SIMILAR: Submodular Information Measures Based Active Learning In Realistic Scenarios

Suraj Kothawade
University of Texas at Dallas
suraj.kothawade@utdallas.edu

Nathan Beck
University of Texas at Dallas
nathan.beck@utdallas.edu

Krishnateja Killamsetty
University of Texas at Dallas
krishnateja.killamsetty@utdallas.edu

Rishabh Iyer
University of Texas at Dallas
rishabh.iyer@utdallas.edu

Abstract

Active learning has proven to be useful for minimizing labeling costs by selecting the most informative samples. However, existing active learning methods do not work well in realistic scenarios such as imbalance or rare classes, out-of-distribution data in the unlabeled set, and redundancy. In this work, we propose SIMILAR (**S**ubmodular **I**nformation **M**easures based act**I**ve **L**e**AR**ning), a unified active learning framework using recently proposed submodular information measures (SIM) as acquisition functions. We argue that SIMILAR not only works in standard active learning but also easily extends to the realistic settings considered above and acts as a *one-stop* solution for active learning that is scalable to large real-world datasets. Empirically, we show that SIMILAR significantly outperforms existing active learning algorithms by as much as $\approx 5\% - 18\%$ in the case of rare classes and $\approx 5\% - 10\%$ in the case of out-of-distribution data on several image classification tasks like CIFAR-10, MNIST, and ImageNet.

1 Introduction

Deep neural networks (DNNs) have had a lot of success in a wide variety of domains. However, they require large labeled datasets which are often taxing, time-consuming, and expensive to obtain. Active learning (AL) [12, 13, 38, 3, 9] is a promising approach to solve this problem. It aims to select the most informative data points from an unlabeled dataset to be labeled in an adaptive manner with a human in the loop. The goal of AL is to achieve maximum accuracy of the model while minimizing the number of data points required to be labeled.

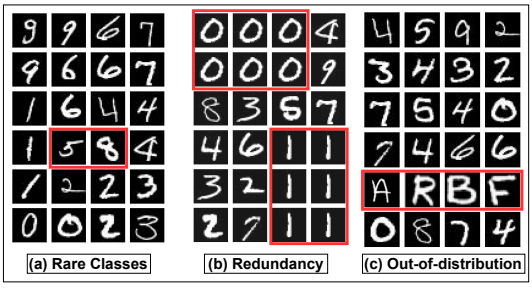

Figure 1: Motivating scenarios for realistic active learning: (a) rare classes: digits 5 and 8 are rare; (b) redundancy: digits 0 and 1 are redundant; (c) out-of-distribution (OOD): letters A, R, B, F in digit classification.

Current AL methods have been tested in relatively simple, clean, and balanced datasets. However, real-world datasets are not clean and have a number of characteristics that makes learning from them challenging [10, 45, 46, 37, 1, 8]. Firstly, these real-world datasets are *imbalanced*, and some classes are *very rare* (e.g., Fig 1(a)). Examples of this imbalance are medical imaging domains where the *cancerous* images are rare. Secondly, real-world data has a lot of *redundancy* (e.g., Fig 1(b)). This redundancy is more prominent in datasets that are created by sampling

Preprint.

frames from videos (e.g., footage from a car driving on a freeway or surveillance camera footage). Thirdly, it is common to have *out-of-distribution* (OOD) (e.g., Fig 1(c)) data, where some part of the unlabeled data is not of concern to the task at hand. Given the amount of unlabeled data, it is not realistic to assume that these datasets can be cleaned manually; hence, it is the need of the hour to have active learning methods that are robust to such scenarios. We show that current AL approaches (including the state-of-the-art approach BADGE [3]) do not work well in the presence of the dataset biases described above. In this work, we address the following question: *Can a machine learning model be trained using a single unified active learning framework that works for a broad spectrum of realistic scenarios?* As a solution, we propose SIMILAR[1], a unified active learning framework which enables active learning for many realistic scenarios like rare classes, out-of-distribution (OOD) data, and redundancy.

1.1 Related Work

Active learning has enabled efficient training of complex deep neural networks by decreasing labeling costs. The most commonly used approach is to select the most uncertain items. Examples of uncertainty strategies include ENTROPY [40], LEAST CONFIDENCE [43], and MARGIN [36]. One challenge of this approach is that all the samples within a batch can be potentially similar even though they are uncertain. To overcome this problem in batch active learning, many recent works have attempted to select diverse yet informative data points. Wei et al. [44] were some of the first to propose a simple approach: Filter a set of points using uncertainty sampling and then select a diverse subset from the filtered set. Sener and Savarese [39] propose CORESET, which forms core-sets using greedy k-center clustering while maintaining the geometric arrangement. BADGE [3], another recent approach, proposes to select data points corresponding to high-magnitude, diverse hypothesized gradients by using K-MEANS++ [2] initialization to distance from previously selected data points in the batch. Though BADGE [3] is currently the state-of-the-art, it fails to ensure diversity across AL selection rounds and does not perform as well when there is a lot of redundancy. Sinha et al. [41] used a variational autoencoder (VAE) [24] to learn a feature space and an adversarial network [31] to distinguish between labeled and unlabeled data points. However, their approach is computationally expensive and requires extensive hyperparameter tuning. Similarly, BATCHBALD [25] does not scale to larger batch sizes since their method would need a large number of Monte Carlo dropout samples to obtain a significant mutual information. Such limitations reduce the scope of applying these methods to realistic settings.

Closely related to our work are two recently proposed works. The first is GLISTER-ACTIVE [23], which formulates the AL acquisition function by maximizing the log-likelihood on a held-out validation set. This validation set could consist of examples from the rare classes or in-distribution examples. The second approach is the work of Gudovskiy et al. [15], who study AL for biased datasets using a self-supervised FISHER kernel and pseudo-label estimators. They address this problem by explicitly minimizing the KL divergence between training and validation sets via maximizing the FISHER kernel. Although their method shows promising results, they make multiple unrealistic assumptions: a) They use a *large labeled validation set*, and b) they use feature representations from a model pretrained using unsupervised learning on a *balanced* unlabeled dataset. In this work, we compare against both GLISTER-ACTIVE [23] and FISHER [15] approaches in the more realistic setting of a small held-out validation set (smaller than the seed labeled set) and an imbalanced unlabeled set. Another work proposed a discrete optimization method for k-NN-type algorithms in the domain shift setting [6]. However, their approach is limited to k-NNs.

This work utilizes submodular information measures (SIM) by [19] and their extensions by [22]. SIMs encompass submodular conditional mutual information (SCMI), which can then be used to derive submodular mutual information (SMI); submodular conditional gain (SCG); and submodular functions (SF). We discuss these functions in detail in Sec. 2. [22] also studies these functions on the closely related problem of targeted data selection.

Both not implemented

1.2 Our Contributions

The following are our main contributions: **1)** Given the limitations of existing approaches in handling active learning in the real world, we propose SIMILAR (Sec. 3), a unified active learning framework that can serve as a comprehensive solution to multiple realistic scenarios. **2)** We treat SIM as a common umbrella for realistic active learning and study the effect of different function instantiations offered under SIM for various realistic scenarios. **3)** SIMILAR not only handles standard active learning but also extends to a wide range of settings which appear in the real world such as rare classes, out-of-distribution (OOD) data, and datasets with a lot of redundancy. Finally, **4)** we

[1] **S**ubmodular **I**nformation **M**easures based act**I**ve **L**e**AR**ning

empirically demonstrate the effectiveness of SMI-based measures for image classification (Sec. 4) in a number of realistic data settings including imbalanced, out-of-distribution, and redundant data. Specifically, in the case of imbalanced and OOD data, we show that SIMILAR achieves improvements of more than 5 to 10% on several image classification datasets.

2 Background

In this section, we enumerate the different submodular functions that are covered under SIM and the relationships between them.

Submodular Functions. We let $\mathcal{U}$ denote the *unlabeled* set of n data points $\mathcal{U} = \{1, 2, 3, ..., n\}$ and a set function $f : 2^{\mathcal{U}} \rightarrow \mathbb{R}$. Formally, a function f is submodular [14] if for $x \in \mathcal{U}$, $f(\mathcal{A} \cup x) - f(\mathcal{A}) \geq f(\mathcal{B} \cup x) - f(\mathcal{B}), \forall \mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{U}$ and $x \notin \mathcal{B}$. For a set $\mathcal{A} \subseteq \mathcal{U}$, $f(\mathcal{A})$ provides a real-valued score for $\mathcal{A}$. In the context of batch active learning, this is the score of an acquisition function f on batch $\mathcal{A}$. Submodularity is particularly appealing because it naturally occurs in real world applications [42, 4, 5, 20] and also admits a constant factor $1 - \frac{1}{e}$ [33] for cardinality constraint maximization. Additionally, variants of the greedy algorithm maximize a submodular function in *near-linear time* [32].

Submodular Mutual Information (SMI). Given sets $\mathcal{A}, \mathcal{Q} \subseteq \mathcal{U}$, the SMI [16, 19] is defined as $I_f(\mathcal{A}; \mathcal{Q}) = f(\mathcal{A}) + f(\mathcal{Q}) - f(\mathcal{A} \cup \mathcal{Q})$. Intuitively, *SMI models the similarity between $\mathcal{Q}$ and $\mathcal{A}$*, and maximizing SMI will select points *similar* to $\mathcal{Q}$ while being diverse. $\mathcal{Q}$ here is the query set.

Submodular Conditional Gain (SCG). Given sets $\mathcal{A}, \mathcal{P} \subseteq \mathcal{U}$, the SCG $f(\mathcal{A}|\mathcal{P})$ is the gain in function value by adding $\mathcal{A}$ to $\mathcal{P}$. Thus, $f(\mathcal{A}|\mathcal{P}) = f(\mathcal{A} \cup \mathcal{P}) - f(\mathcal{P})$ [19]. Intuitively, SCG models how different $\mathcal{A}$ is from $\mathcal{P}$, and maximizing SCG functions will select data points *not similar to the points in* $\mathcal{P}$ while being diverse. We refer to $\mathcal{P}$ as the conditioning set.

