

Article

Deep Individual Active Learning: Safeguarding against Out-of-Distribution Challenges in Neural Networks

Shachar Shayovitz ^{*,†} , Koby Bibas [†] and Meir Feder

School of Electrical Engineering, Tel Aviv University, Tel Aviv 6997801, Israel; kobybibas@gmail.com (K.B.); meir@tauex.tau.ac.il (M.F.)

* Correspondence: shachar.shay@gmail.com

[†] These authors contributed equally to this work.

Abstract: Active learning (AL) is a paradigm focused on purposefully selecting training data to enhance a model's performance by minimizing the need for annotated samples. Typically, strategies assume that the training pool shares the same distribution as the test set, which is not always valid in privacy-sensitive applications where annotating user data is challenging. In this study, we operate within an individual setting and leverage an active learning criterion which selects data points for labeling based on minimizing the min-max regret on a small unlabeled test set sample. Our key contribution lies in the development of an efficient algorithm, addressing the challenging computational complexity associated with approximating this criterion for neural networks. Notably, our results show that, especially in the presence of out-of-distribution data, the proposed algorithm substantially reduces the required training set size by up to 15.4%, 11%, and 35.1% for CIFAR10, EMNIST, and MNIST datasets, respectively.

Keywords: active learning; universal prediction; deep active learning; individual sequences; normalized maximum likelihood; out-of-distribution



Citation: Shayovitz, S.; Bibas, K.; Feder, M. Deep Individual Active Learning: Safeguarding against Out-of-Distribution Challenges in Neural Networks. *Entropy* **2024**, *26*, 129. <https://doi.org/10.3390/e26020129>

Academic Editors: Badong Chen, Luis Gonzalo Sánchez Giraldo, Shuangming Yang and Shujian Yu

Received: 27 December 2023

Revised: 28 January 2024

Accepted: 30 January 2024

Published: 31 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In supervised learning, a training set is provided to a learner, which can then be used to choose parameters for a model that minimize the error on this set. The process of creating this training set requires annotation, where an expert labels the data points. This is a time-consuming and costly process and results in only a small subset of the data being labeled, which may not represent the true underlying model [1]. Active learning, where the training data are actively and purposely chosen, allows the learner to interact with a labeling expert by sequentially selecting samples for the expert to label based on previously observed data, thereby reducing the number of examples needed to achieve a given accuracy level [2].

Recent research has focused on obtaining a diverse set of samples for training deep learning models with reduced sampling bias. The strategies in [3–6] aim to quantify the uncertainties of samples from the unlabeled pool and utilize them to select a sample for annotation. A widely used criterion for active learning is Bayesian Active Learning by Disagreement (BALD), which was originally proposed by Houlisby et al. [3]. This method finds the unlabeled sample \hat{x}_i that maximizes the mutual information between the model parameters θ and the candidate label random variable Y_i given the candidate x_i and training set $z^{n-1} = \{(x_i, y_i)\}_{i=1}^{n-1}$:

$$\hat{x}_i = \underset{x_i}{\operatorname{argmax}} I(\theta; Y_i | x_i, z^{n-1}) \quad (1)$$

where $I(X; Y | z)$ denotes the mutual information between the random variables X and Y conditioned on a realization z . The idea in BALD's core is to minimize the uncertainty

about model parameters using Shannon’s entropy. This criterion also appears as an upper bound on information-based complexity of stochastic optimization [7] and also for experimental design [8,9]. There is an issue of postulating a reasonable prior for this Bayesian approach. Empirically, this approach was investigated by Gal et al. [4], where a heuristic Bayesian method for deep learning was proposed, leading to several heuristic active learning acquisition functions that were explored within this framework.

However, BALD has a fundamental disadvantage if the test distribution differs from the training set distribution, since what is maximally informative for model estimation may not be maximally informative for test time prediction. In a previous work, Shayovitz and Feder [6] derived a criterion named Universal Active Learning (UAL) that takes into account the unlabeled test set when optimizing the training set:

$$\hat{x}_i = \underset{x_i}{\operatorname{argmin}} I(\theta; Y|X, x_i, Y_i, z^{n-1}) \quad (2)$$

where X and Y are the test feature and label random variables. UAL is derived from a capacity–redundancy theorem [10] and implicitly optimizes an exploration–exploitation trade-off in feature selection. In addition, in the derivation of [10], the prior on θ is expressed as the capacity-maximizing distribution for $I(\theta; Y|X, x_i, Y_i, z^{n-1})$. It should be noted that Smith et al. [11] have recently proposed a criterion denoted Expected Predictive Information Gain (EPIG) which also takes into account the unlabelled test set and focuses on prediction and not model estimation (In Appendix A, it is proven that EPIG is equivalent to UAL, but unlike EPIG, which does not optimize the model prior, UAL provides an expression for the optimal model prior.):

$$\hat{x}_i = \underset{x_i}{\operatorname{argmax}} I(Y; Y_i|X, x_i, z^{n-1}) \quad (3)$$

However, the above-mentioned AL schemes assume that both training and test data follow a conditional distribution which belongs to a given parametric hypothesis class, $\{p(y|x, \theta)\}$. This assumption cannot be verified on real-world data, particularly in privacy-sensitive applications where real user data cannot be annotated [12] and the unlabeled pool may contain irrelevant information. In such cases, choosing samples from the unlabeled pool may not necessarily improve model performance on the test set. As an alternative to making distributional assumptions, we build upon the *individual setting* [13]. This setting does not assume any probabilistic connection between the training and test data. Moreover, the relationship between labels and data can even be determined by an adversary. The generalization error in this setting is known as the *regret* [14], which is defined as the log-loss difference between a learner and a *genie*: a learner that knows the specific test label but is constrained to use an explanation from a set of hypotheses. The predictive Normalized Maximum Likelihood (pNML) learner [14] was proposed as the min-max solution of the regret, where the minimum is over the learner choice and the maximum is for any possible test label value. The pNML was previously developed for linear regression [15] and was evaluated empirically for DNN [16].

The setting considered in this work, i.e., active learning with no distributional assumption, is related to the active online learning literature [17,18], which deals primarily with task-agnostic learning that does not assume a connection between the training and test tasks. The research in Yoo and Kweon [17] proposed an active learning method that works efficiently with deep networks. A small parametric module, named “loss prediction module”, is attached to a target network, and learns it to predict target losses of unlabeled inputs. Then, this module can suggest data for which the target model is likely to produce a wrong prediction. This method is task-agnostic, as networks are learned from a single loss regardless of target tasks. The research in Sinha et al. [18] suggested a pool-based semi-supervised active learning algorithm that implicitly learns a sampling mechanism in an adversarial manner. Unlike conventional active learning algorithms, this approach is task-agnostic, i.e., it does not depend on the performance of the task for which we are trying to acquire labeled data. This method learns a latent space using a variational autoencoder

(VAE) and an adversarial network trained to discriminate between unlabeled and labeled data. The minimax game between the VAE and the adversarial network is played such that while the VAE tries to trick the adversarial network into predicting that all data points are from the labeled pool, the adversarial network learns how to discriminate between dissimilarities in the latent space.

