairs of annotated and pool data $(\mathcal{S}_{annot}^{(i)}, \mathcal{S}_{pool}^{(i)})$, with $i \in [1, N_{sim}]$.
2. Use the myopic oracle to compute – for each point in all the $\mathcal{S}_{pool}^{(i)}$ – the classifier improvement observed after retraining with that point and its label to the current dataset $\mathcal{S}_{annot}^{(i)}$.
3. Train a model to predict these improvements from the input $(\mathcal{S}_{annot}^{(i)}, \mathcal{S}_{pool}^{(i)})$.

The challenge is now to design a model and training setup that can generalise strategies learned in the simulated settings to the full test-time AL setting represented by \mathcal{D}_{annot} and \mathcal{D}_{pool} . Here we describe our considerations and resulting approach to this challenge. First, the classifier improvements used for training should not be computed using test data, as this data is not available during training. Instead, we compute these scores on a held-out ‘reward’ dataset \mathcal{D}_{val} . In practice, this reward set was used instead of a validation set, so the usual train-val-test split suffices for training our active learner. Second, our problem setup contains permutation symmetries that can be exploited: the (simulated) annotated dataset forms the context that informs the predictions (improvement scores) of our model, but the order of these points does not matter for the prediction: the context representation should be permutation invariant. Additionally, if our model predicts scores for every (simulated) pool datapoint, then these scores

should be permutation equivariant: exchanging the index of two pool points should simply exchange the scores. Third, in the myopic setting, the score of any pool point is independent of any other pool point, so all point points should be treated individually (i.e., not exchange information). This imposes that the model should be invariant to the number of points in \mathcal{D}_{pool} . Note that the independence condition is broken in the non-myopic setting, as combinations of pool points can lead to stronger improvements than the individual myopic scores would suggest. The combination of the second and third conditions/inductive biases heavily restrict the choice of model. A natural choice is to use Neural Process (NP) models [14, 18, 19, 31] to learn the approximate ORACLE.

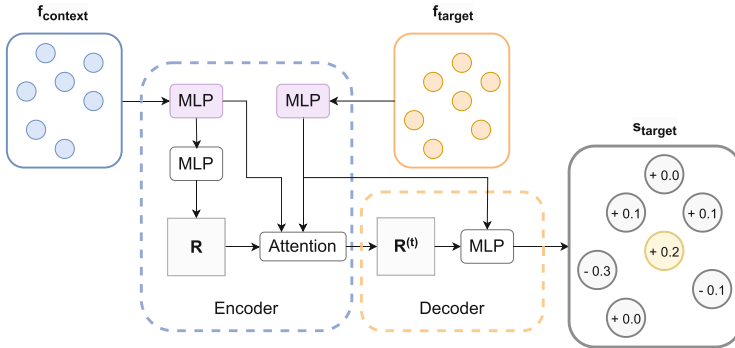


Fig. 2. Computational graph for the Attentive Conditional Neural Process model. The model takes sets of datapoints as input and predicts improvement values for the target points. Context points correspond to annotated data and target points to pool data. All MLPs are applied pointwise. The top two MLPs (in purple) share weights. (Color figure online)

The Neural Process: The Neural Process comprises a class of models for meta-learning context-conditional predictors and is a natural choice for our approximator. Given a context \mathcal{C} and target input features \mathbf{f}_τ , the Neural Process outputs a distribution $p(\mathbf{s}_\tau | \mathbf{f}_\tau; \mathcal{C})$ over target predictions \mathbf{s}_τ . To apply this model to our problem, we identify the context \mathcal{C} with the information stored in the annotated data and the classifier state, the target input features \mathbf{f}_τ with the features of pool datapoints, and the target predictions \mathbf{s}_τ with the predicted classifier improvements associated to those pool points. We can then train the NP by performing supervised learning – maximising the log likelihood of target improvements \mathbf{y}_τ – on simulated AL scenarios. At test time we apply the trained model with the full \mathcal{D}_{annot} as context and \mathcal{D}_{pool} as target input. In particular, we utilise an Attentive Conditional NP (AttnCNP) [18, 31], with cross-attention between the pool and annotated points. The CNP factorises the predictive distribution conditioned on the context set, as