Submodular Conditional Mutual Information (SCMI). Given sets $\mathcal{A}, \mathcal{Q}, \mathcal{P} \subseteq \mathcal{U}$, the SCMI is defined as $I_f(\mathcal{A}; \mathcal{Q}|\mathcal{P}) = f(\mathcal{A} \cup \mathcal{P}) + f(\mathcal{Q} \cup \mathcal{P}) - f(\mathcal{A} \cup \mathcal{Q} \cup \mathcal{P}) - f(\mathcal{P})$. Intuitively, SCMI *jointly models the similarity between $\mathcal{A}$ and $\mathcal{Q}$ and their dissimilarity with $\mathcal{P}$.*

Relationship between SIM The relationship between the above measures is the key component that unifies our AL framework [19, 22]. The unification comes from the rich modeling capacity of SCMI: $I_f(\mathcal{A}; \mathcal{Q}|\mathcal{P})$ where $\mathcal{Q}, \mathcal{P} \subseteq \mathcal{U}$. This facilitates a *single* acquisition function that can be applied to *multiple* scenarios. Concretely, the submodular function f can be obtained by setting $\mathcal{Q} \leftarrow \mathcal{U}$ and $\mathcal{P} \leftarrow \emptyset$. Next, the SMI can be obtained by setting $\mathcal{Q} \leftarrow \mathcal{Q}$ and $\mathcal{P} \leftarrow \emptyset$, while we obtain SCG by setting $\mathcal{Q} \leftarrow \emptyset$, $\mathcal{P} \leftarrow \mathcal{P}$. We summarize the relationships between SIM in Tab. 1.

Function	Setting	Realistic Scenario
Submodular	$\mathcal{Q} \leftarrow \mathcal{U}, \mathcal{P} \leftarrow \emptyset$	Standard AL
SMI	$\mathcal{Q} \leftarrow \mathcal{Q}, \mathcal{P} \leftarrow \emptyset$	Imbalance, OOD
SCG	$\mathcal{Q} \leftarrow \emptyset, \mathcal{P} \leftarrow \mathcal{P}$	Redundancy
SCMI	$\mathcal{Q} \leftarrow \mathcal{Q}, \mathcal{P} \leftarrow \mathcal{P}$	OOD

Table 1: Relationship between SIM and their applications to realistic scenarios by choices of $\mathcal{Q}$ and $\mathcal{P}$.

Instantiations of SIM. The formulations for Facility Location (FL), Graph Cut (GC) and Log Determinant (LOGDET) are as in [19, 22] and we adapt them as acquisition functions for batch active learning. We use two variants for FL: FLQMI, which models pairwise similarities of *only the query set* $\mathcal{Q}$ to the unlabeled dataset, and FLVMI, which additionally considers the pairwise similarities within the unlabeled dataset $\mathcal{U}$. The SCG and SCMI expressions corresponding to FL are referred as FLCG and FLCMI, respectively (see row 1 in Tab. 2a and 2b). For LOGDET, we refer to the SMI, SCG and SCMI expressions as LOGDETMI, LOGDETCG and LOGDETCMI, respectively (see row 5 in Tab. 2a and row 2 in Tab. 2b). Similarly, the SMI and SCG expressions are respectively referred to as GCMI and GCCG for GC (see row 3 in Tab. 2a and 2b). For notation in Tab. 2, the pairwise similarity matrix S between items in sets $\mathcal{A}$ and $\mathcal{B}$ is denoted as $S_{\mathcal{A},\mathcal{B}}$. Also, we denote S_{ij} as the (i, j) entry of S.

3 SIMILAR: Our Unified Active Learning Framework

In this section, we propose a unified active learning framework SIMILAR, which uses SIMs to address the limitations of the current work (see Sec. 1.1). We show that SIMILAR can be effectively applied to a broad range of realistic scenarios and thus acts as *one-stop* solution for AL.

The basic idea behind our framework is to exploit the relationship between the SIMs (Tab. 1) such that it can be applied to any real-world dataset. Particularly, we use the formulation of SCMI and appropriately choose a query set $\mathcal{Q}$ and/or a conditioning set $\mathcal{P}$ depending on the scenario at

Essentiell:
-A ist kleiner als B
-x ist nicht in A oder B
-f wird mit A und x größer, als mit B und x
-Intuition: x als sample für A zu haben ist wertvoller als für B

Table 2: Instantiations of SIM. Note how the relationships in Tab. 1 can be applied to SCMI instantiations to obtain SMI and SCG instantiations.

(a) Instantiations of SMI functions.

SMI	$I_f(\mathcal{A};\mathcal{Q})$
FLVMI	$\sum_{i\in\mathcal{U}} \min(\max_{j\in\mathcal{A}} S_{ij}, \max_{j\in\mathcal{Q}} S_{ij})$
FLQMI	$\sum_{i\in\mathcal{Q}} \max_{j\in\mathcal{A}} S_{ij} + \sum_{i\in\mathcal{A}} \max_{j\in\mathcal{Q}} S_{ij}$
GCMI	$2\sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{Q}} S_{ij}$
LOGDETMI	$\log\det(S_\mathcal{A}) - \log\det(S_\mathcal{A} - S_{\mathcal{A},\mathcal{Q}} S_\mathcal{Q}^{-1} S_{\mathcal{A},\mathcal{Q}}^T)$

(b) Instantiations of SCG and SCMI functions.

SCG	$f(\mathcal{A}\|\mathcal{P})$
FLCG	$\sum_{i\in\mathcal{U}} \max(\max_{j\in\mathcal{A}} S_{ij} - \max_{j\in\mathcal{P}} S_{ij}, 0)$
LogDetCG	$\log\det(S_\mathcal{A} - S_{\mathcal{A},\mathcal{P}} S_\mathcal{P}^{-1} S_{\mathcal{A},\mathcal{P}}^T)$
GCCG	$f(\mathcal{A}) - 2\sum_{i\in\mathcal{A},j\in\mathcal{P}} S_{ij}$

SCMI	$I_f(\mathcal{A};\mathcal{Q}\|\mathcal{P})$
FLCMI	$\sum_{i\in\mathcal{U}} \max(\min(\max_{j\in\mathcal{A}} S_{ij}, \max_{j\in\mathcal{Q}} S_{ij}) - \max_{j\in\mathcal{P}} S_{ij}, 0)$
LogDetCMI	$\log \frac{\det(I - S_\mathcal{P}^{-1} S_{\mathcal{P},\mathcal{Q}} S_\mathcal{Q}^{-1} S_{\mathcal{P},\mathcal{Q}}^T)}{\det(I - S_{\mathcal{A}\cup\mathcal{P}}^{-1} S_{\mathcal{A}\cup\mathcal{P},\mathcal{Q}} S_\mathcal{Q}^{-1} S_{\mathcal{A}\cup\mathcal{P},\mathcal{Q}}^T)}$

hand. Towards this end, we use the inspiration from [3] where they select data points based on diverse gradients. The SIM functions (see Tab. 2) are instantiated using similarity kernels computed using pairwise similarities S_{ij} between the gradients of the current model. Specifically, we define $S_{ij} = \langle \nabla_\theta \mathcal{H}_i(\theta), \nabla_\theta \mathcal{H}_j(\theta) \rangle$, where $\mathcal{H}_i(\theta) = \mathcal{H}(x_i, y_i, \theta)$ is the loss on the ith data point. Similar to [44, 3], we use hypothesized labels for computing the gradients, and the corresponding similarity kernels. The hypothesized label for each data point is assigned as the class with the maximum probability. We then optimize a SCMI function:

$$\max_{\mathcal{A}\subseteq\mathcal{U}, |\mathcal{A}|\leq B} I_f(\mathcal{A};\mathcal{Q}|\mathcal{P}) \tag{1}$$

with appropriate choices of query set $\mathcal{Q}$ and conditioning set $\mathcal{P}$. In the context of batch active learning, $\mathcal{A}$ is the batch and B is the budget (batch size in AL). We present our unified AL framework in Algorithm 1 and illustrate the choices of query and conditioning set for realistic scenarios in Fig. 2.

Algorithm 1 SIMILAR: Unified AL Framework

Require: Initial Labeled set of data points: $\mathcal{L}$, large unlabeled dataset: $\mathcal{U}$, Loss function $\mathcal{H}$ for learning model $\mathcal{M}$, batch size: B, number of selection rounds: N

1: **for** selection round $i = 1 : N$ **do**
2: Train model $\mathcal{M}$ with loss $\mathcal{H}$ on the current labeled set $\mathcal{L}$ and obtain parameters θ
3: Using model parameters θ_i, compute gradients using hypothesized labels $\{\nabla_\theta \mathcal{H}(x_j, \hat{y}_j, \theta), \forall j \in \mathcal{U}\}$ and obtain a similarity matrix X.
4: Instantiate a submodular function f based on X.
5: $\mathcal{A}_i \leftarrow \text{argmax}_{\mathcal{A}\subseteq\mathcal{U}, |\mathcal{A}|\leq B} I_f(\mathcal{A};\mathcal{Q}|\mathcal{P})$ (Optimize SCMI with an appropriate choice of $\mathcal{Q}$ and $\mathcal{P}$, see Tab. 1)
6: Get labels $L(\mathcal{A}_i)$ for batch $\mathcal{A}_i$ and $\mathcal{L} \leftarrow \mathcal{L} \cup L(\mathcal{A}_i)$, $\mathcal{U} \leftarrow \mathcal{U} - \mathcal{A}_i$
7: **end for**
8: Return trained model $\mathcal{M}$ and parameters θ.

In the scenarios below, we will discuss how this paradigm can provide a unified view of active learning, handle aspects like standard active learning (Sec. 3.1), rare classes and imbalance (Sec. 3.2), redundancy (Sec. 3.3) and, OOD/outliers in the unlabeled data (Sec. 3.4).

3.1 Standard Active Learning

We refer to standard active learning for ideal scenarios when there is no imbalance, redundancy or OOD data in the unlabeled dataset. In such cases, there is no requirement for having a query set and conditioning set. Hence, given a SCMI function $I_f(\mathcal{A};\mathcal{Q}|\mathcal{P})$, we get $I_f(\mathcal{A};\mathcal{Q}|\mathcal{P}) = f(\mathcal{A})$ by setting $\mathcal{Q} \leftarrow \mathcal{U}$ (the unlabeled dataset) and $\mathcal{P} \leftarrow \emptyset$. In a nutshell, the standard diversified active learning setting can be seen as a special case of our proposed unified AL framework (Equ. (1)) by choosing $\mathcal{Q}, \mathcal{P}$ as above. Note that this approach is very similar and closely related to BADGE [3], where the authors also choose points based on diverse gradients. Furthermore, the authors discuss the use of Determinantal Point Processes (DPP) [27] for sampling, and this is very similar to maximizing log-determinants. In the supplementary paper, we compare the choice of different submodular functions for AL.

