General Reusability: Ensuring Long-Term Benefits of Deep Active Learning

Paul Hahn*, Denis Huseljic, Marek Herde and Bernhard Sick

University of Kassel, Intelligent Embedded Systems, Wilhelmshöher Allee 73, Kassel, 34121, Germany

Abstract

Deep active learning (DAL) provides a promising framework for reducing labeling costs for data-hungry deep neural networks. In DAL research, experiments are evaluated based on the performance of the model that is responsible for querying data, which can be interpreted as a model-centric evaluation. However, the data queried by DAL should be model agnostic so that new models can be effectively trained on it. An example can be derived from the rapid progress in deep learning, where new architectures are proposed in quick succession. Naturally, we want to employ these new and more effective models while still using the queried data from DAL. This results in the need to ensure high performance for both current and future models, which we refer to as reusability. In line with this topic, we propose general reusability, a novel, data-centric evaluation metric that measures the long-term value of queried data. Our experiments demonstrate that the current evaluation protocol in DAL research does not assess the reusability of data and that GR can fill this gap.

Keywords

Deep Learning, Deep Active Learning, Evaluation, Computer Vision, Reusability

1. Introduction

Deep neural networks (DNNs) are state-of-the-art for solving various tasks, such as image classification, object detection, or speech recognition, when trained on large datasets of labeled samples [1, 2, 3, 4]. However, obtaining labels remains expensive and time-consuming, requiring human experts [5]. Active learning offers a solution based on the idea that not all samples are equally valuable for maximizing model performance. Therefore, the cost of labeling can be reduced by intelligently labeling the most valuable samples. In an iterative process, a model is trained on available labeled samples and, subsequently, queries new samples to be labeled based on a query strategy [6]. Together, DNNs and active learning form the popular research field of deep active learning (DAL) [7, 8, 9, 10, 11, 12], which our study is centered around. Furthermore, we focus on image classification, a prominent task for DAL research.

In a typical DAL setting, the query model and query strategy are key components. While the *query model* provides information about the samples (e.g., feature representations), the *query strategy* determines which samples to query based on the information provided [6]. Investigating their influence is a major part of current DAL research [7, 8, 9, 10, 11, 12]. Considering recent advancements, this especially includes how the query model is trained [13, 14, 15, 16, 17], e.g., by using semi-supervised learning (Semi-SL) and self-supervised learning (Self-SL) techniques.

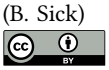
The standard evaluation protocol for DAL experiments measures the querying model's performance (QMP), e.g., its classification accuracy. Therefore, almost all studies compare different configurations of DAL components (e.g., Margin vs. BADGE as query strategy) based on the resulting QMPs [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]. We consider using the QMP a **model-centric** evaluation because it heavily centers around the querying model. However, in recent years, DNN architectures have made frequent advances in which their size and structure have changed, e.g., AlexNet [18], ResNet [19], EfficientNet [20], and ViT [21]. With this progress, replacing DNNs will be unavoidable, while the queried data (i.e., queried

IAL@ECML-PKDD'24: 8th Intl. Worksh. & Tutorial on Interactive Adaptive Learning, Sep. 9th, 2024, Vilnius, Lithuania

*Corresponding author.

paul.hahn@uni-kassel.de (P. Hahn); dhuseljic@uni-kassel.de (D. Huseljic); marek.herde@uni-kassel.de (M. Herde); bsick@uni-kassel.de (B. Sick)

0009-0005-1012-1281 (P. Hahn); 0000-0001-6207-1494 (D. Huseljic); 0000-0003-4908-122X (M. Herde); 0000-0001-9467-656X (B. Sick)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

CEUR-WS.org/Vol-3770/paper5.pdf

samples and their respective labels) will continue to be used. Consequently, the reusability of the queried data to future models must be guaranteed to ensure high model performances in the long term. *However, we question whether the QMP is a fitting indicator for this.*

Figure 1 illustrates an example on a 2D toy dataset two-moons [22]. We refer to Appendix A for details on the experimental setting. After conducting a DAL experiment, we present the decision boundaries of several models trained on the queried data. In addition to the query model, we use other models that slightly differ in their architecture (hidden layer size) and training hyperparameters (learning rate and weight decay). All samples are colored in their respective class color (red/blue), and labeled samples are enlarged. The query model reaches an almost perfect separation of both classes. In contrast, the other models perform significantly worse despite being able to perform well when querying their own data. *This suggests that a high QMP does not ensure the high performance of other models, as the data queried may only meet the specific requirements of the query model to achieve high performance.*

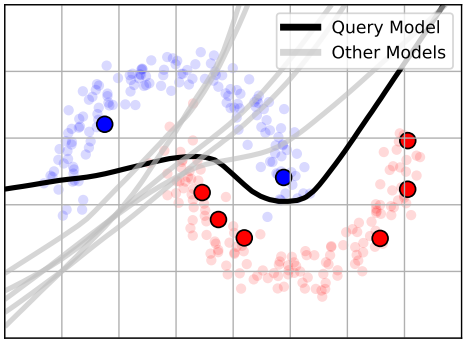


Figure 1: Different models trained on data queried by one of them.

To tackle this issue, we propose an alternative evaluation metric with a generalized version of *reusability*. Queried data is reusable for a model if it enables higher performance than training on randomly queried data. The work that is most similar to ours is by Tomanek and Morik [23], which only considers reusability between pairs of models apart from DNNs. We aim to extend their work towards DAL and generalize their notion of reusability to ensure high performance for future models. Hence, in this article, we define *general reusability* (GR) as the average reusability of queried data for a diverse set of models. While models can differ in many ways, we focus on different architectures and training methods as these have played a central role in recent DAL research [14, 15, 16, 17, 24]. Unlike the standard DAL evaluation protocol, using GR can be considered a **data-centric** evaluation as it marginalizes the role of a single model. Extensive experiments show that, in contrast to model-centric evaluation, our data-centric evaluation promotes DAL experiments that exhibit high reusability for future models. Furthermore, this new perspective reveals that the recent suggestions from research promoting Semi- and Self-SL [13, 14, 15, 16, 17] can negatively impact querying reusable data.