Moreover, as an additional incentive for the individual setting, in scenarios involving Out-Of-Distribution (OOD) data, the application of uncertainty-based Active Learning (AL) without meticulous consideration may increase the likelihood of selecting OOD samples for labeling, surpassing the selection of in-distribution (IND) data. OOD data typically demonstrate high uncertainty, leading the AL algorithm to preferentially choose such samples for labeling, thereby inefficiently utilizing the labeling budget. Consequently, there is an urgent need for active learning methods resilient to such scenarios.

While empirical evidence has demonstrated the real-life impact of the OOD problem on AL [19], there is a scarcity of research addressing this crucial issue. The research in Kothawade et al. [20] approached OOD as a sub-task, and its sub-modular mutual information-based sampling scheme is marked by both time and memory consumption. In contrast, Du et al. [21] mandated the pre-training of additional self-supervised models like SimCLR [22], introducing hyperparameters to balance semantic and distinctive scores. The values of these hyperparameters exert a significant influence on the final performance, thereby limiting the broader applicability of the proposed approach.

In addition to the challenges highlighted in the aforementioned context, another promising avenue of research explores counterfactual training [23] to enhance OOD generalization. This approach involves learning model parameters by comparing pairs of factual samples and counterfactual samples, illustrating how changes in features lead to changes in labels. Notably, modifications to causal features and labels disrupt spurious correlations, as non-causal features are present in both factual and counterfactual samples with distinct classes [24]. Through counterfactual training, the model avoids relying on spurious correlations for predictions, enhancing its ability for OOD generalization [24,25]. This approach effectively breaks the link between non-causal features and labels, contributing to an improved OOD generalization capability. Nevertheless, counterfactual learning may be considered less feasible, as generating meaningful counterfactual samples requires sufficient and representative data, which may be challenging to obtain in some cases, especially if the dataset is limited or biased.

The research in Shayovitz and Feder [26] proposed an active learning criterion for the individual setting that takes into account a trained model, the unlabeled pool, and a small set of unlabeled test features. This criterion, denoted IAL (Individual Active Learning), is designed to select a sample to be labeled in such a way that, when added to the training set with its worst-case label, it attains the minimal pNML regret for the test set. The algorithm proposed by Shayovitz and Feder [26] for Gaussian Process Classification is based on an Expectation Propagation approximation of the model posterior. This approximation is both computationally expensive for large-scale deep neural networks (DNNs) and does not provide good enough performance in empirical tests. The computational complexity associated with the re-training for each candidate sample is extremely demanding.

Main Contributions

Our contributions can be succinctly outlined as follows:

- In this investigation, we address AL in the presence of OOD challenges by utilizing a small unlabeled sample from the test distribution. We focus on the individual data setting and leverage an existing active learning criterion [26]. However, the computation of this criterion is deemed impractical for DNNs.
- Our primary contribution lies in the development of an efficient algorithm aimed at mitigating the challenging computational complexity associated with approximating the mentioned criterion for neural networks. Termed DIAL (Deep Individual Ac-

tive Learning), this algorithm facilitates faster and more practical implementation of Individual Active Learning (IAL) for DNNs.

- We demonstrate that, in the presence of OOD samples, our algorithm requires only 66.2%, 91.9%, and 77.2% of labeled samples compared to recent leading methods for CIFAR10 [27], EMNIST [28], and MNIST [29] datasets, respectively, for the same accuracy level. When considering only IND samples, our approach necessitates 64.9%, 99.0%, and 64.9% labeled samples on the aforementioned datasets.
- In OOD scenarios, DIAL does not rely on the annotator to provide semantic information or counterfactual examples. The criterion is universally applicable across various datasets and can be implemented immediately.

This paper is organized as follows. In Section 2, the individual learning setting is introduced and the pNML is reviewed. In Section 3, IAL is presented and motivated by the minimax regret problem discussed in the previous section. In Section 4, IAL is applied to the DNN hypothesis class and a novel low-complexity algorithm denoted as DIAL is presented. In Section 5, the performance of DIAL is analyzed in comparison with state-of-the-art deep active learning algorithms. Throughout this paper, a sequence of samples will be denoted $x^n = (x_1, x_2, \dots, x_n)$. The variables $x \in \mathbb{X}$ and $y \in \mathbb{Y}$ will represent the features and labels, respectively, with \mathbb{X} and \mathbb{Y} being the sets containing the features' and labels' alphabets, respectively.

2. The Individual Data Setting

In the supervised learning framework, a training set consisting of n pairs of examples is provided to the learner:

$$z^n = \{(x_i, y_i)\}_{i=1}^n \quad (4)$$

where x_i is the i -th data point and y_i is its corresponding label. The goal of a learner is to predict an unknown test label y given its test data, x , by assigning a probability distribution $q(\cdot|x, z^n)$ for each training set z^n .

In the commonly used stochastic setting as defined in [13], the data follow a distribution assumed to be part of some parametric family of hypotheses. A more general framework, named *individual setting* [13], does not assume that there exists some probabilistic relation between a feature x and a label y , and so the sequence $z^n = \{x^n, y^n\}$ is an individual sequence where the relation can even be set by an adversary. Since there is no distribution over the data, finding the optimal learner, $q(\cdot|x, z^n)$, is an ill-posed problem. In order to mitigate this problem, an alternative objective is proposed: find a learner $q(\cdot|x, z^n)$ which performs as well as a reference learner on the test set.

Denote Θ as a general index set. Let P_Θ be a set of conditional probability distributions:

$$P_\Theta = \{p(y|x, \theta) | \theta \in \Theta\} \quad (5)$$

It is assumed that the reference learner knows the test label value y but is restricted to using a model from the given hypothesis set P_Θ . This reference learner then chooses a model, $\hat{\theta}(x, y, z^n)$, that attains the minimum loss over the training set and the test sample:

$$\hat{\theta} = \arg \max_{\theta \in \Theta} [p(y|x, \theta) w(\theta) \prod_{i=1}^n p(y_i|x_i, \theta)] \quad (6)$$

where performance is evaluated using the log-loss function, i.e., $-\log(q(\cdot|x, z^n))$.

Note that, in this work, we extended the individual setting of [30] and allowed the usage of some prior $w(\theta)$ over the parameter space, which may be useful for regularization purposes. The learning problem is defined as the log-loss difference between a learner q and the reference learner (genie):

$$R_n(q, y; x) = \log \frac{p(y|x, \hat{\theta})}{q(y|x, z^n)}. \quad (7)$$

An important result for this setting is provided in Fogel and Feder [14] and provides a closed-form expression for the minimax regret along with the optimal learner, q_{pNML} :

Theorem 1 (Fogel and Feder [14]). *The universal learner, denoted as the pNML, minimizes the worst case regret:*

$$R_n(x) = \min_q \max_{y \in \mathbb{Y}} \log \left(\frac{p(y|x, \hat{\theta})}{q(y|x, z^n)} \right)$$

The pNML probability assignment and regret are:

$$q_{\text{pNML}}(y|x, z^n) = \frac{p(y|x, \hat{\theta})}{\sum_y p(y|x, \hat{\theta})}$$

$$R_n(x) = \log \sum_{y \in \mathbb{Y}} p(y|x, \hat{\theta})$$

Since the main contribution of this work relies on this theorem, we provide a short proof here:

Proof. We note that the regret, $R_n(x)$, is equal for all choices of y . Now, if we consider a different probability assignment, then it would assign a smaller probability for at least one of the possible outcomes. In this case, choosing one of those outcomes will lead to a higher regret and then the maximal regret will be higher, leading to a contradiction. \square

The pNML regret is associated with the *stochastic complexity* of a hypothesis class, as discussed by Rosas et al. [31] and Zhou and Levine [16]. It is clear that for pNML, a model that fits almost every data pattern would be much more complex than a model that provides a relatively good fit to a small set of data. Thus, high pNML regret indicates that the model class may be too expressive and overfit. The pNML learner is the min-max solution for supervised batch learning in the individual setting [14]. For sequential prediction it is termed the conditional normalized maximum likelihood [32,33].