3.2 Rare Classes

A very common and naturally occurring scenario is that of imbalanced data. This imbalance is because some classes or attributes are naturally more frequently occurring than others in the real-world. For example, in a self-driving car application, there may be very few images of pedestrians at night on highways, or cyclists at night. Another example is medical imaging, where there are many rare yet important diseases (e.g., various forms of cancers), and it is often the case that non-cancerous images are much more than compared to the cancerous ones. While such classes are rare, it is also critical to be able to perform well in these classes. The problem with running standard active learning algorithms in such a case is that they may not sample too many data points from these rare classes, and as a result, the model continues to perform poorly on these classes. In such cases, we can create a (small) held-out set $\mathcal{R}$ which contains data points from these rare classes, and try to encourage the AL by sampling more of these rare classes by maximizing the SMI function $I_f(\mathcal{A}; \mathcal{R})$:

$$\max_{\mathcal{A} \subseteq \mathcal{U}, |\mathcal{A}| \leq B} I_f(\mathcal{A}; \mathcal{R}) \tag{2}$$

This setting is shown in Fig. 2(a). $\mathcal{R}$ contains a small number of held-out examples of classes $5, 8$ which are rare, and the AL acquisition function is Equ. (2). Note that this is exactly equivalent to maximizing the SCMI function with $\mathcal{Q} \leftarrow \mathcal{R}$ and $\mathcal{P} \leftarrow \emptyset$ (i.e. Equ. (1) in Line 5 of Algorithm 1). Furthermore, since the SMI functions naturally model query relevance and diversity, they will also try to pick a diverse set of data points which are relevant to $\mathcal{R}$. Finally, we also point out that this setting was considered in [15] where they use a FISHER kernel based approach to sample data points. Note that for this setting to be realistic, it is critical that the size of this validation set is very small – [15] uses a much larger validation set which is not very realistic (e.g., $200\times$ our set, see Appendix B for more details).

3.3 Redundancy in Unlabeled Data

Another commonplace scenario is where we are dealing with a lot of redundancy – *e.g.,* frames sampled from a video, where subsequent frames are visually similar. In such cases, existing AL algorithms tend to pick data points that are semantically similar to the ones selected in some earlier batch. This is true even for the state-of-the-art AL algorithm BADGE [3] that attempts to enforce diversity, but only in the current batch of data points and not the already selected labeled set. To illustrate this, consider the scenario in Fig. 2(b). The digits $0, 1$ are redundant in the unlabeled set, and they are already present in the labeled set $\mathcal{L}$. Algorithms which just focus on diversity in the current batch could fail at ensuring diversity across batches. To mitigate inter-batch redundancy, we use SCG acquisition function and condition upon the already labeled set $\mathcal{L}$:

$$\max_{\mathcal{A} \subseteq \mathcal{U}, |\mathcal{A}| \leq B} f(\mathcal{A}|\mathcal{L}) \tag{3}$$

Notice that this is a special case of our proposed unified AL framework (Equ. (1)) since the SCG function $f(\mathcal{A}|\mathcal{L})$ is basically a SCMI function with $\mathcal{Q} \leftarrow \mathcal{U}$ and $\mathcal{P} \leftarrow \mathcal{L}$.

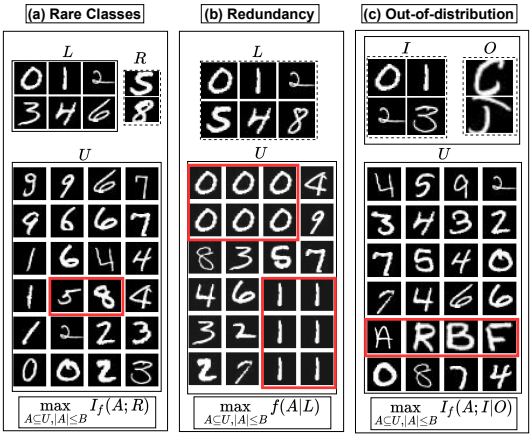

Figure 2: An illustration of realistic scenarios where SIMILAR is applied with appropriate choices of query and conditioning sets: a) SIMILAR finds rare digits $5, 8 \in \mathcal{U}$, by optimizing the SMI function $I_f(\mathcal{A}; \mathcal{R})$ with $\mathcal{R}$ containing $5, 8$ as *queries*, b) select samples from $\mathcal{U}$ which are diverse among themselves and also diverse w.r.t those in $\mathcal{L}$ by optimizing $f(\mathcal{A}|\mathcal{L})$ (here, we want to *avoid* digits $0, 1 \in \mathcal{U}$ altogether because they are present in $\mathcal{L}$), c) select digits (in-distribution) and avoid alphabets (out-of-distribution) in $\mathcal{U}$ by optimizing $I_f(\mathcal{A}; \mathcal{I}|\mathcal{O})$, where $\mathcal{I}$ are ID labeled points and $\mathcal{O}$ are OOD points selected so far.

3.4 Out of Distribution Data

In real world scenarios, we often have out-of-distribution (OOD) data or irrelevant classes in the unlabeled set. Such OOD data is not useful for the given classification task at hand. Using an acquisition function that selects a lot of OOD data points will lead to a waste of labeling effort and time. This is because annotators have to spend time in filtering out OOD data points and discard them from the training dataset. To account for OOD data,

we add an additional class called "OOD" in our model. Since the goal is to improve on in-distribution classes , we ignore the prediction for the OOD class at test time. For our AL acquisition function, we use the currently labeled OOD points $\mathcal{O}$ as the conditioning set $\mathcal{Q}$, and the currently labeled in-distribution (ID) points $\mathcal{I}$ as the query set. In other words, our acquisition function is to optimize:

$$\max_{\mathcal{A} \subseteq \mathcal{U}, |\mathcal{A}| \leq B} I_f(\mathcal{A}; \mathcal{I} | \mathcal{O}) \tag{4}$$

This is illustrated in Fig. 2(c), where the labeled set consists of six examples, four of them being ID data points (set $\mathcal{I}$) and two being OOD data points (set $\mathcal{O}$). In Fig. 2(c), the ID data are digits (digit classification) and the OOD examples are alphabets. This SCMI based approach will naturally pick points "close" to the ID data while avoiding the OOD points.

Another approach for designing the acquisition function is to not explicitly condition on the OOD data points. In other words, we can just optimize the SMI function:

$$\max_{\mathcal{A} \subseteq \mathcal{U}, |\mathcal{A}| \leq B} I_f(\mathcal{A}; \mathcal{I}) \tag{5}$$

We contrast the choices of SCMI (Equ. (4)) and SMI (Equ. (5)) functions in our experiments.

3.5 Scalability and Computational Aspects of SIMILAR

Computational Complexity: The computational complexity of the different SMI functions are determined by (1) the kernel computation time, and (2) the time complexity of the greedy algorithm. All functions considered here are graph based functions and require computing a kernel matrix. The LOGDET functions (LOGDET, LOGDETMI, LOGDETCG, LOGDETCMI), some FL functions (FL, FLVMI, FLCMI), and GC, GCMI all require the $n \times n$ similarity matrix ($n = |\mathcal{U}|$ is the number of unlabeled points) which entails a complexity of $O(n^2)$ to construct the similarity kernel. Once constructed, the complexity of the greedy algorithm for LOGDET class of functions is roughly $O(B^3 n)$ [11], while the complexity of the greedy algorithm with FL, FLVMI, and FLCMI is $O(Bn^2)$ [18, 20](B is the batch size). Different from others, FLQMI does not require computing a $n \times n$ kernel, but only a $n \times q$ kernel (where $q = |\mathcal{Q}|$ is the number of query points). Correspondingly, the complexity of the greedy algorithm with FLQMI is $O(nqB)$, and is linear in n. In Appendix. A, we provide a detailed summary of the complexity of different SF, SMI, SCG, and SCMI functions.

Partition Trick: The deal with the high $O(n^2)$ of the LOGDET, GC, and some of the FL variants (except FLQMI), we also propose the following partitioning algorithm: We randomly split the unlabeled set $\mathcal{U}$ into p partitions $\mathcal{U}_1, \cdots, \mathcal{U}_p$, and we then define the corresponding function (SF, SMI, SCMI, SCG) on each of the partitions and independently optimize them. In each partition, we select B/p points. The complexity of this reduces from $O(n^2)$ to $O(n^2/p)$ and with an appropriate choice of p, we can significantly reduce the computational complexity. We use this in our ImageNet experiments (see Sec. 4.1), and observe that our approaches continue performing well while being more scalable. We provide more details on partitioning in Appendix. A.

Last Layer Gradients: Deep models have numerous parameters leading to very high dimensional gradients. Since our kernel matrix is computed using the cosine similarity of gradients, this becomes intractable for most models. To solve this problem, we use last-layer gradient approximation by representing data points using last layer gradients. BADGE [3], CORESET [39] and GLISTER [23] are other baselines that also use this approximation. Using this representation, we compute a pairwise cosine similarity matrix to instantiate acquisition functions in SIMILAR (see lines 3,4 in Algorithm 1).

4 Experimental Results

In this section, we empirically evaluate the effectiveness of SIMILAR on a wide range of scenarios like rare classes (Sec. 4.1), redundancy (Sec. 4.2) and out-of-distribution (Sec. 4.3). We do so by comparing the accuracy and selections of various SCMI based acquisition functions with existing AL approaches. Using these experiments, we cover the issues with the current AL methods and show that these issues can be mitigated by using a unified implementation using SCMI with appropriate choices of query and/or conditioning sets. Although this section focuses on realistic scenarios, we also study SIMILAR in a standard active learning setting and show that it performs at par with current AL methods (see Appendix. C).

Baselines in all scenarios: We compare SCMI based functions against several methods. Particularly, we compare against: (1) three uncertainty based AL algorithms: i)ENTROPY: Selects the top B data

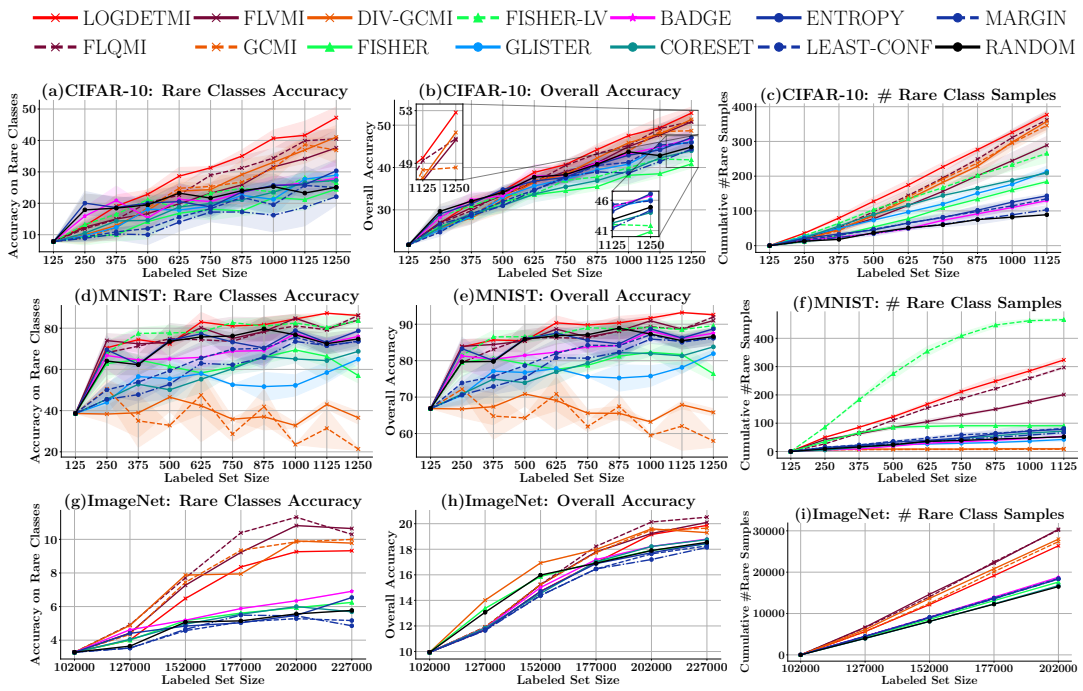

Figure 3: Active Learning with rare classes on CIFAR-10 (top row), MNIST (middle row), and ImageNet (bottom row). Left side plots (a,d,g) are rare class accuracies, center plots (b,e,h) are overall test accuracies, right plots (c,f,i) are a number of rare class samples selected. The SMI functions (specifically LOGDETMI, FLQMI) outperform other baselines by more than 10% on the rare classes.

points with the highest *entropy* [40], ii) MARGIN: Select the bottom B data points that have the least difference in the confidence of first and the second most probable labels [36], iii)LEAST-CONF: Select B samples with the smallest predicted class probability [43], (2) state-of-the-art diversity based algorithms: iv) BADGE [3] v) GLISTER [23] vi) CORESET [39] which are all discussed in section Sec. 1.1, and, 3) RANDOM: Select B samples randomly. Additionally, in the rare classes scenario, we compare against FISHER [15] which is also discussed in Sec. 1.1.