$$p(\mathbf{s}_\tau | \mathbf{f}_\tau; \mathcal{C}) = \prod_{t=1}^T p(s^{(t)} | f_\tau^{(t)}; \mathcal{C}), \quad (1)$$

where T is the number of target datapoints. This modelling choice satisfies the independence of pool point predictions. The context \mathcal{C} should be permutation invariant and is typically encoded into a global representation R . The NP is parameterised by a neural network with parameters $\{\theta, \phi\}$ and each factor is typically set to be a Gaussian density [14], as:

$$p_{\theta, \phi}(\mathbf{s}_\tau | \mathbf{f}_\tau; \mathcal{C}) = p_{\theta, \phi}(\mathbf{s}_\tau | \mathbf{f}_\tau; R) = \prod_{t=1}^T p_{\theta, \phi}(s^{(t)} | \mathbf{f}_\tau^{(t)}; R) = \prod_{t=1}^T \mathcal{N}(s^{(t)}; \mu^{(t)}, \sigma^{2(t)}), \quad (2)$$

where $R = \text{Enc}_\theta(\mathcal{C})$ encodes the context and $(\mu^{(t)}, \sigma^{2(t)}) = \text{Dec}_\phi(R, \mathbf{f}_\tau^{(t)})$ decodes the context encoding and the target features into target predictive parameters. The AttnCNP extends this model by replacing the global representation R with a target-specific representation $R^{(t)}$ through the use of an attention mechanism. In particular, we use the attention mechanism taken from the Image Transformer [42] to perform cross-attention between context and target features, constructing $R^{(t)}$. Here context features $\mathbf{f}_\mathcal{C}$ are treated as keys and target features \mathbf{f}_τ as queries. Values are constructed from $\mathbf{f}_\mathcal{C}$ by applying a pointwise MLP with 2 hidden layers of size 32 and ReLU activations. Our implementation does not use self-attention on the context or target features, as applying self-attention to the target features violates the independence of the pool point scores. In preliminary experimentation, we found that omitting the attention mechanism – e.g. $R^{(t)} = R$ – resulted in performance drops due to underfitting the target function, as has been observed in the Neural Process literature [31]. A computational graph of our model is presented in Fig. 2. This model satisfies the required permutation symmetries while allowing scores of pool points to be given by expressive functions that depend on the context and pool point. In this proof-of-concept study we do not explore the use of uncertainty information for acquisition, rather opting to acquire the datapoint for which $\mu^{(t)}$ – the predicted mean score – is maximal, as $j = \arg \max_{t \in [1, T]} \mu^{(t)}$. We then acquire the pool datapoint with index j , completing a single step in the Active Learning process. The Neural Process is then initialised from scratch, in preparation for the next acquisition step.

Data: The experiments for our Neural Process model (NP) are performed on the datasets described in the previous section. In order to train the NP model, we simulate active learning scenarios by sampling from the existing annotated dataset \mathcal{D}_{annot} . We define a set of fractions Q and uniformly sample from these a total of N_{sim} times, leading to a set of annotation fractions $\{q_i\}_{i=1}^{N_{sim}}$. For each value of i , we then assign the corresponding fraction q_i of datapoints from \mathcal{D}_{annot} to a *simulated* annotated dataset $\mathcal{S}_{annot}^{(i)}$; the remaining points are assigned to a *simulated* pool dataset $\mathcal{S}_{pool}^{(i)}$. This procedure results in a set of N_{sim} simulated/sampled active learning problems of various sizes. We then compute oracle scores of all pool points in each of the resulting AL problems ($\mathcal{S}_{annot}^{(i)}, \mathcal{S}_{pool}^{(i)}$). Since we do not have access to test data at train time, the oracle scores are instead computed on the held-out \mathcal{D}_{val} . We present pseudocode in Algorithm 1.

Algorithm 1: Training the NP model.

Data: Annotated dataset \mathcal{D}_{annot} , Neural Process model NP_θ , number of simulations N_{sim} , set of fractions Q , oracle ORACLE, base classifier model C , scoring function SCORE evaluated on \mathcal{D}_{val} .