Several methods deal with obtaining the pNML learner for different hypothesis sets. The research in Bibas et al. [15] and Bibas and Feder [34] showed the pNML solution for linear regression. The research in Rosas et al. [35] proposed an NML-based decision strategy for supervised classification problems and showed that it attains heuristic PAC learning. The research in Fu and Levine [36] used the pNML for model optimization based on learning a density function by discretizing the space and fitting a distinct model for each value. For the DNN hypothesis set, Bibas et al. [37] estimated the pNML distribution with DNN by fine-tuning the last layers of the network for every test input and label combination. This approach is computationally expensive since training is needed for every test input. The research in Zhou and Levine [16] suggested a way to accelerate the pNML computation in DNN by using approximate Bayesian inference techniques to produce a tractable approximation to the pNML.

3. Active Learning for Individual Data

In active learning, the learner sequentially selects data instances x_i based on some criterion and produces n training examples: z^n . The objective is to select a subset of the unlabelled pool and derive a probabilistic learner $q(y|x, z^n)$ that attains the minimal prediction error (on the test set) among all training sets of the same size. Most selection criteria are based on uncertainty quantification of data instances to quantify their informativeness. However, in the individual setting, there is no natural uncertainty measure, since there is no distribution governing the data.

As proposed in [26], the min-max regret R_n as defined in Theorem 1 is used as an active learning criterion, which essentially quantifies the prediction performance of the training set z^n for a given unlabeled test feature x . A “good” z^n minimizes the min-max

regret for any test feature and thus provides good test set performance. Since R_n is a point-wise quantity, the average over all test data is taken:

$$C_n = \min_{x^n} \max_{y^n} \sum_x \log \left(\sum_y p(y|x, \hat{\theta}) \right) \quad (8)$$

where $\hat{\theta} = \hat{\theta}(x, y, z^n)$ is the Maximum Likelihood estimator, as defined in (6).

The idea is to find a set of training points, x^n , that minimizes the averaged log normalization factor (across unlabeled test points) for the worst possible labels y^n . This criterion looks for the worst-case scenario since there is no assumption on the data distribution. Since (8) selects a batch of points x^n , it is computationally prohibitive to solve for a general hypothesis class. In order to reduce complexity, a greedy approach denoted Individual Active Learning (IAL) is proposed in [26] which performs well empirically:

$$C_{n|n-1} = \min_{x_n} \max_{y_n} \sum_x \log \left(\sum_y p(y|x, \hat{\theta}) \right) \quad (9)$$

Note that when computing (9), the previously labeled training set, z^{n-1} , is assumed to be available for the learner and $\hat{\theta} = \hat{\theta}(x, y, x_n, y_n, z^{n-1})$. The objective in (9) is to find a single point x_n from the unlabelled pool as opposed to the objective in (8) that tries to find an optimal batch x^n .

4. Deep Individual Active Learning

The DNN (deep neural network) hypothesis class poses a challenging problem for information-theoretic active learning since its parameter space is of very high dimension and the weights' posterior distribution (assuming a Bayesian setting) is analytically intractable. Moreover, direct application of deep active learning schemes is unfeasible for real-world large-scale data, since it requires training the entire model for each possible training point. To make matters worse, for IAL, the network also needs to be trained for every test point and every possible corresponding label.

In this section, we derive an approximation of IAL for DNNs which is based on variational inference algorithms [4,38,39]. We define the hypothesis class in this case as follows:

$$p(y|x, \theta) = \text{softmax}(f_\theta(x)) \quad (10)$$

where θ represents all the weights and biases of the network and $f_\theta(x)$ is the model output before the last softmax layer. Note that x , y , and $p(\theta)$ represent the test feature, test label, and prior on the weights, respectively.

The MAP estimation for θ is:

$$\hat{\theta} = \arg \max_{\theta} p(y^n, y|x^n, x, \theta)p(\theta), \quad (11)$$

where the prior $p(\theta)$ acts as a regularizer over the latent vector θ . It is common practice to use some regularization mechanism to control the training error for DNNs. In order to embed the regularization mechanism into the MAP, we introduced this prior $p(\theta)$.

Given a training set x^n, y^n and test couple x, y , the maximization in (11) is performed by training the DNN with all the data and converging to a steady-state maximum. Note that x^{n-1}, y^{n-1} are assumed to be known, while x_n, y_n, x and y are not known, and all the different possibilities need to be considered, resulting in multiple training sessions of the network. In order to avoid re-training the entire network for all possible values of x, y, x_n , and y_n , we utilize the independence between soft-max scores in the MAP estimation. Using Bayes, we observe that (11) can be re-written as:

$$\hat{\theta} = \arg \max_{\theta} p(y|x, \theta)p(y_n|x_n, \theta)p(\theta|y^{n-1}, x^{n-1}) \quad (12)$$

where $p(\theta|y^{n-1}, x^{n-1})$ is the posterior of θ given the available data $z^{n-1} = (x^{n-1}, y^{n-1})$.

The posterior $p(\theta|z^{n-1})$ is not dependent on the test data (x, y) and the evaluated labeling candidate (x_n, y_n) , and thus can be computed once per selection iteration and then used throughout the IAL selection process. This is a very important point which needs to be highlighted; there is no need to re-train the network for every (x, y) and (x_n, y_n) . We only need to train the network using x^{n-1}, y^{n-1} and then, during the IAL selection process, run forward passes on different θ with high $p(\theta|z^{n-1})$ values, to compute $p(y|x, \theta)$ and $p(y_n|x_n, \theta)$. This fact represents a significant computational complexity reduction since the number of possible points x_n can be significant and we wish to avoid re-training the network for each point.

In order to acquire the weight posterior for a DNN, some advanced techniques are required [40–42]; these involve multiple training passes over the network. For a DNN, the posterior, $p(\theta|y^{n-1}, x^{n-1})$, is multi-modal and intractable to compute directly. Therefore, we propose approximating it by some simpler distribution, which will allow easier computation of the maximum likelihood $\hat{\theta}$.