Datasets, model architecture and experimental setup: We apply our framework to CIFAR-10 [26] and MNIST [29] classification tasks. Additionally, we also evaluate our method on down sampled 32×32 ImageNet-2012 [37] for the rare classes setting (Sec. 4.1). Due to the lack of test split on ImageNet, we used the validation split for evaluation. In the sections below, we discuss the individual splits for $\mathcal{L}, \mathcal{U}, \mathcal{R}, \mathcal{I}$, and $\mathcal{O}$ in each realistic scenario. To ensure that all the selection algorithms that we are studying are given fair and equal treatment across all realistic scenarios, we use a common training procedure and hyperparameters. We use standard augmentation techniques like random crop, horizontal flip followed by data normalization except for MNIST which does not use horizontal flip to preserve labels. For training, we use an SGD optimizer with an initial learning rate of 0.01, the momentum of 0.9, and a weight decay of 5e-4. We decay the learning rate using cosine annealing [30] for each epoch. On all datasets except MNIST, we train a ResNet18 [17] model, while on MNIST we train a LeNet [28] model. For all the experiments in a particular scenario (rare classes, redundancy and OOD), we start with an identical initial model $\mathcal{M}$ and initial labeled set $\mathcal{D}$. We reinitialize the model parameters at the beginning of every selection round using Xavier initialization and train the model until either the training accuracy reaches 99% or the epoch count reaches 150. We run each experiment $3\times$ on CIFAR-10 and MNIST and $1\times$ on ImageNet and provide error bars (std deviation). All experiments were run on a V100 GPU. For more details on the experimental setup, baselines, and datasets see Appendix. B.

4.1 Rare Classes

Custom dataset: Following [15, 23], we simulate these rare classes by creating a class imbalance. We initialize the batch active learning experiments by creating a custom dataset which is a subset of the full dataset with the same marginal distribution. Given that $\mathcal{C}$ consists of data points from the imbalanced classes and $\mathcal{D}$ consists of data points from the balanced classes, we create an initial labeled set $\mathcal{L}$ such that $|\mathcal{C}_\mathcal{L}| = \rho|\mathcal{D}_\mathcal{L}|$ and an unlabeled set $|\mathcal{C}_\mathcal{U}| = \rho|\mathcal{D}_\mathcal{U}|$, where ρ is the imbalance

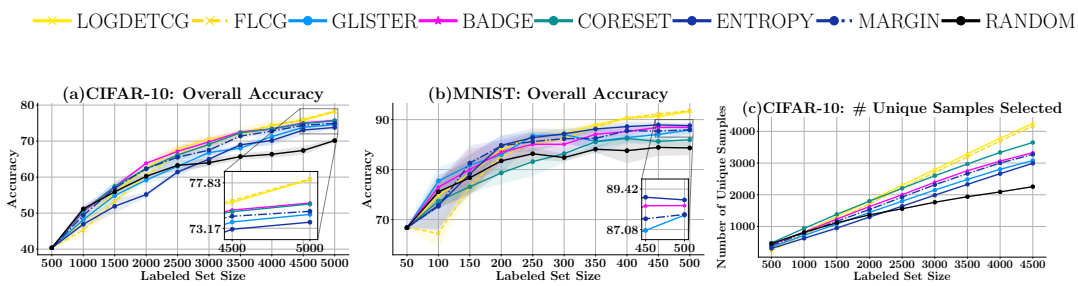

Figure 4: Active Learning under $10\times$ redundancy for CIFAR-10 and MNIST. The CG functions (LOGDETCG, FLCG) pick more unique points and outperform existing algorithms including BADGE.

factor. We use a small and clean validation/query set $\mathcal{R}$ containing data points from the imbalanced classes (≈ 3 data points per imbalanced class). We create an imbalance in CIFAR-10 using 5 random classes, $\rho = 10$ and for MNIST we create an imbalance using the same classes as in [15] $(5 \cdots 9)$ and use $\rho = 20$. For both datasets: $|\mathcal{C}_{\mathcal{L}}| + |\mathcal{D}_{\mathcal{L}}| = 125$, $|\mathcal{C}_{\mathcal{U}}| + |\mathcal{D}_{\mathcal{U}}| = 21K$, $B = 125$ (AL batch size) and, $|\mathcal{R}| = 25$ (size of the held out rare instances). For MNIST, we also present the results for $B = 25$ and $\rho = 100$ in the supplementary. On ImageNet, we randomly select 500 classes out of 1000 classes for imbalance and $\rho = 5$ such that $|\mathcal{C}_{\mathcal{L}}| + |\mathcal{D}_{\mathcal{L}}| = 102K$, $|\mathcal{C}_{\mathcal{U}}| + |\mathcal{D}_{\mathcal{U}}| = 664K$, $B = 25K$ and, $|\mathcal{R}| = 2.5K$. These data splits are chosen to simulate a low initial accuracy on the rare classes and at the same time maintain the imbalance factor in the labeled and unlabeled datasets.

Results: The results are shown in Fig. 3. We observe that SMI based functions not only consistently outperform uncertainty based methods (ENTROPY, LEAST-CONF and MARGIN) but also all the state-of-the-art diversity based methods (BADGE, GLISTER, CORESET) by $\approx 5 - 10\%$ in terms of overall accuracy and $\approx 10 - 18\%$ in terms of average accuracy on rare classes (see Fig. 3a, 3d, 3g). The reason for the same can be seen in Fig. 3c, 3f, 3i which illustrates that they fail to pick an adequate number of examples from the rare classes. Evidently, FLQMI and LOGDETMI which balance between diversity and relevance perform better than GCMI which only models relevance. Furthermore, DIV-GCMI which is a linear combination of GCMI and a diversity term performs consistently worse, which suggest that a naive combination of the two may not be as effective. This suggests the need of SMI based acquisitions functions (Equ. (2)) with richer modeling capabilities like FLQMI and LOGDETMI within SIMILAR. Furthermore, all SMI based functions also outperform the FISHER kernel based method when the validation set is small and realistic, *i.e.,* $|\mathcal{R}| = 25$. Since, [15] use a very large validation set in their experiments, we try their method FISHER-LV with a $40\times$ larger validation set of size 1000 (which is *not practical*) and observe a comparable performance with the SMI functions which use a small validation set. Furthermore, we see that FISHER-LV actually picks significantly larger number of rare class instances in MNIST, but yet is comparable in performance of FLQMI and LOGDETMI. This suggests that both these methods select higher quality and diverse rare class instances. We observe that the GC SMI variants(GCMI and DIV-GCMI) do not perform well on MNIST classification. Finally, we point out in the case of ImageNet, FLQMI performs the best and outperforms FLVMI and LOGDETMI – this is because we do not need to do the partition trick for FLQMI since it is already linear in time complexity. For FLVMI and LOGDETMI, we set the number of partitions $p = 50$ for ImageNet. Finally, we do a pairwise t-test to compare the performance of the algorithms (Appendix. D) and observe that the *SMI functions (and particularly* FLVMI *and* LOGDETMI*) statistically significantly outperform all AL baselines.*

4.2 Redundancy

Custom dataset: To simulate a realistic redundancy scenario we create a custom dataset by duplicating 20% of the unlabeled dataset $10\times$. For CIFAR-10, the number of unique points in the unlabeled set $|\mathcal{U}| = 5K$, the initial labeled set $|\mathcal{L}| = 500$, $B = 500$, whereas for MNIST $|\mathcal{U}| = 500$, $|\mathcal{L}| = 50$ and $B = 50$. For MNIST, we also present the results for $5\times$ and $20\times$ in the Appendix. E.

SCG vs Baselines: As expected, the diversity and uncertainty based methods outperform random. Importantly, we observe that the SCG functions (FLCG and LOGDETCG) significantly outperform all baselines by $\approx 3 - 5\%$ towards the end as the conditioning gets stronger with increase in $\mathcal{L}$ (see Fig. 4a, 4b). This implies that simply relying on model parameters for diversity and/or uncertainty is not sufficient and that conditioning on the updated labeled set $\mathcal{L}$ (Equ. (3)) is required in batch active learning. In Fig. 4c we show that SCG based acquisition functions select significantly more unique

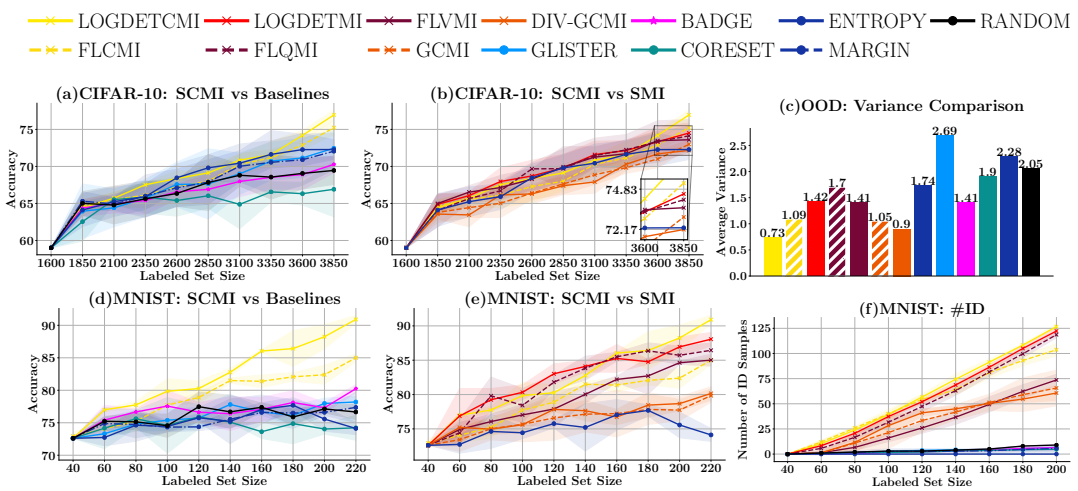

Figure 5: Active Learning with OOD data in unlabeled set. Top row: CIFAR-10 results for (a) SCMI vs Baselines, (b) SCMI vs SMI, and (c) variance comparison of different baselines, bottom row: MNIST results for (d) SCMI vs Baselines, (e) SCMI vs SMI, and (f) Number of ID points selected. We see that, i) the SCMI functions consistently outperform the baselines by $5\% - 10\%$, ii) SCMI functions outperform the corresponding SMI functions for later rounds, and (iii) SCMI functions have the least variance compared to the rest, showing that they are more robust in performance.

data points than other baselines. We also perform a pairwise t-test (Appendix. E), to prove that the SCG functions consistently and statistically significantly outperform BADGE and other baselines.