Result: Trained parameters θ^* for NP.

```

for  $i = 1, 2, \dots, N_{sim}$  do
   $q_i \leftarrow \text{sample}(Q)$ ; /* Uniformly sample an ‘annotation fraction’ */
   $S_{annot}^{(i)} \leftarrow \emptyset$ ; /* Initialise a simulated annotated set */
   $S_{pool}^{(i)} \leftarrow \mathcal{D}_{annot}$ ; /* Initialise a simulated pool set */
  while  $|S_{annot}^{(i)}| < \text{round}(q_i \cdot |\mathcal{D}_{annot}|)$  do
    Sample index  $j$  of datapoints in  $\mathcal{D}_{annot}$  uniformly without replacement;
     $S_{annot}^{(i)} \leftarrow S_{annot}^{(i)} \cup (\mathbf{x}_j, \mathbf{y}_j)$ ;
     $S_{pool}^{(i)} \leftarrow S_{pool}^{(i)} \setminus (\mathbf{x}_j, \mathbf{y}_j)$ ;
  end
   $V_i \leftarrow \text{ORACLE}(S_{annot}^{(i)}, S_{pool}^{(i)}, C, \text{SCORE})$ ; /* Obtain improvement scores
  with Oracle (pseudocode in supplementary) */
end
 $\theta^* \leftarrow \text{NP}_\theta.\text{FIT}(\{S_{annot}^{(i)}, S_{pool}^{(i)}, V_i\}_{i=1}^{N_{sim}})$ ; /* Train the NP on the simulated
AL settings */
return  $\theta^*$ 

```

Experimentally we find that simulating with a variety of fractions in Q improves generalisation to the target problem over using a fixed single fraction. Our experiments use $Q = \{0.1, 0.2, \dots, 0.8, 0.9\}$ and $N_{sim} = 300$. Preliminary experimentation showed no performance increase for larger values of N_{sim} , while using $N_{sim} = 100$ led to slight performance decreases. The held-out dataset \mathcal{D}_{val} consists of the same 100 datapoints for all i .

Results: Table 1 shows the performance of our method – NP – on the UCI waveform dataset with logistic regression classifier. Ignoring ORACLE, the Neural Process ranks best of all active learning methods in AUAC on the Balanced setting, second on Imbalanced weighted, and fourth on Imbalanced. In Fig. 3 we show the performance difference between our method and a chosen baseline. Here we choose the average of AL strategies – AL AVERAGE – as the baseline, where we exclude ORACLE, RANDOM, and NP from the average. This choice of baseline allows us to clearly see whether any particular method is expected to improve over a naive application of active learning. We also show the performance of the best AL strategy – BEST – again excluding ORACLE, RANDOM, and NP from the selection. This represents the relative performance of choosing the best AL strategy post-hoc. We observe that NP performs on par with BEST for Balanced and Imbalanced weighted, and performs similarly to AL AVERAGE for the Imbalanced setting. In all cases, the gap with ORACLE remains large, indicating potential room for improvement. Shaded regions correspond to twice

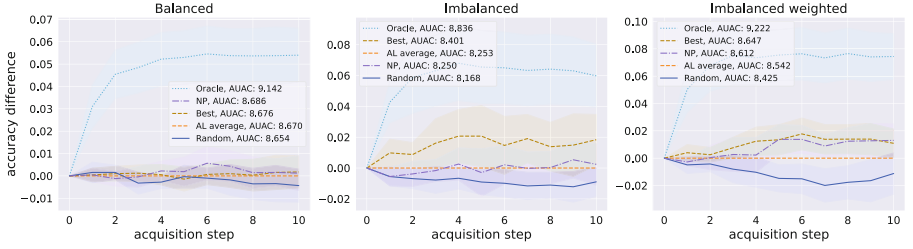


Fig. 3. Relative performance of acquisition strategies for waveform dataset and logistic regression classifier, 1 acquisition per step, and 100 initial labels. Accuracy differences of RANDOM, ORACLE, NP and the best remaining AL strategy (BEST) are computed w.r.t. the average of remaining AL strategies (AL AVERAGE). The shaded region represents twice the standard error of the mean over nine seeds.

the standard error of the mean, i.e., $2 \cdot \frac{\sigma}{\sqrt{n}}$, where σ is the standard deviation and n the number of runs. Performance tables and figures for the mushrooms and adult dataset are provided in the supplementary material. NP at least slightly outperforms AL AVERAGE on Imbalanced and Imbalanced weighted for these datasets and in half those cases achieves near-BEST performance. However, NP ranks near the bottom in the Balanced setting here. Interestingly, RANDOM outperforms almost all methods on Balanced, possibly indicating increased difficulty in active learning, although ORACLE does still demonstrate a large performance gap.