4.1. Variational Inference

Variational inference is a technique used in probabilistic modeling to approximate complex probability distributions that are difficult or impossible to calculate exactly [42–44]. Variational inference has been used in a wide range of applications, including in Bayesian neural networks, latent Dirichlet allocation, and Gaussian processes. The goal of variational inference is to find an approximation, $q^*(\theta)$ from a parametric family \mathbb{Q} , to the true distribution, $p(\theta|z^{n-1})$, that is as close as possible to the true distribution, but is also computationally tractable. This goal is formulated as minimizing the Kullback–Leibler (KL) divergence between the two distributions (also called information projection):

$$q^*(\theta) = \underset{q \in \mathbb{Q}}{\operatorname{argmin}} D_{KL}(q(\theta) || p(\theta|z^{n-1}))$$

There are different algorithms for implementing variational inference; most involve optimizing a lower bound on the log-likelihood of the data under the true distribution (called evidence). The lower bound is defined as the difference between the true distribution's data log-likelihood and the Kullback–Leibler (KL) divergence between the true distribution and the approximation. The KL divergence measures the distance between the two distributions, and so optimizing the lower bound is equivalent to minimizing the distance between the true distribution and the approximation.

One common algorithm for implementing variational inference is called mean field variational inference [45]. In this approach, the approximation to the true distribution is factorized into simpler distributions that are easier to work with, such as Gaussians or Bernoullis. The parameters of these simpler distributions are then optimized to minimize the KL divergence between the true distribution and the approximation. Another algorithm for variational inference is called stochastic variational inference [46]. In this approach, the optimization is performed using stochastic gradient descent, with a random subset of the data used in each iteration. This allows the algorithm to scale to large datasets and complex models.

4.2. Deep Individual Active Learning (DIAL)

In this work, we opted to use the method in Gal and Ghahramani [41], denoted as MC dropout (Monte Carlo dropout), due to its computational simplicity and favorable performance. MC dropout represents a sophisticated extension of the conventional dropout regularization technique within the domain of machine learning, and it is particularly associated with improving the robustness and uncertainty quantification of neural networks. This concept finds its roots in the broader effort to address the challenge of overfitting, a common concern in training deep learning models where the network becomes excessively attuned to the training data, hindering its generalization to new, unseen data.

Traditional dropout involves randomly deactivating, or “dropping out”, a fraction of the neurons during the training phase. This stochastic process introduces a level of noise, preventing the neural network from relying too heavily on specific features, thus enhancing its ability to generalize to diverse datasets. However, dropout is typically applied solely during the training phase, and the model’s predictions during the inference phase are based on a single deterministic forward pass through the network.

Monte Carlo dropout introduces a novel approach to the inference phase by extending the dropout mechanism beyond training. In this context, during inference, the model performs multiple forward passes with different dropout masks applied each time. This process generates a set of predictions, and the final output is obtained by averaging or aggregating these predictions. The rationale behind this technique lies in its ability to capture and quantify uncertainty associated with the model’s predictions.

By leveraging Monte Carlo dropout during inference, practitioners can gain valuable insights into the uncertainty inherent in the model’s predictions. This uncertainty is crucial in real-world applications where understanding the model’s confidence level is essential. For instance, in autonomous vehicles, medical diagnostics, or financial predictions, knowing the uncertainty associated with a model’s output can inform decision making and improve overall system reliability.

In Gal and Ghahramani [41], the authors argued that performing dropout during training on DNNs, with dropout applied before every weight layer, is mathematically equivalent to minimizing the KL divergence between the weight posterior of the full network and a parametric distribution which is controlled by a set of Bernoulli random variables defined by the dropout probability. Therefore, $p(\theta|y^{n-1}, x^{n-1})$ can be approximated in KL-sense by a distribution which is controlled by the dropout parameter. We can use this idea in order to approximate (12) and find an approximated weight distribution, $q(\theta)$. Therefore, we can re-write (12) using the variational approximation $q(\theta)$:

$$\hat{\theta} \approx \arg \max_{\theta} p(y|x, \theta) p(y_n|x_n, \theta) q(\theta) \quad (13)$$

However, $q(\theta)$ as described in Gal and Ghahramani [41] is still complex to analytically compute. In fact, in Gal and Ghahramani [41], the authors do not explicitly sample from this distribution but compute integral quantities on this distribution (such as expectation and variance) using averaging of multiple independent realizations and the Law of Large Numbers (LLN). Since we focus on point-wise samples from $q(\theta)$, we cannot use the same approach as in Gal and Ghahramani [41].

In this work, we propose to sample M weights from $q(\theta)$ and find $\hat{\theta}$ among all the different samples. Since the weights are embedded in a high-dimensional space, the probability of the sampled weights can be assumed to be relatively uniform. Therefore, we propose approximating (13) as:

$$\hat{\theta} \approx \arg \max_{\{\theta_m\}_{m=1}^M} p(y|x, \theta_m) p(y_n|x_n, \theta_m) \quad (14)$$

As observed by Gal and Ghahramani [41], (14) can be computed by running multiple forward passes on the network trained using dropout with z^{n-1} during inference with x and x_n . The resulting algorithm, denoted Deep Individual Active Learning (DIAL), is shown in Algorithm 1 and follows these steps:

1. Train a model on the labeled training set z^{n-1} with dropout.
2. For each pair of x and x_n , run M forward passes with different dropout masks and compute the product of the softmax outputs.
3. Find the weight that maximizes DNN prediction of the test input and the unlabeled candidate input as in (12).
4. Accumulate the pNML regret of the test point given these estimations.
5. Find the unlabeled candidate for which the worst-case averaged regret of the test set is minimal, as in (9).

For step 2, since the variational posterior associated with MC dropout is difficult to evaluate, we assume that it is uniform for all the sampled weights.

We emphasize the significant complexity reduction provided by our approximation; a naïve implementation of pNML computation would require training the network over all possible training points x_n and test points x with all possibilities of their respective labels y_n, y . This would render our criterion unfeasible for real-world applications. Our proposed approach, DIAL, only requires performing training with dropout on z^{n-1} only once per selection iteration and then performing forward passes (considerably faster than training passes) to obtain multiple samples of the weights.

Algorithm 1: DIAL: Deep Individual Active Learning

Input Training set z^{n-1} , unlabeled pool and test samples $\{x_i\}_{i=1}^N$ and $\{x_k\}_{k=1}^K$.
Output Next data point for labeling \hat{x}_i
 Run MC-Dropout using z^{n-1} to get $\{\theta_m\}_{m=1}^M$
 $\mathbf{S} = \text{zeros}(N, |\mathbb{Y}|)$
for $i \leftarrow 1$ to N **do**
 for $y_i \in \mathbb{Y}$ **do**
 for $k \leftarrow 1$ to K **do**
 $\Gamma = 0$
 for $y_k \in \mathbb{Y}$ **do**
 $\hat{\theta} = \text{argmax}_{\theta_m} p(y_k|x_k, \theta_m) p(y_i|x_i, \theta_m)$
 $\Gamma = \Gamma + p(y_k|x_k, \hat{\theta})$
 end for
 $\mathbf{S}(i, y_i) = \mathbf{S}(i, y_i) + \log \Gamma$
 end for
 end for
end for
 $\hat{x}_i = \text{argmin}_{x_i} \max_{y_i} \mathbf{S}$

5. Experiments

In this section, we analyze the performance of DIAL and compare its performance to state-of-the-art active learning criteria. We tested the proposed DIAL strategy in two scenarios:

- The initial training, unlabeled pool, and test data come from the same distribution (IND scenario).
- There are OOD samples present in the unlabeled pool (OOD scenario).