Contributions

- We propose general reusability, a novel, data-centric evaluation metric for DAL that measures the reusability of queried data for a diverse set of models.
- We perform an extensive set of DAL experiments by varying components, e.g., query strategies, model architectures, and training methods, and compare QMP and GR based on their evaluation.
- We discuss GR and QMP and their ability to indicate high reusability of queried data for future model performances.
- We provide the code of the conducted experiments for reproducibility at https://github.com/nhaH-luaP/general-reusability.

2. Related Work

Query strategies can mainly be divided into uncertainty-based and diversity-based strategies. While uncertainty-based strategies focus on samples the model considers "hard" to classify, diversity-based strategies aim to find samples representative of the underlying data distribution. *Margin* [10] is a popular uncertainty-based query strategy [25] that queries the samples with the largest difference between the two highest probable class probabilities. In contrast, *CoreSets* [7] is a diversity-based query strategy that works in the feature representation space. There, it queries the samples with the maximum distance to their closest labeled sample. In recent years, various works have tried to combine the strengths of different types of strategies [8, 9]. *BADGE* [8] combines uncertainty and diversity by using predicted labels to calculate gradient embeddings for each unlabeled sample. In this gradient embedding space, they use the k-means++ initialization scheme to query a diverse batch of samples that significantly changes the model's weights. While also focusing on diversity, *TypiClust* [9] replaces the notion of uncertainty with typicality. It queries samples from different clusters (diversity) in the feature representation space with high density (typicality). Lastly, as a baseline, we consider randomly querying samples without a strategy and refer to it as *Random*.

Criticism of DAL research was expressed in several papers throughout recent years. The uprise of Semi- and Self-SL methods lead to large-scale comparative studies investigating their use for DAL. Siméoni et al. [17] find that using Semi- and Self-SL in DAL experiments to train the query model significantly boosts the performance of any strategy, ranking it above all strategies in the supervised setting. Mittal et al. [15] confirm these results and come to similar conclusions when applying state-of-the-art data augmentation for query model training. Furthermore, they find that while the absolute performance of each strategy increases, the performance differences between strategies diminish, making it difficult to identify a clear winner. Munjal et al. [13] investigate the importance of correctly tuned hyperparameters and how optimal hyperparameter choices may change from cycle to cycle. Specifically, they optimize each model's learning rate and weight decay in each cycle based on an external validation set. In addition to findings similar to Siméoni et al. [17] and Mittal et al. [15], they find that the performance of the random baseline is commonly under-reported and becomes competitive with other strategies when evaluated in equal settings. Li et al. [14] conducted the largest comparative study covering 19 different query strategies and the impact of Semi-SL. In addition to previous findings, they propose one should perform Semi-SL as early as possible and seek more unlabeled data whenever possible as it benefits Semi-SL. Finally, Lüth et al. [16] present five key pitfalls commonly performed in the DAL literature and propose a more extensive evaluation protocol focusing on real-world conditions. The proposals are to conduct experiments with varying data distributions, starting budgets, and query sizes while optimizing model hyperparameters and using Semi- and Self-SL.

Reusability has not been discussed for DAL and only scarcely for active learning. Lewis and Catlett [26], motivated by the need to reduce computational complexity, used logistic regression to query samples later used for a decision tree classifier. They report high reusability of the data queried in the context of text classification. In contrast, Baldridge and Osborne [27] find that data queried from an active learning experiment using an uncertainty-based strategy can result in performance degradation when reused for training a different classifier. Tomanek et al. [28] query samples using query-by-committee with a committee consisting of maximum entropy classifiers. They report a high reusability for a subsequently trained conditional random field. In another work, Tomanek and Morik [23] examine the pairwise reusability of queried data between two models. They find that most pairs of models query reusable data for each other, i.e., they achieved better performance on the queried data by their opposite than on randomly queried data. Finally, Hu et al. [29] build on the work of Tomanek and Morik [23] and investigate reusability in the text classification setting. They find that SVMs query the most reusable data, while naive Bayes classifiers generally work best on data queried by other models. However, both Tomanek and Morik [23] and Hu et al. [29] report that the performance of a model trained on its queried data is usually an upper bound for training on data queried from others. In contrast to our work, these studies focus on models outside of deep learning (e.g., SVM, decision tree, Naive Bayes), tabular or natural language datasets, and examination of reusability between pairs of models.

3. General Reusability

Notation: The typical DAL cycle begins with a large pool of unlabeled samples $\mathcal{U}$ and a pool of labeled samples $\mathcal{L}$[1], where typically $|\mathcal{L}| \ll |\mathcal{U}|$. Then, it iteratively trains a model $m \in \mathcal{M}$ on $\mathcal{L}$ and queries new samples from $\mathcal{U}$ based on a query strategy $q \in \mathcal{Q}$, which are labeled and added to $\mathcal{L}$. This process repeats until a stopping criterion is met, e.g., labeling budget or number of cycles K. Due to the recent progress of Semi- and Self-SL [30, 31] and their resulting appliance in DAL [24], $\mathcal{U}$ may also be required for model training. We introduce $\mathcal{Q}$ as the set of all possible query strategies, e.g., BADGE [8], and $\mathcal{M}$ as the set of all models. For DNNs, $\mathcal{M}$ can further be specified by $\mathcal{T}$ as the set of all possible training methods, e.g., Semi-SL, and $\mathcal{A}$ as the set of all possible model architectures, e.g., ResNet-18 [19], resulting in $\mathcal{M} = \mathcal{A} \times \mathcal{T}$[2]. While DNNs can differ in many ways, we focus on model architectures and training methods for $\mathcal{M}$ as they play a central role in recent DAL research [14, 15, 16, 17, 24]. For the remainder of this work, we refer to $\mathcal{Q}$, $\mathcal{A}$, and $\mathcal{T}$ as *DAL components* and to a specific choice of DAL components, e.g., Margin $\in \mathcal{Q}$, ResNet-18 $\in \mathcal{A}$, and Semi-SL $\in \mathcal{T}$, as a *DAL configuration*.

Considering this setup, the query process depends on query strategy q and query model m. Therefore, a labeled pool resulting from a DAL configuration with query strategy q and model m can be identified with $\mathcal{L}_{q,m}$. Concerning *Random*, we denote the resulting labeled pool with $\mathcal{L}_{\text{Random}}$, as the model does not influence the random selection.

