FALCUN: A Simple and Efficient Deep Active Learning Strategy

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

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

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

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

1 Introduction

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

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

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

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

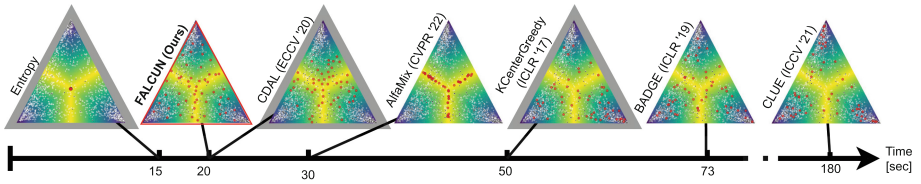


Fig. 1. Each simplex illustrates the probability space of a three-class subset of MNIST. The highest probabilities are in the corners (implied by darker colors). Small black and white dots are objects in $\mathcal{L}$ and $\mathcal{U}$, respectively. Red dots are instances selected by an AL method. FALCUN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

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

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

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

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

The main benefits of FALCUN are:

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

2 Related Work

AL techniques can be grouped into the following categories.

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

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

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

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

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

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

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

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

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

3 Methodology of FALCUN

3.1 Notation

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

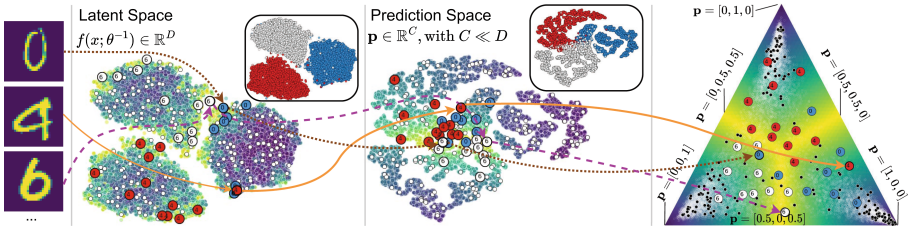


Fig. 2. FALCUN selects diverse and uncertain instances (colored circles) in the probability space (see 3-class simplex on the right). In the latent space on the left, they cover the most informative regions (yellow) while being highly diverse and stemming from different clusters. Red, white, blue imply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

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

3.2 Overview

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

3.3 Acquisition

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

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

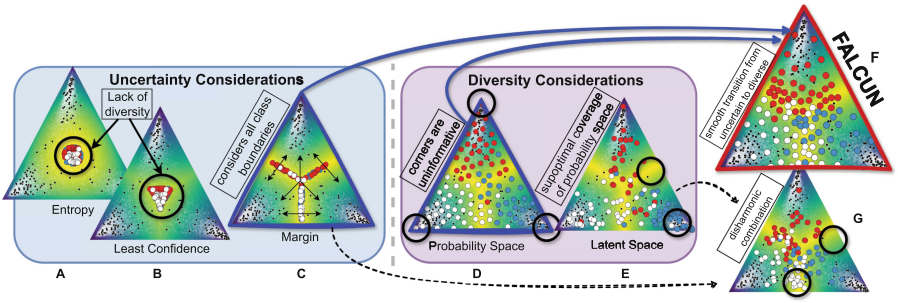


Fig. 3. Uncertainty Considerations (Left): In contrast to least confidence and entropy, the margin estimate focuses on the class boundaries between all class pairs, covering a more diverse spectrum. **Diversity Considerations** (Middle): Maximizing diversity in the probability space automatically covers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

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

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

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

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

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

Algorithm 1. Our AL Algorithm FALCUN

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

```
Train initial weights θ_0 on L by minimizing E_L[l_ce(f(x;θ), y)]
for r = 1, 2, ..., R do
    Initialize empty query set: Q = {}
    ∀x ∈ U : Compute class probabilities p(x)
    ∀x ∈ U : Initialize u(x) and d(x) with Equations (1) and (3)
    for q = 1, ..., Q do
        ∀x ∈ U : Calculate relevance score r(x) with Equation (5)
        Sample according to Equation (6)
        Q = Q ∪ x_q
        ∀x ∈ U : Update diversity values d(x) using Equation (4)
    end for
    Receive new labels from oracle for instances in Q
    L = L ∪ Q, U = U \ Q
    Train new model θ_r from scratch on L by minimizing E_L[l_ce(f(x;θ), y)]
end for
return Final parameters θ_R obtained in round R
```

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

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

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

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

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

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

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

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

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

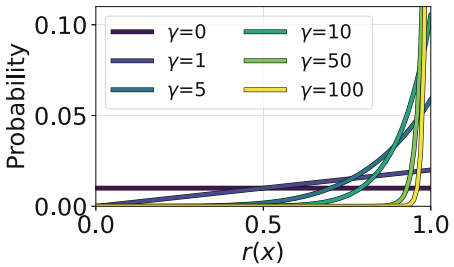


Fig. 4. Selection probability of an instance x for different γ values as a function of its relevance score $r(x)$.