4.3 Out-Of-Distribution

Custom dataset: We simulated a scenario where we convert the classification problem in CIFAR-10 and MNIST to a 8-class classification, where the first 8 classes represent the set $\mathcal{I}_F$ of in-distribution (ID) data points and the last 2 represent the set $\mathcal{O}_F$ of out-of-distribution(OOD) data points. The initial labeled set $\mathcal{L}$ *consists only of ID points*, i.e. $\mathcal{O}_F \cap \mathcal{L} = \emptyset$. The unlabeled set is simulated to reflect a realistic and somewhat extreme setting where the unlabeled ID data points $|\mathcal{I}_F|$ is much smaller than the unlabeled OOD data points $|\mathcal{O}_F|$. Additionally, we also assume we have a very small validation set of ID points $\mathcal{I}_V$. For CIFAR-10: $|\mathcal{L}| = 1.6K$, $|\mathcal{I}_F| = 4K$, $|\mathcal{O}_F| = 10K$, $|\mathcal{I}_V| = 40$, $B = 250$ whereas for MNIST which is a relatively simpler task, we use a smaller initial labeled sets and keep the unlabeled sets of the same size: $|\mathcal{L}| = 40$, $|\mathcal{I}_F| = 400$, $|\mathcal{O}_F| = 10K$, $|\mathcal{I}_V| = 16$, $B = 20$. Recall that our algorithm uses ID set $\mathcal{I}$ (initialized to $\mathcal{I}_V$) and OOD set $\mathcal{O}$ which we build as follows. Every time our selection approach selects a set $\mathcal{A}$, we update $\mathcal{I} = \mathcal{I} \cup (\mathcal{A} \cap \mathcal{I}_F)$ and $\mathcal{O} = \mathcal{O} \cup (\mathcal{A} \cap \mathcal{O}_F)$, i.e. we augment the ID and OOD points in $\mathcal{A}$ to the sets $\mathcal{I}$ and $\mathcal{O}$ respectively.

SCMI vs Baselines: Since we care about the predictive performance of the ID classes, we report the ID classes accuracy. We see that SCMI based acquisition functions significantly outperform existing AL approaches by $\approx 5 - 10\%$ (see Fig. 5a, 5d). We also observe that existing acquisition functions have a high variance which is undesirable in real-world deployment scenarios where deep models are been continuously developed. Our SCMI based acquisition functions (LOGDETCMI and FLCMI) show the lowest variance in training (see Fig. 5c). This reinforces the need of having a framework like SIMILAR that facilitates query and conditioning sets.

SCMI vs SMI: We compare SCMI functions against SMI functions to study the effect of conditioning and observe that the SCMI functions are comparable to the SMI functions initially but in the later selection rounds of active learning, the SCMI functions consistently outperform SMI functions. In particular, we see an improvement of $2 - 3\%$ as the conditioning becomes stronger (see Fig. 5b, 5e). We also observe the SCMI tends to select more ID points than SMI and other baselines (see Fig. 5f), and SCMI functions have a lower variance overall compared to even the SMI functions (Fig. 5c).

5 Conclusion

In this paper, we proposed a unified active learning framework SIMILAR using the submodular information functions. We showed the applicability of the framework in three realistic scenarios for active learning, namely rare classes, redundancy, and out of distribution data. In each case, we

observed that the functions in SIMILAR significantly outperform existing baselines in each of these tasks. Our real world experiments on MNIST, CIFAR-10, and ImageNet show that many of the SIM functions (specifically the LOGDET and FL variants) yield $\approx 5\% - 18\%$ gain compared to existing baselines particularly in the rare class scenario and $\approx 5\% - 10\%$ OOD scenarios. The main limitations of our work is the dependence on good representations to compute similarity. A potential negative societal impact of this work is the use of SIMILAR to perpetuate certain biases through a malicious use of the query and conditioning set. We discuss this in more detail in Appendix. G.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 978-0-898716-24-5.

[3] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

[4] Francis Bach. Learning with submodular functions: A convex optimization perspective. *arXiv preprint arXiv:1111.6453*, 2011.

[5] Francis Bach. Submodular functions: from discrete to continuous domains. *Mathematical Programming*, 175(1):419–459, 2019.

[6] Christopher Berlind and Ruth Urner. Active nearest neighbors in changing environments. In *International Conference on Machine Learning*, pages 1870–1879. PMLR, 2015.

[7] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.

[8] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[9] Colin Campbell, Nello Cristianini, Alex Smola, et al. Query learning with large margin classifiers. In *ICML*, volume 20, page 0, 2000.

[10] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.

[11] Laming Chen, Guoxin Zhang, and Hanning Zhou. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 5627–5638, 2018.

[12] Shai Fine, Ran Gilad-Bachrach, and Eli Shamir. Query by committee, linear separation and random walks. *Theoretical Computer Science*, 284(1):25–51, 2002.

[13] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2):133–168, 1997.

[14] Satoru Fujishige. *Submodular functions and optimization*. Elsevier, 2005.

[15] Denis Gudovskiy, Alec Hodgkinson, Takuya Yamaguchi, and Sotaro Tsukizawa. Deep active learning for biased datasets via fisher kernel self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9041–9049, 2020.

[16] Anupam Gupta and Roie Levin. The online submodular cover problem. In *ACM-SIAM Symposium on Discrete Algorithms*, 2020.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Rishabh Iyer and Jeffrey Bilmes. A memoization framework for scaling submodular optimization to large scale problems. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2340–2349. PMLR, 2019.

[19] Rishabh Iyer, Ninad Khargoankar, Jeff Bilmes, and Himanshu Asanani. Submodular combinatorial information measures with applications in machine learning. In *Algorithmic Learning Theory*, pages 722–754. PMLR, 2021.

[20] Rishabh Krishnan Iyer. *Submodular optimization and machine learning: Theoretical results, unifying and scalable algorithms, and applications*. PhD thesis, 2015.

[21] Kimmo Karkkainen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1548–1558, 2021.

[22] Vishal Kaushal, Suraj Kothawade, Ganesh Ramakrishnan, Jeff Bilmes, and Rishabh Iyer. Prism: A unified framework of parameterized submodular information measures for targeted data subset selection and summarization. *arXiv preprint arXiv:2103.00128*, 2021.

[23] Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. *arXiv preprint arXiv:2012.10630*, 2020.

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[25] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *arXiv preprint arXiv:1906.08158*, 2019.

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[27] Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *arXiv preprint arXiv:1207.6083*, 2012.

[28] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[29] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. at&t labs, 2010.

[30] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[31] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[32] Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, Amin Karbasi, Jan Vondrák, and Andreas Krause. Lazier than lazy greedy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[33] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

[34] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[36] Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *European Conference on Machine Learning*, pages 413–424. Springer, 2006.

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[38] Greg Schohn and David Cohn. Less is more: Active learning with support vector machines. In *ICML*, volume 2, page 6. Citeseer, 2000.

[39] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[40] Burr Settles. Active learning literature survey. 2009.

[41] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019.

[42] Ehsan Tohidi, Rouhollah Amiri, Mario Coutino, David Gesbert, Geert Leus, and Amin Karbasi. Submodularity in action: From machine learning to signal processing applications. *IEEE Signal Processing Magazine*, 37(5):120–133, 2020.

[43] Dan Wang and Yi Shang. A new active labeling method for deep learning. In *2014 International joint conference on neural networks (IJCNN)*, pages 112–119. IEEE, 2014.

[44] Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pages 1954–1963. PMLR, 2015.

[45] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *arXiv preprint arXiv:1509.01626*, 2015.

[46] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

Supplementary Material for SIMILAR: Submodular Information Measures Based Active Learning In Realistic Scenarios

Table of Contents

A Computational Aspects of SIM Functions in SIMILAR

A.1 Computational complexity for selection using each function in SMI and baselines

Below, we provide a detailed analysis of the complexity of creating and optimizing the different SIM functions. Denote $|\mathcal{X}|$ as the size of set $\mathcal{X}$. Also, let $|\mathcal{U}| = n$ (the ground set size, which is the size of the unlabeled set in this case). In the main paper, we provided the high level intuition of the complexity, ignoring the terms of $|\mathcal{P}|$ and $|\mathcal{Q}|$ since they would be typically much smaller than the number of unlabeled points n. For completeness, we provide the detailed complexity below:

- **Facility Location:** We start with FLVMI. The complexity of creating the kernel matrix is $O(n^2)$. The complexity of optimizing it is $\tilde{O}(n^2)$ (using memoization [18])[2] if we use the stochastic greedy algorithm [32] and $O(n^2k)$ with the naive greedy algorithm. The overall complexity is $\tilde{O}(n^2)$. For FLQMI, the cost of creating the kernel matrix is $O(n|\mathcal{Q}|)$, and the cost of optimization is also $\tilde{O}(n|\mathcal{Q}|)$ (with naive greedy, it is $O(nB|\mathcal{Q}|)$). The complexity of FLCG is $O([n + |\mathcal{P}|]^2)$ to compute the kernel matrix and $\tilde{O}(n^2)$ for optimizing (using the stochastic greedy algorithm). Finally, for FLCMI, the complexity of computing the kernel matrix is $O([n + |\mathcal{Q}| + |\mathcal{P}|]^2)$, and the complexity of optimization is $\tilde{O}(n^2)$.
- **Log-Determinant:** We start with LogDetMI. The complexity of the kernel matrix computation (and storage) is $O(n^2)$. The complexity of optimizing the LogDet function using the stochastic greedy algorithm is $\tilde{O}(B^2n)$, so the overall complexity is $\tilde{O}(n^2 + B^2n)$. For LogDetCG, the complexity of computing the matrix is $O([n + |\mathcal{P}|]^2$, and the complexity of optimization is $\tilde{O}([B +$

[2] $\tilde{O}$: Ignoring log-factors

$|\mathcal{P}|]^2 n)$. For the LogDetCMI function, the complexity of computing the matrix is $O([n+|\mathcal{P}|+|\mathcal{Q}|]^2$, and the complexity of optimization is $\tilde{O}([B+|\mathcal{P}|+|\mathcal{Q}|]^2 n)$.