In DAL, two common ways exist to measure the performance of a model m for a given experiment [14, 15, 16, 17, 24]. The first way is to measure the model's performance once at the end of the experiment. Therefore, we define $\textbf{PERF}(\mathcal{L}, \mathcal{U}, m, K)$ as the performance of model m trained on the labeled pool $\mathcal{L}$ and potentially on the unlabeled pool $\mathcal{U}$ after the final K-th DAL cycle. The second way is to measure the model performance in each cycle, considering the whole learning process. Based on this, we define $\textbf{AUC}(\mathcal{L}, \mathcal{U}, m, K)$ as the area under the curve of the performance over K cycles. Note that we remove K and $\mathcal{U}$ from both measure denominations for the further course of the work for ease of notation. In addition, model performance is a placeholder for different performance metrics that depend on the task and setting. In our case, we define model performance as the classification accuracy in percent on a given test dataset.

Current DAL literature evaluates experiments with the QMP, i.e., $\textbf{PERF}(\mathcal{L}, m)$ or $\textbf{AUC}(\mathcal{L}, m)$, where the query model $m \in \mathcal{M}$ is responsible for providing information for querying and $\mathcal{L}$ the resulting labeled pool [8, 9, 7, 24, 17, 13, 14, 16, 15]. Based on the QMP, recommendations are made, e.g., Semi- and Self-SL benefit DAL [17, 13, 14, 16, 15]. Considering recent advancements in DNN architectures [18, 19, 20, 21] and training methods [30, 31], a DAL experiment should query data reusable for future DNNs. Queried data is reusable for a model if it leads to higher performance than randomly queried data. However, the QMP does not account for reusability as it strongly centers around the query model. Therefore, relying only on QMP could substantially decrease model performance once the original query model is exchanged. *As a result, we propose a shift from a* ***model-centric*** *to a* ***data-centric*** *evaluation by also considering the reusability of queried data for a diverse set of models.*

To formalize the idea of reusability for a diverse set of models, we build on the concept of reusability introduced by Tomanek and Morik [23]. For a pair of models (m, $\overline{m}$), they define the reusability of a labeled pool $\mathcal{L}_{q,m}$ for $\overline{m}$ as

$$\textbf{REU}(\mathcal{L}_{q,m}, \overline{m}) = \frac{\textbf{AUC}(\mathcal{L}_{q,m}, \overline{m}) - \textbf{AUC}(\mathcal{L}_{\text{Random}}, \overline{m})}{\textbf{AUC}(\mathcal{L}_{q,\overline{m}}, \overline{m}) - \textbf{AUC}(\mathcal{L}_{\text{Random}}, \overline{m})} - 1. \tag{1}$$

Intuitively, **REU** can be seen as the performance gain of a model $\overline{m}$ when trained on data queried by another model m relative to being trained on its own queried data. For example, a $\textbf{REU} < 0$ indicates that $\overline{m}$ performs better on its queried data, while a $\textbf{REU} > 0$ indicates that $\overline{m}$ performs better on data queried by m. Tomanek and Morik [23] investigate reusability between pairs of machine learning

[1]Throughout this work, we use "queried data", "pool of labeled samples", and "labeled pool" synonymous. "Labeled pool" is a well-known formal concept in pool-based active learning while "queried data" aids conceptual explanations.

[2]This serves as a simplified representation. Note that in practice there are pairs of models and training methods that cannot be combined.

models outside DAL, such as decision trees or SVMs. In contrast, we focus on DNNs and examine a more general notion of reusability concerning various other DNNs. Therefore, we propose to apply three major changes to the reusability defined in Eq. (1).

At first, we propose to remove the denominator. In our setting, as future models should reuse the queried data, we are not concerned with the performance they could have achieved when querying their own data but only with their performance gain on currently queried data over randomly queried data. This broadens the choice of models for the evaluation, as it allows using a model that has not performed a DAL experiment. Otherwise, this would require considerable computational effort, especially for DNNs, where training a model is expensive. Additionally, when using the denominator, it is not clear for a **REU** < -1 whether the performance on the self-queried data is worse than on the randomly queried data or on the data queried by the other model, which hinders interpretation.

Second, we exchange the **AUC** for **PERF** to focus our investigation on the queried data resulting from a DAL experiment, which reduces the computational complexity by a factor of K. However, whether the **AUC** may provide additional information that could benefit GR is unclear, so we consider it an option for future work. As a result, our version of reusability can then be defined as

$$\mathbf{REU}^*(\mathcal{L}_{q,m}, \overline{m}) = \mathbf{PERF}(\mathcal{L}_{q,m}, \overline{m}) - \mathbf{PERF}(\mathcal{L}_{\text{Random}}, \overline{m}).$$

Note that we removed the -1 as it was used to center the original metric around 0. Intuitively, we simplified the original reusability to the performance that a model gains when trained on a queried pool $\mathcal{L}_{q,m}$ compared to randomly queried data.

As a third and final step, we extend the metric to multiple models to get a more general view of reusability. Recall that our goal is to ensure the reusability of queried data for future models. However, we cannot measure reusability for future models as they are unknown. Therefore, we propose to approximate the reusability of queried data for future models by its average reusability for the set of all known models $\mathcal{M}$. We provide empirical evidence for this claim in Section 4.3. In other words, the key idea is that queried data, which is reusable for various known models, should also be reusable for any potential future model. In a theoretical sense, we define the **general reusability** (**GR**) of a queried pool $\mathcal{L}_{q,m}$ as

$$\mathbf{GR}(\mathcal{L}_{q,m}) = \int_{\overline{m} \in \mathcal{M}} \mathbf{REU}^*(\mathcal{L}_{q,m}, \overline{m}) \cdot p(\overline{m}) \mathrm{d}\overline{m},$$

where $p(\overline{m})$ is an unknown distribution weighing each model's reusability. The key idea behind p is that, depending on the application, the reusability of some models may be more meaningful in terms of general reusability than others. By marginalizing the role of a single model, we arrive at a measure that evaluates the general reusability of queried data. However, as the integral is infeasible to compute, we approximate it by using Monte Carlo integration over a finite number of models $\overline{\mathcal{M}} \subset \mathcal{M}$. By choosing $\overline{\mathcal{M}}$ to include frequently used models with equal weighting (i.e., $p(\overline{m}) = 1/|\overline{\mathcal{M}}|$), we approximate general reusability with

$$\mathbf{GR}(\mathcal{L}_{q,m}) \approx \frac{1}{|\overline{\mathcal{M}}|} \sum_{\overline{m} \in \overline{\mathcal{M}}} \mathbf{REU}^*(\mathcal{L}_{q,m}, \overline{m}). \tag{2}$$

Intuitively, a **GR** > 0 indicates that the queried data is reusable for other models. In contrast, a **GR** < 0 suggests that, on average, models perform better on randomly queried data than on $\mathcal{L}_{q,m}$.