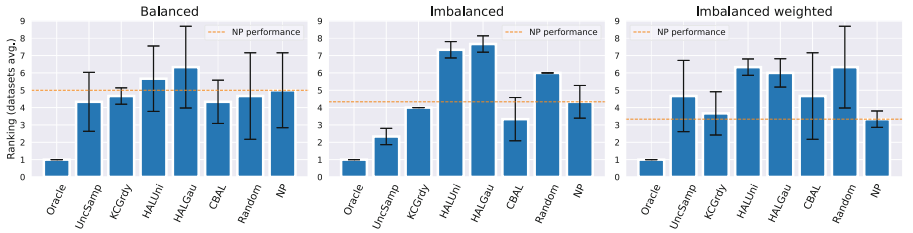


Fig. 4. Relative AUAC rank of AL strategies averaged over the three UCI datasets for logistic regression. The standard deviation of this rank is denoted by the error bars.

Our method is partially motivated by the need for AL algorithms that perform more stably across different data settings. To this end, Fig. 4 shows the average AUAC ranking of every AL method across the three UCI datasets. We observe that NP is the best performing AL method on average for the Imbalanced weighted setting and has more middling performance for the other two settings, with Balanced being the worst for our model. Inspecting the ranking standard deviation, we further see that our model achieves a relatively stable ranking across the three datasets in the Imbalanced weighted setting. This stability again degrades for Imbalanced and even further for Balanced. However,

note that low standard deviation is only desirable for models with low performance rank, as it otherwise indicates a stable underperformance. These results suggest that NP is better able to exploit information encoded by the ORACLE in imbalanced settings. We present results for the SVM classifier in the supplementary. This setting seems more difficult for NP to learn, suggesting that the choice of the underlying classifier is important.

6 Conclusion and Discussion

It has been observed in the literature that a wide range of current pool-based active learning methods do not perform better than uniform acquisition on average across standard deep learning benchmarks. We have experimentally verified these results and extended them to imbalanced data settings, which are relevant alternative objectives for many real-world applications. We have explored the validity of using a myopic oracle as a target function for learning active learning (LAL) and have shown its dominating performance on simple active learning tasks. Finally, we have identified symmetry and independence properties of such active learning problems and have modelled these using an Attentive Conditional Neural Process. Unlike existing LAL methods, our model (NP) is not based on existing heuristics, and requires no feature engineering and/or additional datasets to train. Our model generally outperforms the average of the competing AL methods in imbalanced data settings, and occasionally all of them individually. However, future work is needed to evaluate performance on additional datasets, reduce the performance gap with the myopic oracle, and improve scalability. We present our work as a proof-of-concept for LAL on non-standard objectives – with a focus on imbalanced data settings – and hope our analysis and modelling considerations inspire future LAL work.

Limitations: The primary limitation of our Neural Process approach is scalability. Supervised learning on the myopic oracle requires retraining the base classifier a large number of times during NP training, which is infeasible for large neural network models. Future work may explore to which degree functions learned on simple classifiers can be transferred to more powerful models. Next, the acquisition procedure may be improved through the use of uncertainty information present in the NP. Finally, the NP input may be augmented with additional features – e.g., predicted class probabilities of pool points – to potentially improve learning. Preliminary experimentation showed little effect on performance: we leave further exploring the use of extra features for future work.

