Deep Active Learning for Biased Datasets via Fisher Kernel Self-Supervision

Denis Gudovskiy
Panasonic β AI Lab
denis.gudovskiy@us.panasonic.com

Alec Hodgkinson
Panasonic β AI Lab
alec.hodgkinson@us.panasonic.com

Takuya Yamaguchi
Panasonic AI Solutions Center
yamaguchi.takuya2015@jp.panasonic.com

Sotaro Tsukizawa
Panasonic AI Solutions Center
tsukizawa.sotaro@jp.panasonic.com

Abstract

Active learning (AL) aims to minimize labeling efforts for data-demanding deep neural networks (DNNs) by selecting the most representative data points for annotation. However, currently used methods are ill-equipped to deal with biased data. The main motivation of this paper is to consider a realistic setting for pool-based semi-supervised AL, where the unlabeled collection of train data is biased. We theoretically derive an optimal acquisition function for AL in this setting. It can be formulated as distribution shift minimization between unlabeled train data and weakly-labeled validation dataset. To implement such acquisition function, we propose a low-complexity method for feature density matching using self-supervised Fisher kernel (FK) as well as several novel pseudo-label estimators. Our FK-based method outperforms state-of-the-art methods on MNIST, SVHN, and ImageNet classification while requiring only 1/10th of processing. The conducted experiments show at least 40% drop in labeling efforts for the biased class-imbalanced data compared to existing methods[1].

1. Introduction

Active learning (AL) algorithms aim to minimize the number of expensive labels for supervised training of deep neural networks (DNNs) by selecting a subset of relevant examples from a large unlabeled collection of data [20] as sketched in Figure 2. The subset is annotated by an oracle in semi-supervised setting and added to the training dataset in a single *pool* or, more often, in an iterative fashion. The goal is to maximize prediction accuracy while minimizing the pool size and number of iterations.

The existing AL methods assume that distribution of collected train examples is somewhat similar to test cases and, hence, relevant data points can be found only by accessing train data. This assumption rarely holds for the unlabeled data where very *rare examples* have to be identified as illustrated in Figure 1. The classifier learned on train data selected by the existing AL methods can have high error rate on underrepresented instances. For example, distribution of digits "1" prevails over *rare* digits "9" in train data and, as a result, test digits "9" are misclassified. Moreover, consider an autonomous vehicle only trained to perform well in the most frequent conditions rather than in a rare critical situations such as car crashes. To overcome this limitation, we propose a new acquisition function for AL. It is based on distribution matching between the validation dataset and the AL-selected training data. Validation dataset in such setting covers important cases from the long-tail of distribution that can be continuously identified and added after field trials.

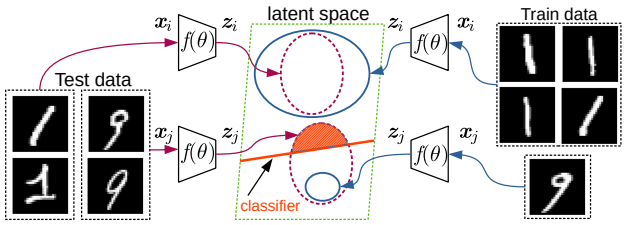

Figure 1. Problem statement for AL with biased data: distribution of unlabeled train data is not aligned with the test data. As a result, prior methods select examples from another distribution and the learned classifier $f(\boldsymbol{x}, \boldsymbol{\theta})$ misses on underrepresented instances.

We achieve distribution matching by pooling multi-scale low-dimensional discriminative features from the *task classifier model*. Our key contribution is the usage of Fisher kernel (FK) to find the most important examples with the improved pseudo-label estimators using several novel metrics. Finally, we incorporate recent unsupervised pretraining method [8] to speed up representation learning by the task model. Our framework is well-suited for large-scale data because its complexity is only a single forward and

[1] Our code is available at github.com/gudovskiy/al-fk-self-supervision

backward pass per data point. We show the effectiveness of our method on MNIST, SVHN, and ImageNet classification including biased training data with long-tailed distribution, where the proposed method is able to decrease labeling efforts by at least 40% compared to prior methods.

2. Related work

AL is a well-studied approach to decrease annotation efforts in traditional machine learning pipelines [27]. Recently, AL has been applied to DNN-based models in semi-supervised setting with oracle labeling or weakly-supervised setting with pseudo-labeling. While our method can be applied to both types, we mainly focus on prior work of a more robust semi-supervised pool-based AL.

Gal *et al*. [7] introduced a measure of uncertainty for approximate Bayesian inference that can be estimated using stochastic forward passes through a DNN with dropout layers. Their AL acquisition function selects data points with the highest uncertainty which is measured at the output of classifier's *softmax* layer using several metrics. Recent work by Beluch *et al*. [5] improved this method by using an ensemble of networks for uncertainty estimation.