4. Experiments

In this section, we compare GR to QMP in terms of how they evaluate DAL experiments and their ability to indicate the reusability of queried data for future models. Therefore, we begin by conducting extensive DAL experiments, in which we vary DAL components such as model architecture, training method, and query strategy and evaluate them according to the literature with the QMP. The goal is to reproduce some of the findings of recent DAL criticism, indicating the superiority of Semi- and Self-SL in DAL. Next, we compare GR and QMP from two different perspectives to assess their differences in evaluating the conducted DAL experiments. Finally, we train various models on the highest-ranking settings of each metric that have neither been used for querying nor evaluation (simulated future models). The goal is to investigate our metrics' ability to promote settings that query data reusable for future models.

4.1. Setting

All experiments are conducted on CIFAR-10 [32], a classic benchmark in DAL literature [8, 24, 7, 9, 11, 12]. It consists of 60,000 32x32 color images with 10 different classes. A split of 10,000 images serves as a test dataset to measure **PERF**, i.e., the classification accuracy in percent.

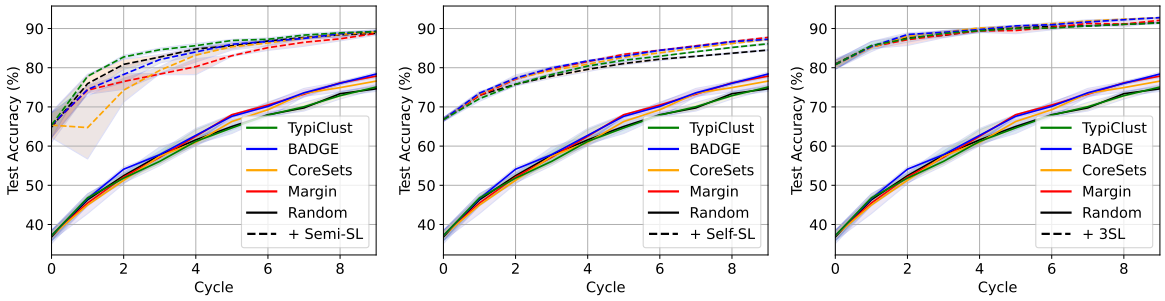
As our general query model architecture, we use residual neural networks (ResNets) [19]. We use varying depths to provide models of different complexity, namely $\mathcal{A} := \{$*ResNet-6, ResNet-10, ResNet-18*$\}$. As simulated future models, we use wide ResNets (WideResNets) [33], namely the *WideResNet-28-2* and the *WideResNet-28-10*. WideResNets were proposed after ResNets and, therefore, represent a realistic future model architecture in relation to ResNets.

For training methods, we use $\mathcal{T} := \{$*Base, Semi-SL, Self-SL, 3SL*$\}$. While these terms are general descriptions for training types, they are represented by a unique procedure within our experiments. *Base* describes standard supervised training using stochastic gradient descent and a cosine annealing learning rate scheduler. We train each model $m \in \overline{\mathcal{M}}$ for 200 epochs with a learning rate of 0.01, a weight decay of 0.0005, a batch size of 64, a momentum of 0.9 and nesterov set to true. For the two smaller architectures *ResNet-6* and *ResNet-10*, we reduce the weight decay to 0.00001. *Semi-SL* extends *Base* with a Semi-SL method. We use MixMatch [30], which applies MixUp [34] to both pairs of labeled and pairs of labeled and pseudo-labeled images. Pseudo-labels are based on the model's prediction rather than a ground truth label. *Self-SL* performs an auxiliary task and uses the resulting weights as initial values for the subsequent supervised training according to *Base*. We use SimCLR [31] as the auxiliary task, where a model learns matching differently augmented instances of the same image. In particular, the model's task is minimizing a contrastive loss function (i.e., maximizing similarity) between the resulting feature representations of the same original image while penalizing similarity to any other feature representation of a different image. Finally, *3SL* describes the combination of *Semi-* and *Self-SL*. We refer to our implementation for more details on the method-specific hyperparameters.

For each DAL experiment, we start with an initial labeled pool of 300 samples and query an additional 300 for 9 cycles for a total of 3000 labeled samples. We perform a cold start in each cycle by retraining the model and examine five different query strategies, namely $\mathcal{Q} := \{$*Random, Margin, CoreSets, Badge, TypiClust*$\}$, which have been described in more detail in Section 2. For QMP, we measure $\textbf{PERF}(\mathcal{L}, m)$ of the respective querying model m on the final labeled pool $\mathcal{L}$. For GR, we apply Eq. 2 with $\overline{\mathcal{M}} = \mathcal{A} \times \mathcal{T}$ as defined above. Simply put, we average the reusability of the queried data of an experiment over each model $m \in \overline{\mathcal{M}}$. In summary, we perform the described DAL procedure for each configuration in $\mathcal{Q} \times \mathcal{A} \times \mathcal{T}$ for three different seeds and average the results to reduce the impact of randomness.