Ethical Considerations. In recent years, machine learning has had a large impact on society by enabling the development of a variety of new, widely-deployed technologies. Opinions on the value of such technologies vary, but it is clear that they have had both positive and negative impacts. Our research topic of active learning is a promising technology for increasing the efficiency of

machine learning model training. Developments in active learning may reduce barrier-to-entry for training and deploying high-performing predictive models, which potentially has both positive and negative downstream consequences. On the positive side, wider access to strong models may increase the adoption of life-saving or simply quality-of-life-improving technologies. Additionally, it may allow relatively less powerful interest groups to not fall behind larger or more powerful institutions in capabilities, thus supporting democratisation of AI. On the negative side, improved active learning has the potential to exacerbate the negative effects of machine learning applications as well. Such exacerbation may happen through widening the aforementioned capability gap between less and more powerful institutions (e.g., by potentially easing model scaling), or through reinforcing existing model biases during training. Additionally, training large-scale models consumes a large amount of energy, potentially worsening the current energy and climate crises. Finally, any machine learning capabilities research potentially exacerbates the future risks of AI misalignment; risks that are worrying to an increasing share of the research community [22, 25].

Acknowledgments. This work is supported by the ‘Efficient Deep Learning’ (EDL, <https://efficientdeeplearning.nl>) research programme, which is financed by the Dutch Research Council (NWO) domain Applied and Engineering Sciences (TTW). We are have used Weights & Biases [6] for experiment tracking.

References

1. Aggarwal, U., Popescu, A., Hudelot, C.: Active learning for imbalanced datasets. In: WACV (2020)
2. Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds. In: ICLR (2020)
3. Baram, Y., El-Yaniv, R., Luz, K.: Online choice of active learning algorithms. *JMLR* **5**, 255–291 (2004)
4. Beluch, W.H., Genewein, T., Nurnberger, A., Kohler, J.M.: The power of ensembles for active learning in image classification. In: CVPR (2018)
5. Bengar, J., van de Weijer, J., Fuentes, L., Raducanu, B.: Class-balanced active learning for image classification. In: WACV (2022)
6. Biewald, L.: Experiment tracking with Weights and Biases (2020). <https://www.wandb.com/>
7. Bıyık, E., Wang, K., Anari, N., Sadigh, D.: Batch active learning using determinantal point processes. arXiv e-prints (2019)
8. Caramalau, R., Bhattarai, B., Kim, T.K.: Sequential graph convolutional network for active learning. In: CVPR (2021)
9. Caramalau, R., Bhattarai, B., Kim, T.K.: Visual transformer for task-aware active learning. arXiv e-prints (2021)
10. Choi, J., et al.: VaB-AL: incorporating class imbalance and difficulty with variational Bayes for active learning. In: CVPR (2020)
11. Deng, L.: The MNIST database of handwritten digit images for machine learning research. *IEEE-SPM* **26**, 141–142 (2012)
12. Desreumaux, L., Lemaire, V.: Learning active learning at the crossroads? Evaluation and discussion. arXiv e-prints (2020)