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

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

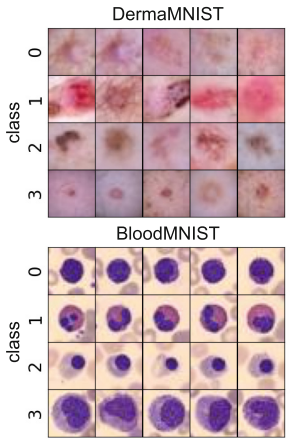


Fig. 5. Exemplary images of the two medical datasets.

4 Experiments

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

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

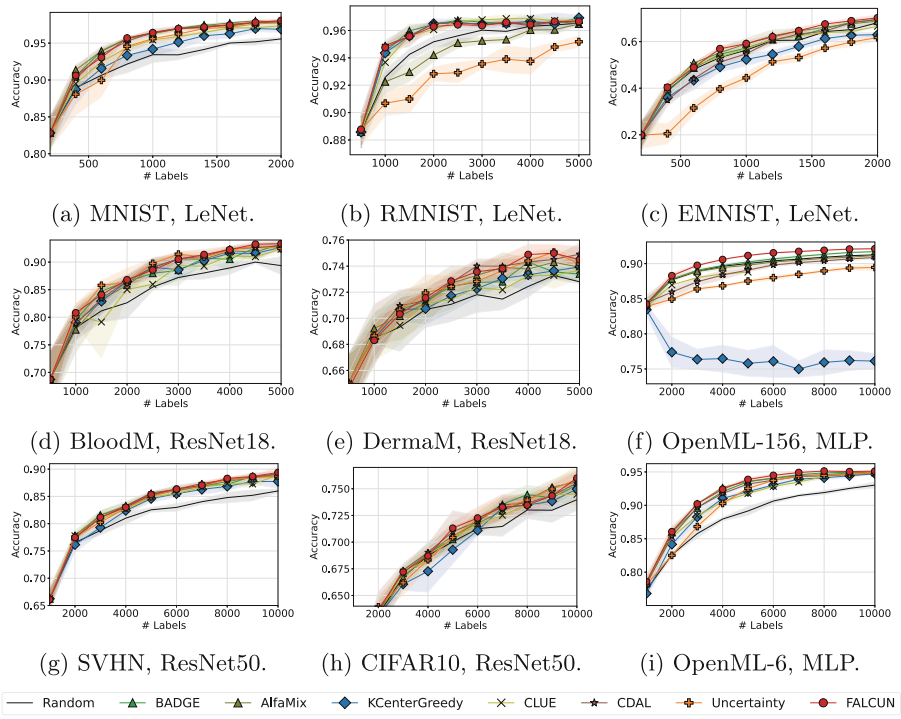


Fig. 6. Average test accuracy vs labeling budget for all active learning methods evaluated on greyscale (a, d), RGB (b, e) and tabular data (c, f).

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

4.1 Label Efficiency

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

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

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

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

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

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

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

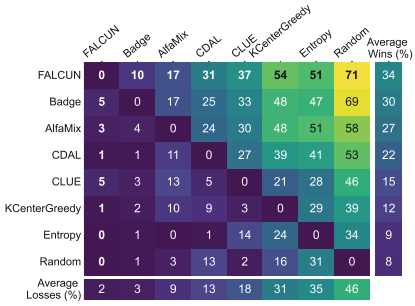


Fig. 7. Dueling matrix: The last column gives the percentage of wins of the respective method. The last row gives the percentage of losses.

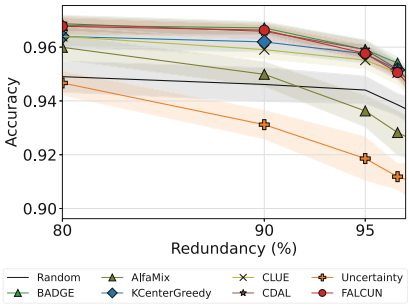


Fig. 8. Final average test accuracy for varying redundancy ratios.

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

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

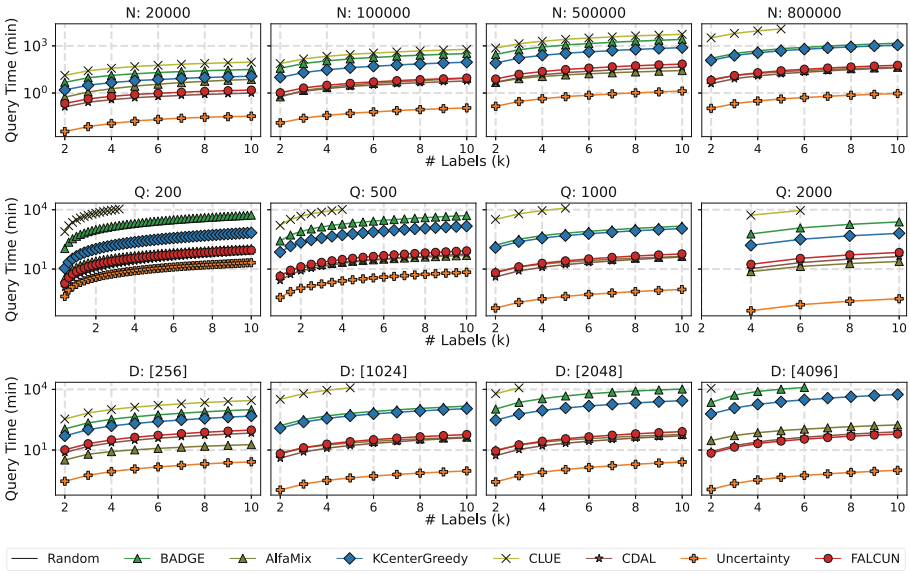


Fig. 9. Average cumulated acquisition times (y-axis) on a log-scale vs. annotated samples (x-axis) over varying unlabeled pool sizes N (first row), query sizes Q (second row), and dimensionality of the penultimate layer D (third row).