4.2. DAL Experiments

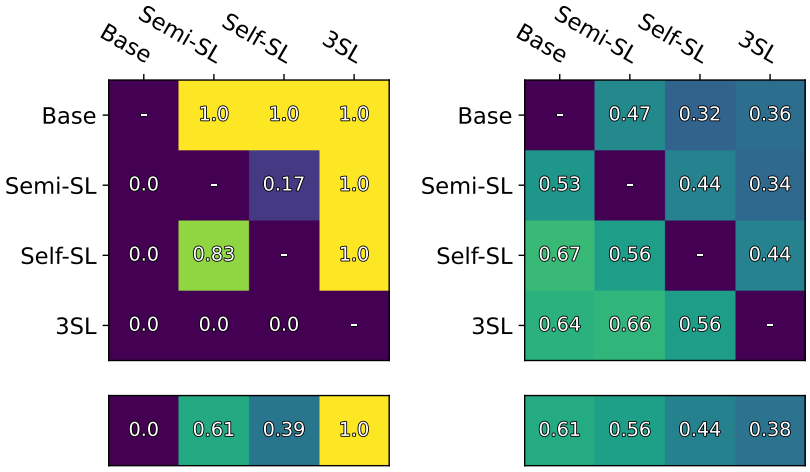
Figure 2 shows the QMP per cycle for a *ResNet-18* with varying query strategies and training methods. It demonstrates a significant improvement in QMP when using *Semi-SL*, *Self-SL*, or *3SL* compared to



(a) Semi-SL (b) Self-SL (c) 3SL

Figure 2: The effects of different training methods (dashed lines), i.e., *Semi-SL*, *Self-SL* and *3SL*, on the QMP (ResNet-18) over multiple cycles in DAL experiments compared to *Base* (solid lines).

Base. Although *Semi-SL* achieves a higher QMP than *Self-SL*, it has a higher standard deviation in earlier cycles, suggesting that it may not achieve good convergence with only a few labeled samples. *3SL* not only surpasses both *Semi-SL* and *Self-SL*, but shows higher QMP on the initial labeled pool than *Base* reaches on its final labeled pool. In addition, applying *Semi-SL* or *3SL* diminishes any differences in query strategies observed for *Base*. The remaining learning curves in Appendix B show that the best-performing model architecture is *ResNet-18*, although *ResNet-10* is competitive in combination with *3SL*. Evaluation of the conducted DAL experiments with QMP (i.e., a model-centric view) reproduces the findings of the literature [17, 14, 15, 13, 16] and strongly promotes the use of larger model architectures and *3SL* for any DAL experiment while differences between query strategies diminish.



(a) QMP (b) GR

Figure 3: Pairwise penalty matrices for different training methods.

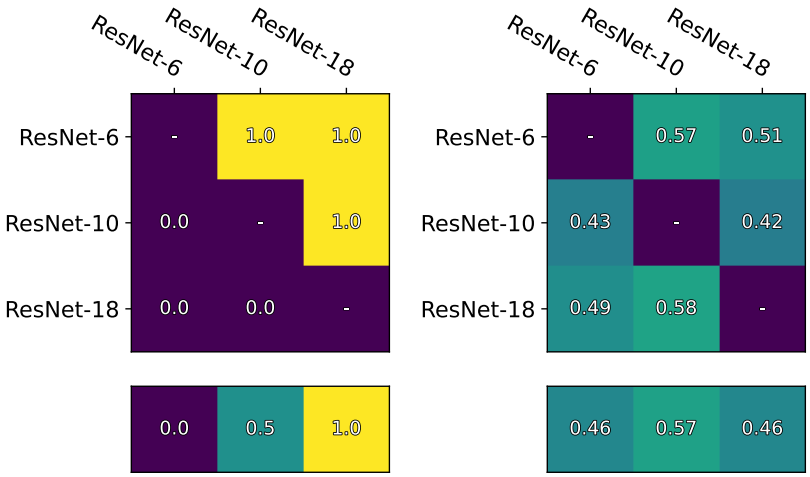
Next, we compare GR and QMP based on their evaluation of the conducted DAL experiments, focusing on different DAL components. Figure 3 shows pairwise penalty matrices concerning different training methods. In a pairwise penalty matrix, each cell contains the percentage of DAL experiments in which the column instance scored higher than the row instance concerning the respective evaluation metric. In addition, below each matrix is an average for each column instance. While each cell directly compares two instances, the averages below serve as a general ranking. Figure 3a shows that each training method scores higher in QMP than *Base* in 100% of cases, while *3SL* scores the highest. This aligns with the previous finding of using *3SL* to maximize QMP. In contrast, Fig. 3b shows a less unevenly distributed scoring, with *Base* scoring the highest and *3SL* the lowest. This indicates that while *3SL* significantly boosts QMP, it can negatively affect querying generally reusable data, and *Base* would be the preferable

Table 1
Top five DAL configurations determined by GR in red and by QMP in blue. Each configuration is listed with its scores in QMP and GR and the performance of *ResNet-18* trained on the resulting data using *3SL*.

	$\mathcal{A}$	$\mathcal{T}$	$\mathcal{Q}$	QMP	GR	3SL-PERF
1.	ResNet-18	Base	BADGE	78.39 ± 0.45	1.04 ± 1.69	91.79 ± 0.13
1.	ResNet-18	3SL	BADGE	92.78 ± 0.07	-3.79 ± 4.51	92.78 ± 0.07
2.	ResNet-10	Base	Margin	77.22 ± 0.42	0.95 ± 1.72	91.15 ± 0.29
2.	ResNet-18	3SL	CoreSets	92.78 ± 0.06	-6.89 ± 6.87	92.78 ± 0.06
3.	ResNet-6	Self-SL	BADGE	64.90 ± 0.17	0.69 ± 0.98	91.39 ± 0.30
3.	ResNet-10	3SL	CoreSets	92.32 ± 0.08	-5.41 ± 6.34	93.00 ± 0.05
4.	ResNet-6	Base	BADGE	60.46 ± 0.49	0.64 ± 0.77	91.60 ± 0.10
4.	ResNet-18	3SL	Margin	92.10 ± 0.41	-6.90 ± 5.95	92.10 ± 0.41
5.	ResNet-18	Base	Margin	77.85 ± 0.38	0.58 ± 1.45	91.06 ± 0.15
5.	ResNet-10	3SL	BADGE	91.94 ± 0.21	-3.51 ± 4.27	92.43 ± 0.12

choice for this.