13. Dua, D., Graff, C.: UCI machine learning repository (2017). <http://archive.ics.uci.edu/ml>
14. Dubois, Y., Gordon, J., Foong, A.Y.: Neural process family (2020). <http://yanndubs.github.io/Neural-Process-Family/>
15. Ebert, S., Fritz, M., Schiele, B.: RALF: a reinforced active learning formulation for object class recognition. In: CVPR (2012)
16. Ertekin, S., Huang, J., Bottou, L., Giles, L.: Learning on the border: active learning in imbalanced data classification. In: CIKM (2007)
17. Gal, Y., Islam, R., Ghahramani, Z.: Deep Bayesian active learning with image data. In: ICML (2017)
18. Garnelo, M., et al.: Conditional neural processes. In: ICML (2018)
19. Garnelo, M., et al.: Neural Processes. arXiv e-prints (2018)
20. Gissin, D., Shalev-Shwartz, S.: Discriminative active learning. arXiv e-prints (2019)
21. Gonsior, J., Thiele, M., Lehner, W.: ImitAL: learning active learning strategies from synthetic data. arXiv e-prints (2021)
22. Grace, K.: What do ML researchers think about AI in 2022? <https://aiimpacts.org/what-do-ml-researchers-think-about-ai-in-2022/>
23. Haussmann, M., Hamprecht, F., Kandemir, M.: Deep active learning with adaptive acquisition. In: IJCAI (2019)
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
25. Hendrycs, D.: Statement on AI Risk. <http://www.safe.ai/statement-on-ai-risk>
26. Houlshby, N., Huszár, F., Ghahramani, Z., Lengyel, M.: Bayesian active learning for classification and preference learning. arXiv e-prints (2011)
27. Hsu, W.N., Lin, H.T.: Active learning by learning. In: AAAI (2015)
28. Jesson, A., Tigas, P., van Amersfoort, J., Kirsch, A., Shalit, U., Gal, Y.: Causal-BALD: deep Bayesian active learning of outcomes to infer treatment-effects from observational data. In: NeurIPS (2021)
29. Johnson, J.M., Khoshgoftaar, T.M.: Survey on deep learning with class imbalance. *J. Big Data* **6**, 27 (2019). <https://doi.org/10.1186/s40537-019-0192-5>
30. Kazerouni, A., Zhao, Q., Xie, J., Tata, S., Najork, M.: Active learning for skewed data sets. arXiv e-prints (2020)
31. Kim, H., et al.: Attentive neural processes. In: ICLR (2019)
32. Kirsch, A., Gal, Y.: PowerEvaluationBALD: efficient evaluation-oriented deep (Bayesian) active learning with stochastic acquisition functions. arXiv e-prints (2021)
33. Kirsch, A., Rainforth, T., Gal, Y.: Test distribution-aware active learning: a principled approach against distribution shift and outliers. arXiv e-prints (2021)
34. Konyushkova, K., Sznitman, R., Fua, P.: Learning active learning from data. In: NeurIPS (2017)
35. Konyushkova, K., Sznitman, R., Fua, P.: Discovering general-purpose active learning strategies. arXiv e-prints (2018)
36. Kremer, J., Steenstrup Pedersen, K., Igel, C.: Active learning with support vector machines. *DMKD* **4**, 313–326 (2014)
37. Krizhevsky, A.: Learning multiple layers of features from tiny images. University of Toronto, Technical report (2009)
38. Liu, M., Buntine, W., Haffari, G.: Learning how to actively learn: a deep imitation learning approach. In: ACL (2018)
39. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NeurIPS Workshop (2011)

40. Nikoloska, I., Simeone, O.: BAMLd: Bayesian active meta-learning by disagreement. arXiv e-prints (2021)
41. Pang, K., Dong, M., Wu, Y., Hospedales, T.: Meta-learning transferable active learning policies by deep reinforcement learning. arXiv e-prints (2018)
42. Parmar, N., et al.: Image transformer. In: ICML (2018)
43. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. In: JMLR (2011)
44. Pinsler, R., Gordon, J., Nalisnick, E.T., Hernández-Lobato, J.M.: Bayesian batch active learning as sparse subset approximation. In: NeurIPS (2019)
45. Ramirez-Loaiza, M.E., Sharma, M., Kumar, G., Bilgic, M.: Active learning: an empirical study of common baselines. DMKD **31**, 287–313 (2017). <https://doi.org/10.1007/s10618-016-0469-7>
46. Ravi, S., Larochelle, H.: Meta-learning for batch mode active learning. In: ICLR Workshop (2018)
47. Ren, P., et al.: A survey of deep active learning. arXiv e-prints (2020)
48. Sener, O., Savarese, S.: Active learning for convolutional neural networks: a core-set approach. In: ICLR (2018)
49. Settles, B.: Active learning literature survey. University of Wisconsin-Madison, Technical report (2009)
50. Shui, C., Zhou, F., Gagn'e, C., Wang, B.: Deep active learning: unified and principled method for query and training. In: AISTATS (2020)
51. Sinha, S., Ebrahimi, S., Darrell, T.: Variational adversarial active learning. In: ICCV (2019)
52. Woo, J.O.: BABA: beta approximation for Bayesian active learning. arXiv e-prints (2021)
53. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv e-prints (2017)
54. Yoo, D., Kweon, I.S.: Learning loss for active learning. In: CVPR (2019)
55. Zhan, X., Chan, A.B.: ALdataset: a benchmark for pool-based active learning. arXiv e-prints (2020)