The reason for using the individual setting and DIAL as its associated strategy in the presence of OOD samples is that it does not make any assumptions about the data generation process, making the results applicable to a wide range of scenarios, including PAC [47], stochastic [13], adversarial settings, as well as samples from unknown distributions.

We considered the following datasets for training and evaluation of the different active learning methods:

- **The MNIST dataset** [29] consists of 28×28 grayscale images of handwritten digits, with 60 K images for training and 10 K images for testing.
- **The EMNIST dataset** [28] is a variant of the MNIST dataset that includes a larger variety of images (upper and lower case letters, digits, and symbols). It consists of 240 K images with 47 different labels.
- **The CIFAR10 dataset** [27] consists of 60 K 32×32 color images in 10 classes. The classes include objects such as airplanes, cars, birds, and ships.
- **Fashion MNIST** [48] is a dataset of images of clothing and accessories, consisting of 70 K images. Each image is 28×28 grayscale pixels.

- **The SVHN dataset** [49] contains 600 K real-world images with digits and numbers in natural scene images collected from Google Street View.

We built upon Huang [50] and Smith et al. [11] open-source implementations of the following methods:

The Random sampling algorithm is the most basic approach in learning. It selects samples to label randomly, without considering any other criteria. This method can be useful when the data are relatively homogeneous and easy to classify, but it can be less efficient when the data are more complex or when there is a high degree of uncertainty.

The Bayesian Active Learning by Disagreement (BALD) method [4] utilizes an acquisition function that calculates the mutual information between the model's predictions and the model's parameters. This function measures how closely the predictions for a specific data point are linked to the model's parameters, indicating that determining the true label of samples with high mutual information would also provide insight into the true model parameters.

The Core-set algorithm [5] aims to find a small subset from a large labeled dataset such that a model learned from this subset will perform well on the entire dataset. The associated active learning algorithm chooses a subset that minimizes this bound, which is equivalent to the k-center problem.

The Expected Predictive Information Gain (EPiG) method [11] was motivated by BALD's weakness in prediction-oriented settings. This acquisition function directly targets a reduction in predictive uncertainty on inputs of interest by utilizing the unlabelled test set. It is shown in Appendix A that this approach is similar to UAL [6], where the main difference is that UAL assumes the stochastic setting, where the data follow some parametric distribution.

5.1. Experimental Setup

The first setting we consider consists of an initial training set, an unlabeled pool (from which the training examples are selected), and an unlabeled test set, all drawn from the **same distribution**. The experiment includes the following four steps:

1. A model is trained on the small labeled dataset (initial training set).
2. One of the active learning strategies is utilized to select a small number of the most informative examples from the unlabeled pool.
3. The labels of the selected samples are queried and added to the labeled dataset.
4. The model is retrained using the new training set.

Steps 2–4 are repeated multiple times, with the model becoming more accurate with each iteration, as it is trained on a larger labeled dataset.

In addition to the standard setting, we evaluate the performance in **the presence of OOD samples**. In this scenario, the initial training and test sets are drawn from the same distribution, but the unlabeled pool contains a mix of OOD samples. When an OOD unlabeled sample is selected for annotation, it is not used in training of the next iteration of the model. Across all x-axis values in the subsequent test accuracy figures, the presented metric is the count of Oracle calls, reflecting the instances when a selection strategy chose a sample, whether it be IND or OOD. It is crucial to differentiate this metric from the training set size, as the selection of an OOD sample leads to an increase in the number of Oracle calls, while the training set size remains unaffected. An effective strategy would recognize that OOD samples do not improve performance on the test set and avoid selecting them.

A visual representation of the scenario with OOD samples is illustrated in Figure 1a–c. These figures show the unlabeled pool, which contains a mixture of both IND and OOD samples. Figure 1d–f show the test set, which contains only IND samples. We argue that this is a representative setting for active learning in real life. In the real world, unlabelled pools are collected from many data sources and will most certainly contain OOD data. The process of pruning the unlabelled pool is a costly process and involves human inspection and labeling, which needs to be minimized. This is exactly the goal of active learning and finding a criterion which implicitly filters OOD data is of significant interest.

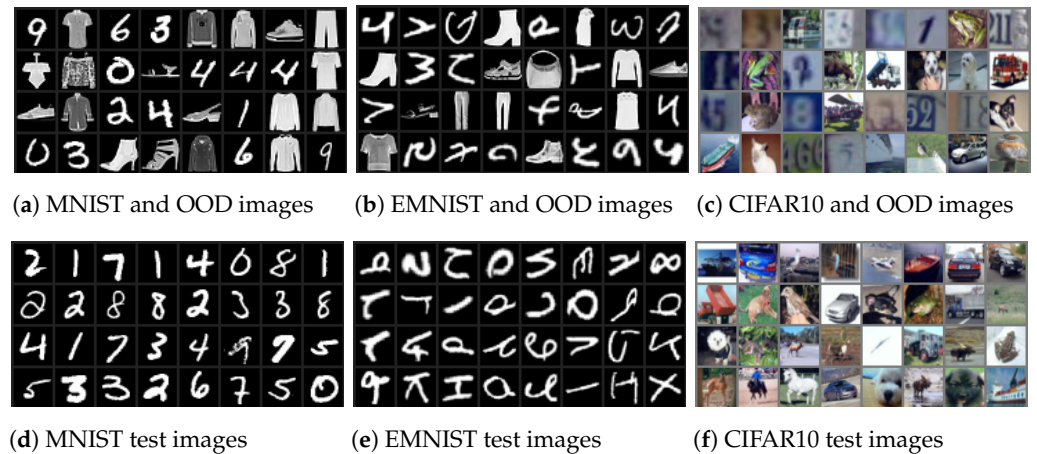


Figure 1. Datasets that contain a mix of images with OOD samples. (Top) Unlabeled pool contains OOD samples (Bottom). Test set includes only valid data.

5.2. MNIST Experimental Results

Following Gal et al. [4], we considered a model consisting of two blocks of convolution, dropout, max-pooling, and ReLu, with 32 and 64 5×5 convolution filters. These blocks are followed by two fully connected layers that include dropout between them. The layers have 128 and 10 hidden units, respectively. The dropout probability was set to 0.5 in all three locations. In each active learning round, a single sample was selected. We executed all active learning methods six times with different random seeds. For BALD, EPIG, and DIAL, we used 100 dropout iterations and employed the criterion on 512 random samples from the unlabeled pool. MNIST results are shown in Figure 2a. The largest efficiency is at a number of Oracle calls of 71, where DIAL attains an accuracy rate of 0.9, while EPIG and BALD achieve an accuracy rate of 0.86.

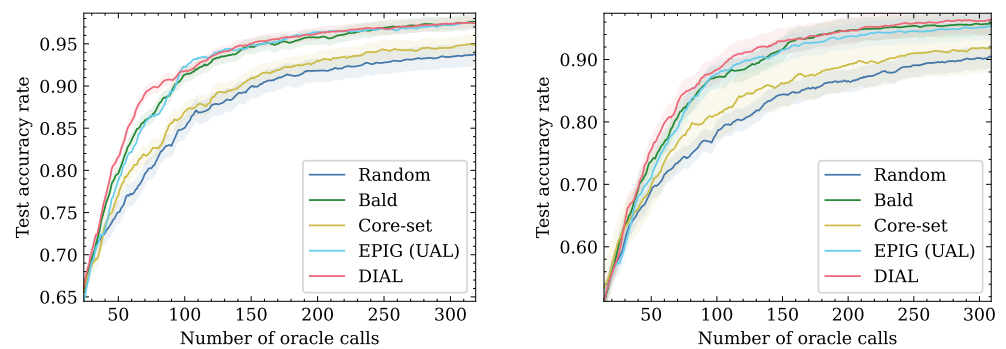


Figure 2. Accuracy as function of number of Oracle calls on MNIST dataset. DIAL outperforms the baselines for the two setups.