Similarly, Fig. 4 shows pairwise penalty matrices concerning different model architectures. Figure 4a demonstrates a clear ranking of model architectures concerning QMP according to their size. The largest architecture, *ResNet-18*, scores highest, and the smallest architecture, *ResNet-6*, scores lowest. This, again, indicates that to maximize QMP, one should always consider the largest model architecture. In contrast, Fig. 4b shows similar scores for each architecture, while the medium-sized model ResNet-10 scores highest in GR. This indicates that a larger model architecture is not required to query reusable data. Compared to the training methods in Fig. 3, the differences in average scores are considerably less, indicating that training methods have a higher impact on GR than model architectures.



(a) QMP (b) GR

Figure 4: Pairwise penalty matrices for different model architectures.

Concerning query strategies, the respective penalty matrices are in Appendix C. One key difference is the assessment of *Random*, which scores higher for GR than for QMP, while the scores of other strategies mostly decrease. This indicates that querying in DAL is generally model-oriented. However, *BADGE* remains the highest-scoring query strategy, suggesting that it presents a good balance between maximizing the query model performance and ensuring reusability.

In addition to DAL components, we compare QMP and GR by investigating their highest-ranking DAL configurations. Table 1 displays the top five highest-ranking DAL configurations for both QMP and GR. For QMP, this follows our previous observations, choosing *3SL* in all cases and primarily larger model architectures. In contrast, GR promotes *Base* in 4 out of 5 cases and chooses smaller and larger model architectures equally. This reinforces our finding that training methods impact GR more

Table 2
Performance of potential future models when trained on the data resulting from the top five highest-ranking DAL configurations for both GR (red) and QMP (blue). We refer to Tab. 1 with the respective colors for the specific DAL configurations.

(a) *WideResNet-28-2.*

	Base	Semi-SL	Self-SL	3SL
1.	79.05	87.44	80.09	88.51
1.	69.77	85.88	74.23	88.86
2.	79.17	87.46	80.96	87.84
2.	62.78	83.98	72.34	89.12
3.	77.15	87.11	79.79	87.79
3.	67.15	89.12	75.05	90.29
4.	77.48	87.84	79.22	88.52
4.	63.46	84.67	71.77	86.91
5.	79.27	86.69	80.35	87.71
5.	70.67	87.10	74.12	88.47

(b) *WideResNet-28-10.*

	Base	Semi-SL	Self-SL	3SL
1.	77.19	86.89	81.20	89.58
1.	67.09	84.33	75.22	90.18
2.	77.50	86.29	82.10	88.99
2.	61.94	84.75	68.16	90.85
3.	76.39	85.92	80.97	89.70
3.	65.44	88.15	75.41	91.33
4.	75.74	87.33	80.38	90.57
4.	63.69	79.02	74.47	87.45
5.	78.16	85.25	82.21	88.86
5.	67.80	86.29	77.89	90.29

(c) *WideResNet-28-2 + Random.*

Base	Semi-SL	Self-SL	3SL
75.95	87.21	77.3	88.07

(d) *WideResNet-28-10 + Random.*

Base	Semi-SL	Self-SL	3SL
74.74	86.6	79.84	89.82

than model architectures. Concerning query strategies, *BADGE* is the most common in both metrics' highest-scoring configuration, indicating its superiority. Another observation is a negative correlation between QMP and GR. While the highest-ranking configurations for QMP score a QMP above 90%, the highest-ranking configurations for GR score a QMP below 80%. Similarly, the highest-ranking configurations for GR reach a positive value for GR while the highest-ranking configurations for QMP score a GR below -3.5. This indicates that a trade-off must be found between maximizing QMP and GR. To investigate whether the highest-ranking settings for GR and QMP can reach a similar performance when trained with the best-performing architecture and training method, we additionally measure the performance of a *ResNet-18* trained on each resulting queried data with *3SL*. Column *3SL-PERF* in Tab. 1 shows comparatively small differences in performance between high-ranking QMP and high-ranking GR configurations. This indicates that maximizing GR does not result in a significant loss of QMP.

Summarized, QMP and GR paint completely different pictures when evaluating DAL experiments. While QMP favors maximizing performance through larger model architectures and complex training methods, GR prefers simpler models and standard training. Furthermore, we showed that optimizing for GR does not result in low QMP when training a larger architecture with *3SL* on the queried data resulting from a DAL experiment.

4.3. Future Model Performance Prediction

So far, we established that QMP and GR differ significantly in their evaluation of DAL experiments. Now, we look at the potential benefits GR can bring when faced with an exchange of models. Thus, we investigate the research question: *How well do potential future models perform on data queried by the highest-ranking configurations in DAL components concerning each QMP and GR?* To investigate this, we train two larger model architectures, *WideResNet-28-2* and *WideResNet-28-10*, with each training method in $\mathcal{T}$ on the resulting data from the configurations listed in Tab. 1 and list their performance in Tab.2. In addition, we provide their performance when trained on randomly queried data below each table.

Concerning *Base*, *Semi-SL*, and *Self-SL*, most future models perform significantly better on the highest-ranking configurations of GR than on those of QMP. In addition, all GR configurations perform similarly or better to the random baselines provided in Tab. 2c and Tab. 2d while most QMP configurations perform similarly or worse. Concerning *3SL*, both metrics promote configurations that lead to high future model performance. This could result from the fact that the future and query models in the highest

ranking QMP configurations were trained with the same method (*3SL*), which may favor the reusability of the queried data. However, query and future models may differ significantly in their architecture and training method, which could lead to worse future model performances on high-ranking QMP configurations even in the *3SL* setting. In addition, future models do not always have to be larger and more powerful than the query model, e.g., if a large model queries data in development for a smaller model operating on an end device.

In summary, the results show that data scoring high in GR consistently enables high performance for future models, but this is not guaranteed when only QMP is considered. In contrast, the QMP seems insufficient in tracing the reusability of data and should primarily be used to make statements about the query model.

4.4. Takeaways

Our experiments compared QMP and GR on a large set of DAL experiments. We found that they substantially differ in their preferences for training methods and model architectures. While QMP prefers larger model architectures and more advanced training methods, GR ranks different model architectures similarly but prefers the baseline training method. This means that contrary to current literature, evaluation with GR does not indicate that using Semi- and Self-SL benefits the querying process. Considering the performance of potential future models, QMP fails to consistently promote configurations that are reusable for future models, resulting in performances worse than on randomly queried data. In contrast, future models perform similarly or often better when using data queried by configurations with a high GR than on randomly queried data. This shows that the GR of queried data should be evaluated to ensure consistently high-performing models in a setup where DAL is applied.