- **Graph-Cut:** Finally, we study GC functions. For GCMI, we require an $O(n|\mathcal{Q}|)$ kernel matrix, and the complexity of the stochastic greedy algorithm is also $\tilde{O}(n|\mathcal{Q}|)$. Finally, for GCCG, the complexity of creating the kernel matrix is $O(n|^2 + n|\mathcal{P}|)$, and the complexity of the stochastic greedy algorithm is $\tilde{O}(n^2 + n|\mathcal{P}|)$.

We end with a few comments. First, most of the complexity analysis above is with the stochastic greedy algorithm [32]. If we use the naive or lazy greedy algorithm, the worst-case complexity is a factor B larger. Secondly, we ignore log-factors in the complexity of stochastic greedy since the complexity is actually $O(n \log 1/\epsilon)$, which achieves a $1 - 1/e - \epsilon$ approximation. Finally, the complexity of optimizing and constructing the FL, LogDet, and GC functions can be obtained from the CG versions by setting $\mathcal{P} = \emptyset$.

A.2 Details on Partitioning Approach

In some of our experiments, we choose to partition the unlabeled set into chunks in order to meet the scale of the dataset used in that experiment. This is because many of the techniques (specifically LogDet functions, FLVMI, FLCG, FLCMI, GCCG) all have $O(n^2)$ space complexity. For n in the range of a few million to a few billion data points (which is not uncommon in big-data applications today), we need to scale our algorithms to be linear in n and not quadratic. For this, we propose a simple partitioning approach where the unlabeled data is chunked into p partitions. In this strategy, we perform unlabeled instance acquisition on each chunk using a proportional fraction of the full AL batch size. The most notable example of the use of our partitioning strategy is in our down-sampled ImageNet experiment. By performing AL acquisition on the full unlabeled set, almost all AL strategies exhaust the available compute resources. Hence, to execute most of our AL strategies, we partitioned the unlabeled set into 50 equally sized chunks, so each partition has around 10k to 20k instances. As n grows, the number of partitions would also grow so that n/p is roughly constant. The complexity of most approaches discussed above would then be $O(n^2/p)$ ($O(n^2/p^2)$ for each chunk, repeated p times), and if $n/p = r$ is a constant, then the complexity $O(nr)$ would be linear in n. We then acquire a number of unlabeled instances from each chunk whose ratio with the full AL batch size is equal to the ratio between the chunk size and the full unlabeled set. The acquired instances from each chunk are then combined to form the full acquired set of unlabeled instances.

B More Details on Experimental Setup, Datasets, and Baselines

B.1 Datasets description in each scenario

We used various standard datasets – namely, MNIST, CIFAR10, and ImageNet – to demonstrate the effectiveness and robustness of SIMILAR. We also provide additional experiments on SVHN in sections below. We use standard sources for all datasets. As previously mentioned, we perform our experiments on a down-sampled version of ImageNet. Beyond the fact that each image is now 32×32, the data set is otherwise identical. Moreover, we find that the provided validation set is often used as the test set in most evaluations on down-sampled ImageNet. The down-sampled ImageNet training set can be procured here, and the validation set can be found here. Note that associated licenses for all datasets apply.

Rare classes setting: In Tab. 3, we show the exact initial splits used in our experiments for the rare classes scenario. In CIFAR-10, ImageNet, and SVHN, we use randomly chose half the number of classes as imbalanced and the other half as balanced. Following [15], we chose classes $(5, \cdots 9)$ as imbalanced classes in MNIST. We use an AL batch size of 125 for the CIFAR-10, MNIST and SVHN datasets. We use the same data setting for the CIFAR-10 and SVHN datasets with an imbalance factor $\rho = 20$. The results for SVHN are in Appendix. D. For MNIST, we additionally show results for $\rho = 100$ in Appendix. D. Due to the scale of down-sampled ImageNet and the natural imbalance present in its full training set, we adopt a different dataset splitting strategy. Following [15], we randomly chose 500 classes (half) as rare classes. Our train set is initialized as having 34 examples per rare class and 170 examples per normal class. Our validation set contains 5 examples per class, making it balanced. The unlabeled set is created to have 1 rare example for every 5 normal examples. In all, our initialization leads our initial train set, validation set, and unlabeled set to have

approximately 100k, 5k, and 660k points, respectively. We use an AL batch size of 25k points, and we use the same training conditions as before. However, we perform AL selection by dividing the unlabeled set into chunks (partitions), selecting a proportionate fraction of the AL batch size from each. In this case, we divide the unlabeled set into 50 to 100 partitions (determined by compute limitations) and perform selection on each partition.

Dataset	Imbalance factor (ρ)	Labeled (per class)	Valid (per class)	Unlabeled (per class)
CIFAR-10 SVHN	20	3	5	150
		22	5	3000
MNIST	20	3	5	200
		22	5	4000
	100	3	5	40
		22	5	4000

Table 3: Number of data points for each dataset in the rare classes scenario. For CIFAR-10, MNIST, and SVHN, we use 5 balanced classes and 5 imbalanced classes. In the main paper, we show experiments for $\rho = 20$. In Appendix. D, we show experiments for $\rho = 100$.

Redundancy setting: In Tab. 4, we show the exact initial splits used in our experiments for the redundancy scenario. For CIFAR-10 and SVHN, we use the same setting. Since MNIST classification is a relatively simpler problem, we use one tenth of the data points used in the CIFAR-10 setting. For all datasets, we create the unlabeled dataset by duplicating 20% of the unlabeled dataset RF $\times$. We denote RF as the redundancy factor. For instance, we consider 5000 unique points and duplicate 20% of them $10\times$ in CIFAR-10. This gives us $(5000 \times 0.2 \times 10 = 10000)$ duplicated points and $(5000 - (5000 \times 0.2 \times 10) = 4000)$ original points for a total of $(10000 + 4000 = 14000)$ points.

Dataset	Total Unique Points	Fraction of points duplicated	Number of duplicated points
CIFAR-10, SVHN	5000	20%	5000*0.2*RF
MNIST	500	20%	500*0.2*RF

Table 4: Number of data points for each dataset in the redundancy scenario. RF here is the redundancy factor. In the main paper, we show experiments for RF=$10\times$. In Appendix. E, we show experiments for RF=$5\times$ and RF=$20\times$.

Out-of-distribution setting: In Tab. 5, we show the exact initial splits used in our experiments for the out-of-distribution scenario. In all datasets, we chose the first 8 classes to be in-distribution (ID) and the last 2 classes to be out-of-distribution (OOD). Initially, the labeled set consists of only ID points. The unlabeled set is designed to reflect a realistic setting with high number of OOD points. For CIFAR-10, we use 200 points per ID class in the labeled set and 500 points per ID class, 5000 points per OOD class in the unlabeled set. This gives us an initial labeled set of size $200 \times 8 = 1600$ and an initial unlabeled set of size $500 \times 8 + 5000 \times 2 = 14000$. We make the task slightly challenging for MNIST by further decreasing the number of ID points in the unlabeled dataset as shown in Tab. 5.

Dataset		Labeled (per class)	Valid (per class)	Unlabeled (per class)
CIFAR-10	ID points	200	5	500
	OOD points	0	0	5000
MNIST	ID points	5	2	50
	OOD points	0	0	5000

Table 5: Number of data points for each dataset in the out-of-distribution scenario.

B.2 Experimental setup

We ran experiments using an SGD optimizer with an initial learning rate of 0.01, a momentum of 0.9, and a weight decay of 5e-4. We decay the learning rate via cosine annealing [30] for each epoch. For MNIST, we use the LeNet model [28]. For all other datasets, we use ResNet18 model [17]. For each

round of active learning, we train until the accuracy reaches 99% or the epoch count reaches 150. We run all our experiments on a single V100 GPU.

B.3 Details on computation of penalty matrix

The penalty matrices computed in this paper follow the strategy used in [3]. In their strategy, a penalty matrix is constructed for each dataset-model pair. Each cell (i, j) of the matrix reflects the fraction of training rounds that AL with selection algorithm i has higher test accuracy than AL with selection algorithm j with statistical significance. As such, the average difference between the test accuracies of i and j and the standard error of that difference are computed for each training round. A two-tailed t-test is then performed for each training round: If $t > t_\alpha$, then $\frac{1}{N_{train}}$ is added to cell (i, j). If $t < -t_\alpha$, then $\frac{1}{N_{train}}$ is added to cell (j, i). Hence, the full penalty matrix gives a holistic understanding of how each selection algorithm compares against the others: A row with mostly high values signals that the associated selection algorithm performs better than the others; however, a column with mostly high values signals that the associated selection algorithm performs worse than the others. As a final note, [3] takes an additional step where they consolidate the matrices for each dataset-model pair into one matrix by taking the sum across these matrices, giving a summary of the AL performance for their entire paper that is fairly weighted to each experiment. We present the penalty matrices for each of the settings in the sections below.

B.4 Licensing details

Datasets. Our experiments with SIMILAR utilize the following datasets.

- CIFAR-10 [26]: MIT License
- MNIST [29]: Creative Commons Attribution-Share Alike 3.0
- SVHN [34]: CC0 1.0 Public Domain
- ImageNet [37]: Custom (Research, Non-Commercial)

Repositories. Our experiments utilize contributions from existing code repositories. Specifically, we utilize the DISTIL repository for AL baselines. We utilize the Fisher Kernel Self-Supervision repository in our usages of FISHER and its variants. We extensively use PyTorch, and we utilize the CORDS repository in our gradient computations. To summarize, the following repositories are used, and their licenses from their original sources are also provided:

- PyTorch [35]: Modified BSD
- DISTIL: MIT License
- CORDS: MIT License
- Fisher Kernel Self-Supervision [15]: (None Listed)
- BADGE [3]: None Listed

B.5 Baselines and Code

For all baselines, we use code either from existing libraries and codebases or from the authors. For BADGE [3], we use the code from the authors[3]. Similarly, for the FISHER baseline, we use the code from the authors[4]. For the other methods like entropy sampling, CORESET, etc., we use DISTIL[5], which implements most of the state-of-the-art standard AL approaches building upon the respective authors code.