To simulate the presence of OOD samples, we added the Fashion MNIST to the unlabeled pool such that the ratio of Fashion MNIST to MNIST was 1:1. In this setting, DIAL outperforms all other baselines, as shown in Figure 2b. DIAL is the top-performing method and has better accuracy than EPIG, BALD, Core-set, and Random. The largest efficiency is an accuracy rate of 0.95, where DIAL uses 240 Oracle calls, while BALD needs 307 (−35.1%). EPIG never reaches this accuracy level. The number of Oracle calls for additional accuracy rates is shown in Table 1.

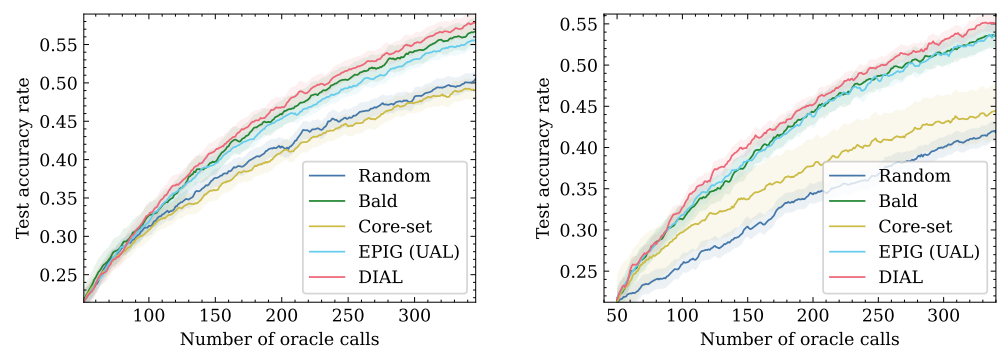
Table 1. MNIST with OOD number of Oracle calls at x% accuracy.

Methods	85% Acc.	75% Acc.	65% Acc.
Random	145	73	36
Core-set	117	61	33
BALD	83	51	32
EPIG	84	56	35
DIAL	73 (−12.1%)	48 (−5.9%)	30 (−6.2%)

5.3. EMNIST Experimental Results

We followed the same setting as the MNIST experiment with a slightly larger model than MNIST consisting of three blocks of convolution, dropout, max-pooling, and ReLU. The experimental results, shown in Figure 3a, indicate that DIAL is the top-performing method. For an accuracy rate of 0.56, it requires 8.3% less Oracle calls when compared to the second best method.

In the presence of OOD samples, the DIAL method outperforms all other baselines, as shown in Figure 3b and Table 2. For 300 Oracle calls, DIAL achieves a test set accuracy rate of 0.52, while BALD, EPIG, Core-set, and Random attain 0.51, 0.5, 0.42, and 0.40, respectively. For an accuracy rate of 0.53, DIAL needs 308 Oracle calls, while BALD and EPIG require 346 and 342, respectively (−11%). Moreover, Core-set and Random do not achieve this accuracy.



(a) EMNIST

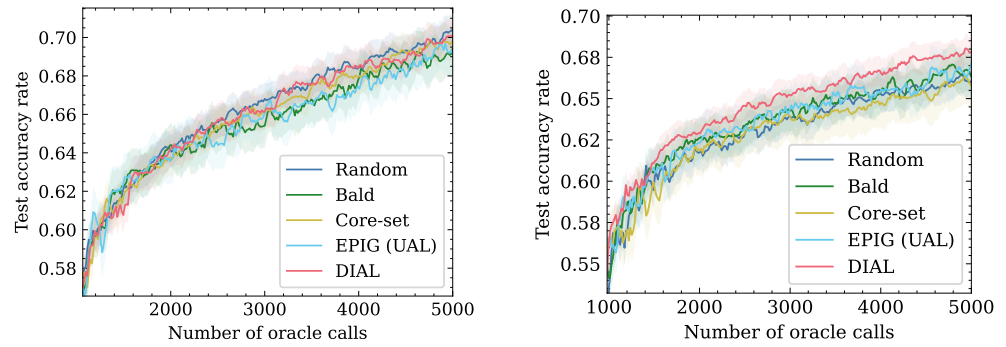
(b) EMNIST with OOD

Figure 3. Active learning performance on the EMNIST dataset. DIAL is more efficient than tested baselines in the number of Oracle calls.**Table 2.** EMNIST with OOD number of Oracle calls at x% accuracy.

Methods	40% Acc.	30% Acc.	25% Acc.
Random	281	140	80
Core-set	221	96	62
BALD	154	85	59
EPIG	157	84	59
DIAL	138 (−10.4%)	84 (−1.2%)	59 (0%)

5.4. Cifar10 Experimental Results

For the CIFAR10 dataset, we utilized ResNet-18 [51] with an acquisition size of 16 samples. We used 1K initial training set size and measured the performance of the active learning strategies up to a training set size of 3K. The CIFAR10 results are shown in Figure 4a. Overall, DIAL and Random perform the same and have a better test set accuracy than the other baselines for Oracle calls greater than 2100.



(a) CIFAR10

(b) CIFAR10 with OOD

Figure 4. The left figure illustrates the performance of CIFAR10 using only IND samples. The DIAL method performs similarly to the Random method. The figure on the right shows the performance of a combination of OOD samples, where DIAL outperforms all other methods.

When the presence of OOD samples in the unlabeled pool is considered, as shown in Figure 4b, DIAL outperforms the other methods. Table 3 shows the number of Oracle calls required for different accuracy levels. For the same accuracy rate of 0.65, DIAL needs up to 15.4% less Oracle calls than the second best method. This can be explained by Figure 5, which shows the ratio of OOD samples to the number of Oracle calls. The figure suggests that DIAL outperforms other criteria by selecting fewer OOD samples, contributing to its commendable performance. It is noteworthy that in all OOD scenarios, DIAL demonstrated superior ability to identify in-distribution samples without explicit knowledge of the distribution and solely utilizing unlabeled test features. This underscores the universality of DIAL, showcasing its adaptability to various distribution shifts. Additionally, the second-best performer, EPIG, also considers the unlabeled test set and performs better than other baseline methods but falls short of DIAL. Notably, BALD and Core-set exhibit similar behavior, possibly attributed to their emphasis on model estimation rather than leveraging the test set for predictive focus.