5. Conclusion and Future Work

In this work, we motivated the perspective of reusability for DAL in the face of recent progress and probable upcoming model exchanges. Furthermore, we proposed a novel evaluation metric, GR, that indicates the reusability of queried data for future models by measuring their reusability for a diverse set of known models. Our experiments show that the QMP, commonly used by the literature, is insufficient in measuring the reusability of queried data and that GR can fill this gap. We believe that the inclusion of GR in the DAL evaluation is required to ensure high model performance in the long term.

As this work pioneers the adaptation of reusability to DAL, some challenges remain to overcome. First, it is unclear which and how many models to consider for $\overline{\mathcal{M}}$ and how to weigh each model's reusability. Depending on the application, the selection of models and their weights may differ. Another point is that similar experiments should be conducted with various datasets and settings to investigate the impact of different DAL components on GR on a more general level. Furthermore, investigations concerning ensembles or query-by-committee seem promising, as multiple models determine the querying process and may, therefore, query more generally reusable data. Finally, combining GR with a hyperparameter-optimized evaluation [35] could further improve data-centricity, as both the model and its hyperparameters are marginalized. However, optimizing the hyperparameters of multiple models for each evaluation increases the computational complexity by a large margin.

References

[1] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012.

[2] S. Srivastava, G. Sharma, Omnivec: Learning robust representations with cross modal sharing, in: Winter Conference on Applications of Computer Vision, 2024.

[3] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, Y. Li, Maxvit: Multi-axis vision transformer, in: European Conference on Computer Vision, 2022.

[4] A. Baevski, Y. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: A framework for self-supervised learning of speech representations, Advances in Neural Information Processing Systems 33 (2020) 12449–12460.

[5] M. J. Willemink, W. A. Koszek, C. Hardell, J. Wu, D. Fleischmann, H. Harvey, L. R. Folio, R. M. Summers, D. L. Rubin, M. P. Lungren, Preparing medical imaging data for machine learning, Radiology 295 (2020) 4–15.

[6] R. M. Monarch, Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI, Simon and Schuster, 2021.

[7] O. Sener, S. Savarese, Active learning for convolutional neural networks: A core-set approach, in: International Conference on Learning Representations, 2018.

[8] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, A. Agarwal, Deep batch active learning by diverse, uncertain gradient lower bounds, in: International Conference on Learning Representations, 2020.

[9] G. Hacohen, A. Dekel, D. Weinshall, Active learning on a budget: Opposite strategies suit high and low budgets, in: International Conference on Machine Learning, 2022.

[10] D. D. Lewis, J. Catlett, Heterogeneous uncertainty sampling for supervised learning, in: Machine Learning Proceedings, 1994.

[11] J. T. Ash, S. Goel, A. Krishnamurthy, S. Kakade, Gone fishing: Neural active learning with fisher embeddings, in: Advances in Neural Information Processing Systems, 2021.

[12] W. Tan, L. Du, W. Buntine, Bayesian estimate of mean proper scores for diversity-enhanced active learning, Transactions on Pattern Analysis and Machine Intelligence 46 (2024) 3463–3479.

[13] P. Munjal, N. Hayat, M. Hayat, J. Sourati, S. Khan, Towards robust and reproducible active learning using neural networks, in: Conference on Computer Vision and Pattern Recognition, 2022.

[14] Y. Li, M. Chen, Y. Liu, D. He, Q. Xu, An empirical study on the efficacy of deep active learning for image classification, arXiv preprint arXiv:2212.03088 (2022).

[15] S. Mittal, M. Tatarchenko, Ö. Çiçek, T. Brox, Parting with illusions about deep active learning, arXiv preprint arXiv:1912.05361 (2019).

[16] C. T. Lüth, T. J. Bungert, L. Klein, P. F. Jaeger, Navigating the pitfalls of active learning evaluation: A systematic framework for meaningful performance assessment, in: Advances in Neural Information Processing Systems, 2023.

[17] O. Siméoni, M. Budnik, Y. Avrithis, G. Gravier, Rethinking deep active learning: Using unlabeled data at model training, in: International Conference on Pattern Recognition, 2021.

[18] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012.

[19] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Conference on Computer Vision and Pattern Recognition, 2016.

[20] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, 2019.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, 2020.

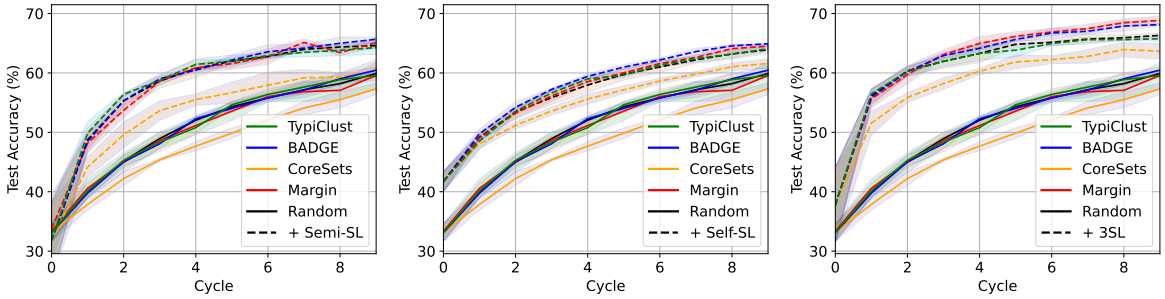
[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–