C Results with Standard Active Learning

In Figure 6, we compare the performance of the SFs on standard AL – i.e., without redundancy, out-of-distribution data, and imbalance. The basic idea here is that we compute the similarity kernels

[3] `https://github.com/JordanAsh/badge`

[4] `https://github.com/gudovskiy/al-fk-self-supervision`

[5] `https://github.com/decile-team/distil`

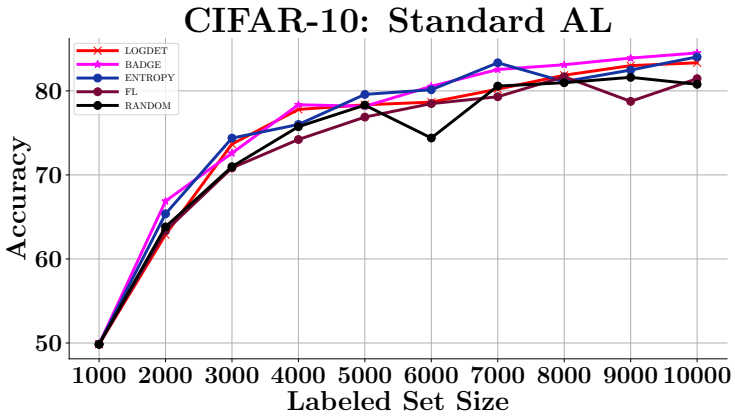

Figure 6: Comparison of submodular functions with baselines in a standard active learning setting.

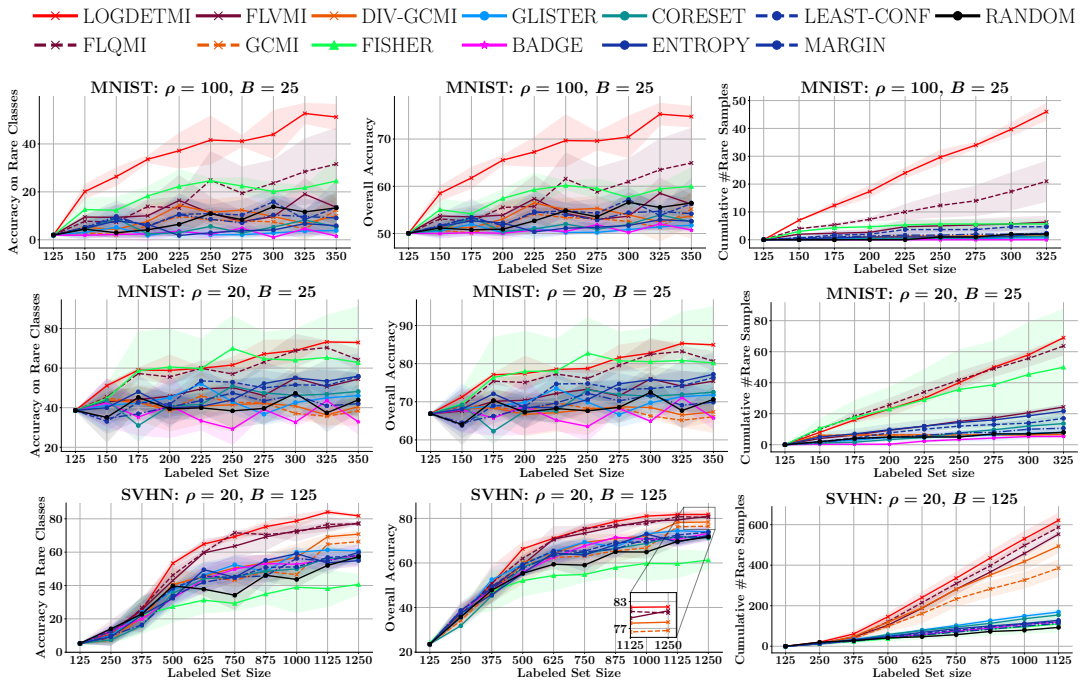

Figure 7: Additional experiments on MNIST and SVHN for active learning with rare classes. **Top row:** MNIST $\rho = 100, B = 25$, LOGDETMI outperforms other methods even in extreme imbalance, with a large gap in accuracy, followed by FLQMI. **Middle row:** MNIST $\rho = 20, B = 25$, LOGDETMI and FLQMI outperform all baselines in the later rounds of AL. **Bottom row:** SVHN $\rho = 20, B = 125$, All SMI methods significantly outperform other baselines.

using the gradients of the model (Algorithm 1) and use just the submodular function – i.e., setting $\mathcal{Q} = \mathcal{U}, \mathcal{P} = \emptyset$. In this work, we use the log-determinant and the facility location functions. We make the following observations: **1)** Log-determinant functions perform comparable to BADGE and entropy sampling, particularly in the beginning. **2)** The facility location function does not perform as well in the standard AL setting, implying that diversity tends to play a more important role in standard active learning compared to representation.

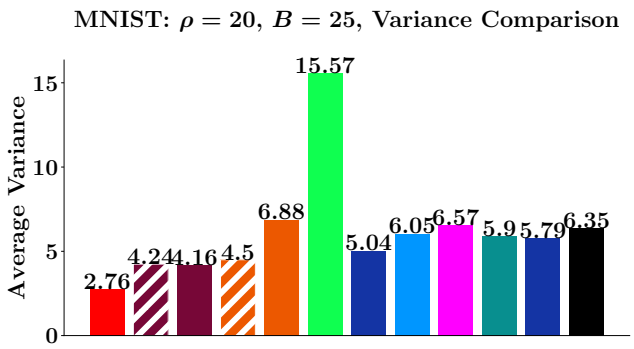

Figure 8: Variance comparison on the rare classes scenario for MNIST $\rho = 20, B = 25$ (Middle row in Fig. 7). FISHER has $\approx 5\times$ variance in comparison with the SMI methods. The figure shares the same legend as Fig. 7.

D Additional Experiments and Takeaways for Active Learning with Rare Classes

In Figure 7, we show additional results for MNIST and SVHN for active learning with rare classes. The top row shows the results for the extreme imbalance scenario, i.e., $\rho = 100, B = 25$ (small batch size and extreme imbalance). We observe that LOGDETMI significantly outperforms all other techniques, and FLQMI and FISHER come next. Note that the FISHER baseline [15] was originally presented in this extreme imbalance scenario. The middle row in Figure 7 contains results for $\rho = 20, B = 25$. This is similar to the results presented in the main paper but using a much smaller batch size. Here, LOGDETMI and FLQMI again outperform the other baselines. While the average performance of the FISHER baseline [15] is comparable to LOGDETMI and FLQMI, it has a much higher variance compared to others (Figure 8). Finally, the bottom row shows the performance of the different techniques on SVHN. Again, we see that LOGDETMI and FLQMI outperform all other techniques.

Takeaways from the Results: The following are the main takeaways of the experiments in this section and the main paper:

- Among the different MI functions, LOGDETMI and FLQMI outperform all other MI functions. They also mostly outperform the Fisher Kernel baseline which was also designed for dealing with rare classes [15].
- LOGDETMI particularly outperforms every other method in the high imbalance regime (100x imbalance). This is mainly because it is able to select the highest number of points from the rare classes (top row, right most plot in Figure 7.
- The FISHER baseline also can have a high variance, particularly when the batch size is high.
- For a fair comparison, we used a very small validation set in all our experiments. As compared in the main paper, FISHER performance does improve when we use a larger validation set, but doing so is not realistic.
- FLQMI is more scalable compared to LOGDETMI and other kernel-based approaches; hence, it is the desired choice of approach for very large datasets.

Penalty Matrix: Figures 9 shows the penalty matrix results on the rare class accuracy (top) and overall accuracy (bottom). We see that LOGDETMI and FLQMI have the smallest column sum, which indicates that most other baselines are not statistically significantly better than them. Furthermore, they also have the highest row sum (followed by some of the other MI functions), which indicates that they are statistically significantly better than other approaches. These matrices are obtained by combining the results on MNIST and CIFAR-10 for $\rho = 20, B = 125$ (i.e., the results in the main paper).

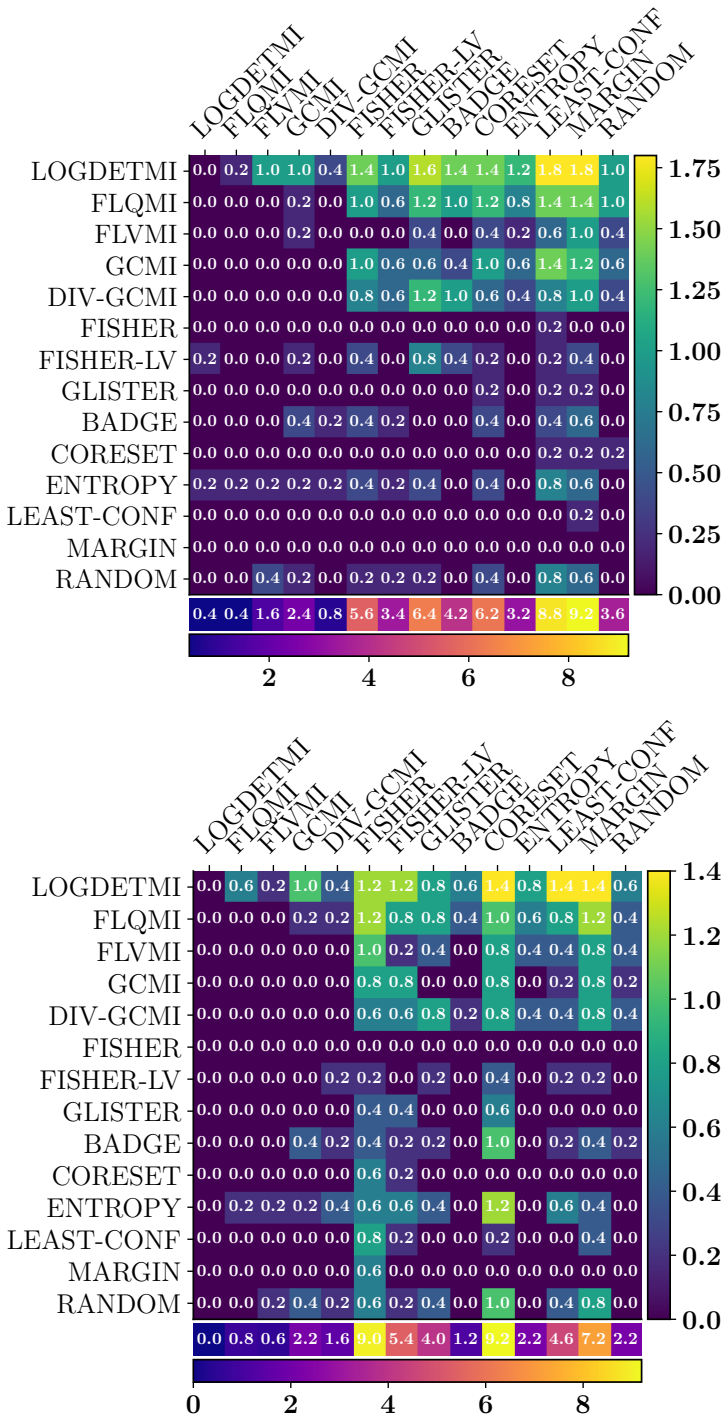

Figure 9: Penalty Matrix comparing the average accuracy of rare classes (**top**) and overall accuracy (**bottom**) of different AL approaches in the class imbalance scenario. We observe that the SMI functions have a much lower column sum compared to other approaches.