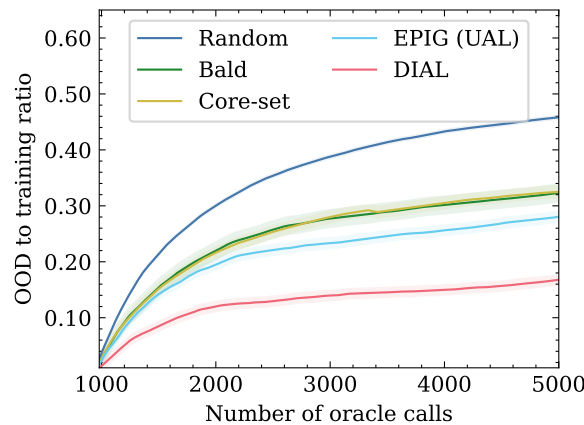


Figure 5. The amount of chosen OOD samples for CIFAR10 with the presence of OOD samples.

Table 3. CIFAR10: the presence of OOD samples: Number of Oracle calls at specific accuracy rate values.

Methods	66% Acc.	62% Acc.	58% Acc.
Random	3956	1828	1220
Core-set	4468	1844	1412
BALD	4020	1636	1202
EPIG	3636	1700	1108
DIAL	3076 (−15.4%)	1556 (−4.9%)	1060 (−4.3%)

6. Limitations

The proposed DIAL algorithm is a min-max strategy for the individual settings. However, DIAL may not be the most beneficial approach in scenarios where the unlabeled pool is very similar to the test set, where different selection strategies may perform similarly and with smaller complexity. This limitation of DIAL is supported by the experimental results of Section 5.4, where the DIAL algorithm performed similarly to random selection for the CIFAR10 dataset (but better than all the other baselines).

Another limitation of DIAL is that it has a higher overhead computation compared to other active learning methods such as BALD. This is because DIAL involves computing the regret on the test set, which requires additional computations and could become significant when the unlabeled pool or the test set are very large.

7. Conclusions

In this study, we propose a min-max active learning criterion for the individual setting, which does not rely on any distributional assumptions. We have also developed an efficient method for computing this criterion for DNNs. Our experimental results demonstrate that the proposed strategy, referred to as DIAL, is particularly effective in the presence of OOD samples in the unlabeled pool. Specifically, our results show that DIAL requires 12%, 10.4%, and 15.4% fewer Oracle calls than the next best method to achieve a certain level of accuracy on the MNIST, EMNIST, and CIFAR10 datasets, respectively.

As future work, we plan to investigate batch acquisition criteria that take into account batch selection. This will allow us to consider the relationship between the selected samples and the overall composition of the batch, which may lead to even further improvements in performance.

Author Contributions: K.B., coding, visualization, writing—review and editing; S.S., conceptualization, formal analysis, writing—review and editing; M.F., writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The paper itself contains all the information required to assess the conclusions. Additional data related to this paper may be requested from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Equivalence between EPIG and UAL

In this section, we will prove that the criterion proposed by Smith et al. [11] is equivalent to the criterion in Shayovitz and Feder [6].

Proof. Assuming some prior $\pi(\theta)$, the UAL criterion is:

$$\hat{x}_n = \underset{x_n}{\operatorname{argmin}} I(\theta; Y|X, x_n, Y_n, z^{n-1})$$

where X and Y are the test feature and label random variables.

The EPIG criterion [11] is:

$$\hat{x}_n = \underset{x_n}{\operatorname{argmax}} I(Y; Y_n|X, x_n, z^{n-1})$$

In order to show the equivalence between the two acquisition functions (UAL and EPIG), we can write the following mutual information equality using the mutual information chain rule:

$$I(Y; Y_n, \theta | X, x_n, z^{n-1}) = I(Y; Y_n | X, x_n, z^{n-1}) + I(Y; \theta | X, Y_n, x_n, z^{n-1})$$

The same mutual information $I(Y; Y_n, \theta | X, x_n, z^{n-1})$ can also be expressed by the chain rule using a different conditioning:

$$I(Y; Y_n, \theta | X, x_n, z^{n-1}) = I(Y; \theta | X, x_n, z^{n-1}) + I(Y; Y_n | X, x_n, \theta, z^{n-1})$$

Therefore,

$$I(Y; Y_n | X, x_n, z^{n-1}) + I(Y; \theta | X, Y_n, x_n, z^{n-1}) = I(Y; \theta | X, x_n, z^{n-1}) + I(Y; Y_n | X, x_n, \theta, z^{n-1}) \quad (\text{A1})$$

Given θ , the test and train are independent; therefore, $I(Y; Y_n | X, x_n, \theta, z^{n-1}) = 0$ and (A1) becomes:

$$I(Y; \theta | X, Y_n, x_n, z^{n-1}) = I(Y; \theta | X, x_n, z^{n-1}) + I(Y; Y_n | X, x_n, \theta, z^{n-1}) \quad (\text{A2})$$

It is assumed that θ is independent of x_n ($p(\theta | x_n) = p(\theta)$) and that $p(z^{n-1} | x_n) = p(z^{n-1})$. These assumptions, coupled with Bayes, lead to the following conditional independence: $p(\theta | x_n, z^{n-1}) = p(\theta | z^{n-1})$. We can now remove the conditioning on x_n and obtain $I(Y; \theta | X, x_n, z^{n-1}) = I(Y; \theta | X, z^{n-1})$. We note that $I(Y; \theta | X, z^{n-1})$ is not dependent on x_n and is a fixed quantity.

Therefore, we can re-write (A3):

$$I(Y; \theta | X, z^{n-1}) = I(Y; \theta | X, Y_n, x_n, z^{n-1}) - I(Y; Y_n | X, x_n, \theta, z^{n-1}) \quad (\text{A3})$$

It is now clear that if we find x_n which minimizes $I(\theta; Y | X, x_n, Y_n, z^{n-1})$ (UAL), it will simultaneously maximize $I(Y; Y_n | X, x_n, z^{n-1})$ (EPIG), since their difference is a constant. \square

References

- 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. (CSUR)* **2021**, *54*, 1–40. [[CrossRef](#)]
- Wang, K.; Zhang, D.; Li, Y.; Zhang, R.; Lin, L. Cost-effective active learning for deep image classification. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *27*, 2591–2600. [[CrossRef](#)]
- Houlsby, N.; Huszár, F.; Ghahramani, Z.; Lengyel, M. Bayesian active learning for classification and preference learning. *arXiv* **2011**, arXiv:1112.5745.
- Gal, Y.; Islam, R.; Ghahramani, Z. Deep bayesian active learning with image data. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.
- Sener, O.; Savarese, S. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
- Shayovitz, S.; Feder, M. Universal Active Learning via Conditional Mutual Information Minimization. *IEEE J. Sel. Areas Inf. Theory* **2021**, *2*, 720–734. [[CrossRef](#)]
- Raginsky, M.; Rakhlin, A. Information-based complexity, feedback and dynamics in convex programming. *IEEE Trans. Inf. Theory* **2011**, *57*, 7036–7056. [[CrossRef](#)]
- MacKay, D.J. Information-based objective functions for active data selection. *Neural Comput.* **1992**, *4*, 590–604. [[CrossRef](#)]
- Fedorov, V.V. *Theory of Optimal Experiments*; Elsevier: Amsterdam, The Netherlands, 2013.
- Shayovitz, S.; Feder, M. Minimax Active Learning Via Minimal Model Capacity. In Proceedings of the 29th IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2019, Pittsburgh, PA, USA, 13–16 October 2019.
- Smith, F.B.; Kirsch, A.; Farquhar, S.; Gal, Y.; Foster, A.; Rainforth, T. Prediction-Oriented Bayesian Active Learning. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Valencia, Spain, 25–27 April 2023.
- Alabduljabbar, A.; Abusnaina, A.; Meteriz-Yildiran, Ü.; Mohaisen, D. TLDR: Deep Learning-Based Automated Privacy Policy Annotation with Key Policy Highlights. In Proceedings of the 20th Workshop on Workshop on Privacy in the Electronic Society, Seoul, Republic of Korea, 15 November 2021.
- Merhav, N.; Feder, M. Universal prediction. *IEEE Trans. Inf. Theory* **1998**, *44*, 2124–2147. [[CrossRef](#)]