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

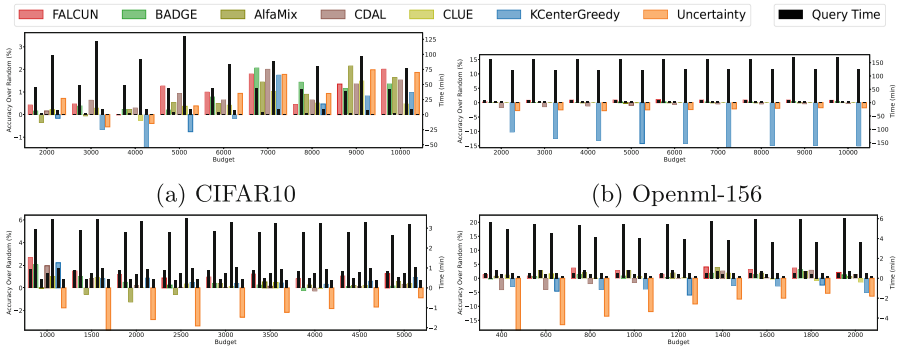
4.2 Query Time Efficiency

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

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

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

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



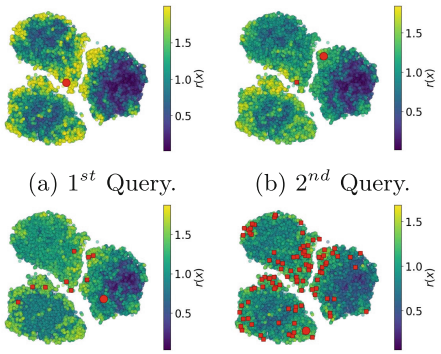
(a) CIFAR10 (b) Openml-156

(c) RMNIST (d) EMNIST

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

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

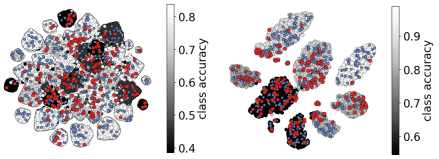
smaller time bars are better. FALCUN (red bars) has strong performance on all datasets and never has worse average accuracy than random sampling (i.e., values smaller than zero). CLUE and especially BADGE perform on par in some settings, but their query times are much higher, in some cases up to > 200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



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

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

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



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

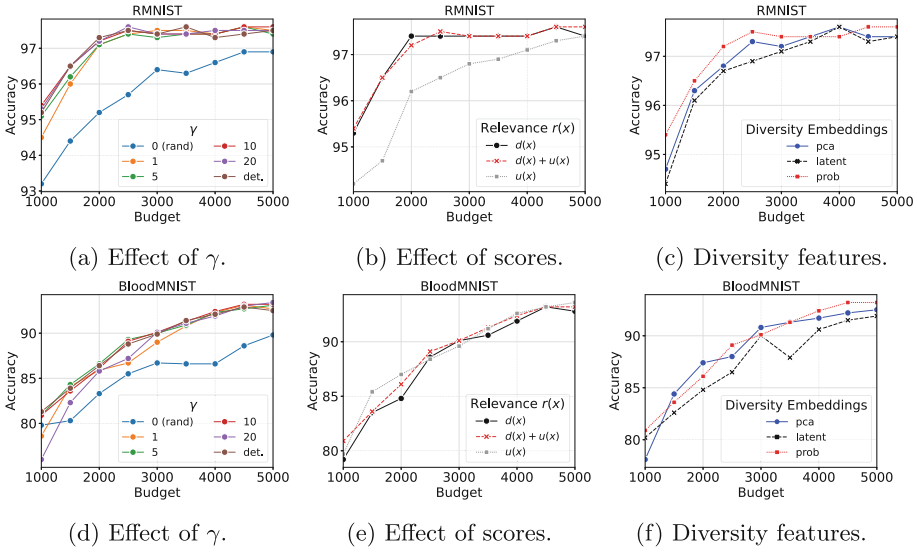
Fig. 12. Hue in the t-SNE visualizations indicates the predictive accuracy of the model on the respective class. Initially sampled objects are blue, samples chosen by FALCUN in the first acquisition round are red. FALCUN selects diverse instances favoring classes that are harder to distinguish by the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

4.3 Qualitative Evaluation

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

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

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



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

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

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

4.4 Ablation

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

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

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

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

5 Conclusion

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

References

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

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

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