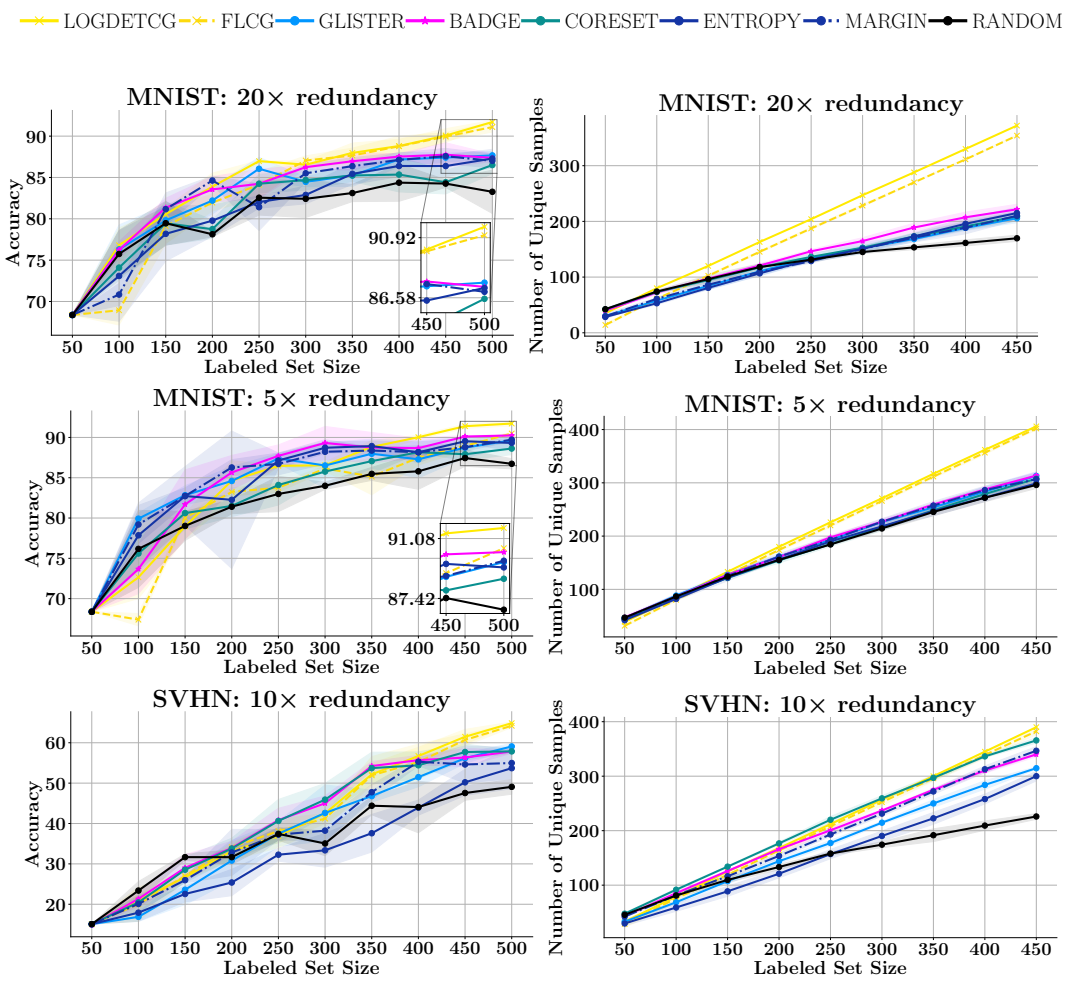

Figure 10: Active Learning under $20\times$ redundancy (**top row**) and $5\times$ redundancy (**middle row**) on MNIST. **Bottom row:** $10\times$ redundancy on SVHN. The CG functions (LOGDETCG, FLCG) pick more unique points and outperform existing algorithms including BADGE.

E Additional Experiments and Takeaways from Active Learning with Redundancy

In the main paper, we show the results on CIFAR-10 and MNIST with $10\times$ redundancy. In this section, we also add results for $5\times$ and $15\times$ redundancy for MNIST. The results are in Figure 10. Furthermore, we also run experiments on SVHN (bottom row) with $10\times$ redundancy. The following are the takeaways of the results:

- The CG functions (LOGDETCG and FLCG) significantly outperform other baselines including BADGE, particularly after a few rounds of AL and towards the end. In particular, there is a improvement of 3% to 5% using the CG functions compared to BADGE and other baselines with a labeled set size of 500.
- The main reason for this is that the CG functions pick more unique points compared to the other techniques.
- Amongst the two CG functions, we see that LOGDETCG performs better than FLCG.
- From the pairwise penalty matrix in Figure 11, we see that LOGDETCG has the lowest column sum and has the highest row sum, which indicates that it statistically significantly outperforms other techniques. In terms of the row sum, LOGDETCG is followed by FLCG and BADGE.

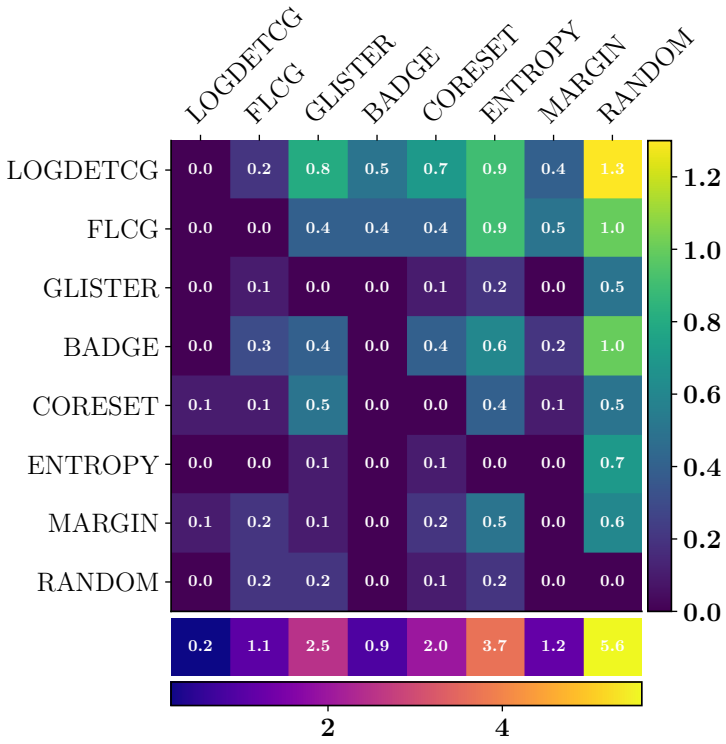

Figure 11: Penalty Matrix comparing the different AL approaches in the redundancy scenario. We observe that the SCG functions have a much lower column sum compared to other approaches.

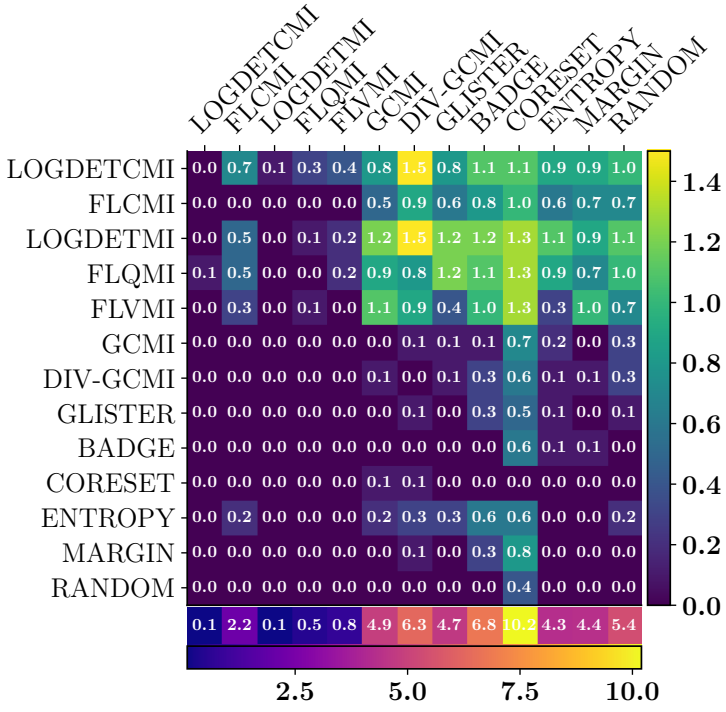

Figure 12: Penalty Matrix comparing the different AL approaches in the OOD Scenario. We observe that the SCMI and SMI functions have a much lower column sum compared to other approaches.

F Additional Experiments and Takeaways for Active Learning with OOD Data

In the case of active learning with OOD data, we additionally add the penalty matrix (figure 12). The following are the main observations and takeaways:

- Figure 12 shows the results of the penalty matrix with the different CMI functions. We observe that LOGDETCMI has the smallest column sum along with LOGDETMI.
- However, as shown in the main paper, the CMI functions have the smallest variance and are hence more stable compared to the SMI variants. Furthermore, the CMI functions generally outperform the SMI counterparts at later rounds.
- However, the SMI functions are often comparable (particularly LOGDETMI and FLQMI) and hence are a good choice for OOD data as well.

G Societal Impacts and Limitations

Limitations of this work: The first limitation of this work is that the MI functions are all graph-based functions. With the exception of FLQMI, all functions have quadratic complexity. The partitioning trick will help, but that comes at the cost of performance. We would like to explore more classes of MI functions (feature-based functions [44] in particular) in future work. Secondly, the MI functions depend on good choices of features. In this work, we use gradients which tend to work very well since they inherently also capture uncertainty [3]. However, the approaches do not perform as well in the early stages, which could be mitigated by the use better features, e.g., self-supervised and unsupervised representations [15].

Societal Impacts: Negative societal impacts of this work include using SIMILAR to mine through large datasets to perpetuate and amplify certain biases in the data. On the flip side, this work can also have a positive impact through its use for fair active learning, where certain under-represented and minority slices or classes can be improved upon by applying it in the rare class and rare slice experiment setting (Sec. 3.2). We would like to explore the use of SIMILAR in applications like improving the performance of biased slices based on race; for example, we would like to improve inference performance on underrepresented Asian woman using SIMILAR for tasks like face recognition, gender recognition, and age recognition. Importantly, recent work has shown that commercial facial recognition and age/gender classification engines perform poorly on these rare slices [7]. A number of recent papers have been proposed to generate such fair face datasets [21], but creating such datasets can take a lot of manual effort to mine the rare slices. We propose to use and study SIMILAR for such scenarios in future work.