14. Fogel, Y.; Feder, M. Universal batch learning with log-loss. In Proceedings of the International Symposium on Information Theory, Vail, CO, USA, 17–22 June 2018; IEEE: Piscataway, NJ, USA, 2018.
15. Bibas, K.; Fogel, Y.; Feder, M. A New Look at an Old Problem: A Universal Learning Approach to Linear Regression. In Proceedings of the 2019 IEEE International Symposium on Information Theory (ISIT), Paris, France, 7–12 July 2019.
16. Zhou, A.; Levine, S. Amortized Conditional Normalized Maximum Likelihood: Reliable Out of Distribution Uncertainty Estimation. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021.
17. Yoo, D.; Kweon, I.S. Learning loss for active learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
18. Sinha, S.; Ebrahimi, S.; Darrell, T. Variational adversarial active learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
19. Karamcheti, S.; Krishna, R.; Fei-Fei, L.; Manning, C.D. Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. *arXiv* **2021**, arXiv:2107.02331.
20. Kothawade, S.; Beck, N.; Killamsetty, K.; Iyer, R. Similar: Submodular information measures based active learning in realistic scenarios. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 18685–18697.
21. Du, P.; Zhao, S.; Chen, H.; Chai, S.; Chen, H.; Li, C. Contrastive coding for active learning under class distribution mismatch. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 8927–8936.
22. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, Virtual, 26–28 August 2020; pp. 1597–1607.
23. Teney, D.; Abbasnejad, E.; van den Hengel, A. Learning what makes a difference from counterfactual examples and gradient supervision. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part X 16. Springer: Berlin/Heidelberg, Germany, 2020; pp. 580–599.
24. Kaushik, D.; Hovy, E.; Lipton, Z.C. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv* **2019**, arXiv:1909.12434.
25. Nie, Y.; Williams, A.; Dinan, E.; Bansal, M.; Weston, J.; Kiela, D. Adversarial NLI: A new benchmark for natural language understanding. *arXiv* **2019**, arXiv:1910.14599.
26. Shayovitz, S.; Feder, M. Active Learning for Individual Data via Minimal Stochastic Complexity. In Proceedings of the 2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, NY, USA, 27–30 September 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–5.
27. Krizhevsky, A.; Nair, V.; Hinton, G. The CIFAR-10 Dataset. 2014. Available online: <http://www.cs.toronto.edu/kriz/cifar.html> (accessed on 29 January 2024).
28. Cohen, G.; Afshar, S.; Tapson, J.; Van Schaik, A. EMNIST: Extending MNIST to handwritten letters. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; IEEE: Piscataway, NJ, USA, 2017.
29. Deng, L. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [[CrossRef](#)]
30. Bibas, K.; Feder, M. Distribution Free Uncertainty for the Minimum Norm Solution of Over-parameterized Linear Regression. *arXiv* **2021**, arXiv:2102.07181.
31. Rosas, F.E.; Mediano, P.A.; Gastpar, M. Learning, compression, and leakage: Minimising classification error via meta-universal compression principles. In Proceedings of the 2020 IEEE Information Theory Workshop (ITW), Virtual, 11–15 April 2020; IEEE: Piscataway, NJ, USA, 2021.
32. Rissanen, J.; Roos, T. Conditional NML universal models. In Proceedings of the 2007 Information Theory and Applications Workshop, San Diego, CA, USA, 18–23 February 2007; IEEE: Piscataway, NJ, USA, 2007.
33. Roos, T.; Rissanen, J. On sequentially normalized maximum likelihood models. *Compare* **2008**, *27*, 256.
34. Bibas, K.; Feder, M. The Predictive Normalized Maximum Likelihood for Over-parameterized Linear Regression with Norm Constraint: Regret and Double Descent. *arXiv* **2021**, arXiv:2102.07181.
35. Rosas, F.E.; Mediano, P.A.; Gastpar, M. Learning, compression, and leakage: Minimizing classification error via meta-universal compression principles. *arXiv* **2020**, arXiv:2010.07382.
36. Fu, J.; Levine, S. Offline Model-Based Optimization via Normalized Maximum Likelihood Estimation. In Proceedings of the International Conference on Learning Representations, Virtual, 3–7 May 2021.
37. Bibas, K.; Fogel, Y.; Feder, M. Deep pnml: Predictive normalized maximum likelihood for deep neural networks. *arXiv* **2019**, arXiv:1904.12286.
38. Karzand, M.; Nowak, R.D. Maximin active learning in overparameterized model classes. *IEEE J. Sel. Areas Inf. Theory* **2020**, *1*, 167–177. [[CrossRef](#)]
39. Kirsch, A.; van Amersfoort, J.; Gal, Y. BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019.
40. Maddox, W.J.; Izmailov, P.; Garipov, T.; Vetrov, D.P.; Wilson, A.G. A simple baseline for bayesian uncertainty in deep learning. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.

41. Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the International Conference Machine Learning, New York, NY, USA, 19–24 June 2016.
42. Daxberger, E.; Kristiadi, A.; Immer, A.; Eschenhagen, R.; Bauer, M.; Hennig, P. Laplace redux-effortless bayesian deep learning. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 20089–20103.
43. Wilson, A.G.; Hu, Z.; Salakhutdinov, R.R.; Xing, E.P. Stochastic variational deep kernel learning. *Adv. Neural Inf. Process. Syst.* **2016**, *29*.
44. Zhang, C.; Bütepage, J.; Kjellström, H.; Mandt, S. Advances in variational inference. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 2008–2026. [[CrossRef](#)] [[PubMed](#)]
45. Blei, D.M.; Kucukelbir, A.; McAuliffe, J.D. Variational inference: A review for statisticians. *J. Am. Stat. Assoc.* **2017**, *112*, 859–877. [[CrossRef](#)]
46. Hoffman, M.D.; Blei, D.M.; Wang, C.; Paisley, J. Stochastic variational inference. *J. Mach. Learn. Res.* **2013**, *14*, 1303–1347.
47. Simon, H.U. An almost optimal PAC algorithm. In Proceedings of the Conference on Learning Theory, Paris, France, 3–6 July 2015.
48. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747.
49. Sermanet, P.; Chintala, S.; LeCun, Y. Convolutional neural networks applied to house numbers digit classification. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, 11–15 November 2012; IEEE: Piscataway, NJ, USA, 2012.
50. Huang, K.H. DeepAL: Deep Active Learning in Python. *arXiv* **2021**, arXiv:2111.15258.
51. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.