2830.
[23] K. Tomanek, K. Morik, Inspecting sample reusability for active learning, in: Artificial Intelligence and Statistics Conference, 2011.
[24] S. Song, D. Berthelot, A. Rostamizadeh, Combining mixmatch and active learning for better accuracy with fewer labels, arXiv preprint arXiv:1912.00594 (2019).
[25] D. Bahri, H. Jiang, T. Schuster, A. Rostamizadeh, Is margin all you need? an extensive empirical study of active learning on tabular data, arXiv preprint arXiv:2210.03822 (2022).
[26] D. D. Lewis, J. Catlett, Heterogeneous uncertainty sampling for supervised learning, in: Machine Learning Proceedings, 1994, pp. 148–156.
[27] J. Baldridge, M. Osborne, Active learning and the total cost of annotation, in: Conference on Empirical Methods in Natural Language Processing, 2004.
[28] K. Tomanek, J. Wermter, U. Hahn, An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data, in: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 2007.
[29] R. Hu, B. Mac Namee, S. J. Delany, Active learning for text classification with reusability, Expert Systems with Applications 45 (2016) 438–449.
[30] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, C. A. Raffel, Mixmatch: A holistic approach to semi-supervised learning, in: Advances in Neural Information Processing Systems, 2019.
[31] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, 2020.
[32] A. Krizhevsky, Learning multiple layers of features from tiny images, Master's thesis, University of Toronto, 2009.
[33] S. Zagoruyko, N. Komodakis, Wide residual networks, in: British Machine Vision Conference, 2016.
[34] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: International Conference on Learning Representations, 2018.
[35] D. Huseljic, M. Herde, P. Hahn, B. Sick, Role of hyperparameters in deep active learning, in: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, 2023.

A. 2D Toy Example

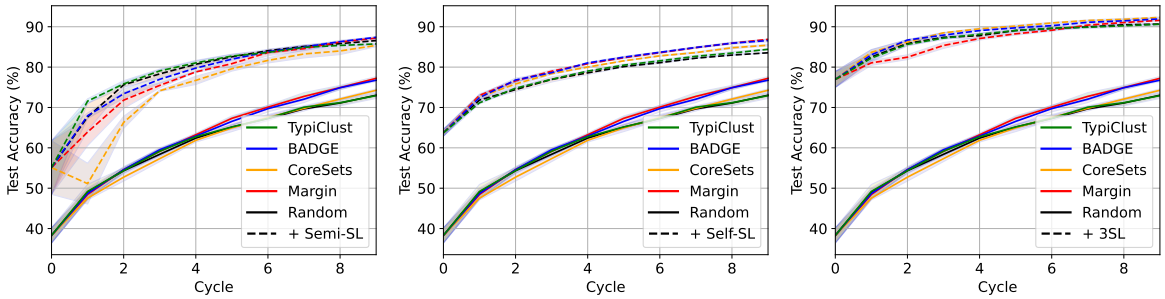
This section details the DAL settings for the graphical abstract in Fig. 1. We used various multi-layered perceptrons with one hidden layer that varied in the learning rate, weight decay, and hidden layer dimension. The query model used a learning rate of 0.1, while the other models used a learning rate of 0.01. For the query model, we chose a weight decay of 0.0005 and a hidden layer size of 256 while we randomly sampled the weight decay from $[0.00001, 0.01]$ and the hidden dimension from $[1, 512]$ for the other models. We randomly queried 4 labels for the initial labeled pool and queried another sample for 4 cycles using Margin [10]. For each cycle, we trained the query model from scratch using SGD as the optimizer with momentum set to 0.9 for 40 epochs and reduced the learning rate using a cosine annealing learning rate scheduler. Similarly, we trained the other models on the resulting labeled data using the same setting as the querying model.

B. DAL Learning Curves



(a) Semi-SL (b) Self-SL (c) 3SL

Figure 5: The effects of different training methods, i.e. *Semi-SL*, *Self-SL* and *3SL*, on model performance (ResNet-6) in DAL experiments compared to *Base* training.



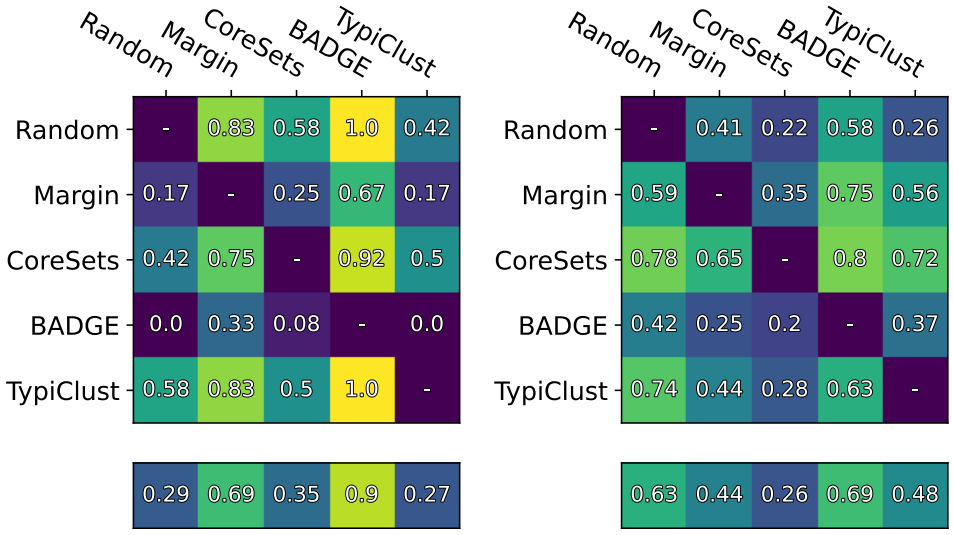
(a) Semi-SL (b) Self-SL (c) 3SL

Figure 6: The effects of different training methods, i.e. *Semi-SL*, *Self-SL* and *3SL*, on model performance (ResNet-10) in DAL experiments compared to *Base* training.

In this section, we display all learning curves concerning ResNet-6 (Fig. 5) and ResNet-10 (Fig. 6) similar to Fig. 2. While there are, again, performance improvements when using *Semi-SL*, *Self-SL*, and *3SL*, the absolute performances are lower than for ResNet-18.

C. Pairwise Penalty Matrices for Query Strategies

Figure 7 shows pairwise penalty matrices for query strategies according to Section 4.2. Considering



(a) QMP (b) GR

Figure 7: Pairwise penalty matrices for different query strategies.

QMP, *BADGE*, followed by *Margin*, performs well, while other strategies perform worse. In contrast, when evaluating with GR, differences between strategies diminish while *Random* scores substantially higher. Nevertheless, *BADGE* remains the highest-scoring query strategy, indicating a good trade-off in query model performance and reusability of queried data.