Sener and Savarese [26] formulated training dataset selection for AL as a geometric *core-set* clustering approach which outperforms greedy k-center clustering. Though their core-set clustering can complement our approach, we are focusing on a discriminative low-dimensional feature extraction followed by inexpensive clustering. Computational complexity of the core-set clustering is a potential bottleneck where two orders of magnitude more processing is needed compared to greedy clustering in our approach.

Recently, Sinha *et al*. [28] proposed to use variational autoencoder (VAE) [18] to learn a latent space followed by an adversarial network [21] to discriminate between labeled and unlabeled data. Their AL acquisition function is the output of discriminator, which *implicitly* learns the most likely to be labeled examples. This variational adversarial active learning (VAAL) approach claims to achieve superior results compared to all previous works. However, VAAL has large number of hyperparameters and high complexity since VAE and discriminator have to be retrained on all unlabeled and labeled train data every AL iteration.

The closest to our method, line of works [19, 17] employs influence functions and Fisher kernels as a measure of feature importance for dataset subsampling and analysis. Khanna *et al*. [17] showed equivalence of FK and influence functions for log-likelihood loss functions. Similar work on online importance sampling using Fisher score similarity [25] upweights samples within the mini-batch during fully-supervised training. However, these approaches require fully-labeled data to estimate FK.

Another related area is unsupervised representation learning that, unfortunately, has not been used in AL literature. At the same time, recent approaches [8, 11, 13] significantly improved previous state-of-the-art. Hence, we incorporate unsupervised pretraining into our AL method to speed up latent representation learning.

The existing methods struggle to deal with biased data as sketched in Figure 1. Motivated by this, we develop our framework with the following contributions:

- We derive an optimal acquisition function $\mathcal{R}_{opt}(\cdot)$ for biased datasets, which is formulated as a task to minimize Kullback–Leibler (KL) divergence between distributions of training and validation datasets.
- We propose a low-complexity non-parametric AL method via self-supervised FK using a set of pseudo-label estimators and derive its connection to $\mathcal{R}_{opt}(\cdot)$.
- We complement our method by the recent unsupervised pretraining method using image rotations [8].
- Our method outperforms prior methods in image classification. In particular, datasets with long-tailed biased train data result in at least 40% less labeling.

3. Problem statement for biased datasets

Let $(\boldsymbol{x}, \boldsymbol{y})$ be an input-label pair where a label $\boldsymbol{y} = \mathbf{1}_d \in \mathbb{R}^D$ is one-hot vector with only dth class not equal to zero for a classification task. There is a relatively small validation dataset $\mathcal{D}_{\text{v}} = \{(\boldsymbol{x}_i^{\text{v}}, \boldsymbol{y}_i^{\text{v}})\}_{i\in\mathbb{M}}$ of size M and a large collection of training pairs $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i\in\mathbb{N}}$ of size N for which, initially, all labels are unknown. The validation dataset can be weakly labeled as discussed below. At every bth iteration AL acquisition function $\mathcal{R}(\cdot)$ selects a pool of P new labels to be annotated and added to train data which creates a training dataset indexed by subset $\mathbb{N}^b$.

A feed-forward DNN model $f(\boldsymbol{x}, \boldsymbol{\theta})$ is optimized with respect to parameter vector $\boldsymbol{\theta}$ using supervised learning framework by minimizing objective function

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N^b}\sum_{i\in\mathbb{N}^b} L(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i) = \frac{1}{N^b}\sum_{i\in\mathbb{N}^b} L(\boldsymbol{y}_i, f(\boldsymbol{x}_i, \boldsymbol{\theta})), \tag{1}$$

where $L(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i)$ is a loss function and $\hat{\boldsymbol{y}}_i$ is output prediction. The loss function is a negative log probability of discrete $\boldsymbol{y}$ for classification task. This is equivalent to minimization of approximate KL divergence D_{KL} between joint training data distribution $Q_{\boldsymbol{x},\boldsymbol{y}}$ with density $q(\boldsymbol{x}, \boldsymbol{y})$ and the learned model distribution $P_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{\theta})$ with corresponding density $p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})$. Since $q(\boldsymbol{x}, \boldsymbol{y}) = q(\boldsymbol{y}|\boldsymbol{x})q(\boldsymbol{x})$ and $p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})q(\boldsymbol{x})$, KL objective learns only *conditional distribution* of $\boldsymbol{y}$ given $\boldsymbol{x}$ as

$$D_{KL}(Q_{\boldsymbol{x},\boldsymbol{y}} \| P_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{\theta})) = \int q(\boldsymbol{x}) \int q(\boldsymbol{y}|\boldsymbol{x}) \log \frac{q(\boldsymbol{y}|\boldsymbol{x})q(\boldsymbol{x})}{p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})q(\boldsymbol{x})} d\boldsymbol{y} d\boldsymbol{x} = \mathbb{E}_{Q_{\boldsymbol{x}}}[D_{KL}(Q_{\boldsymbol{y}|\boldsymbol{x}} \| P_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{\theta}))]. \tag{2}$$

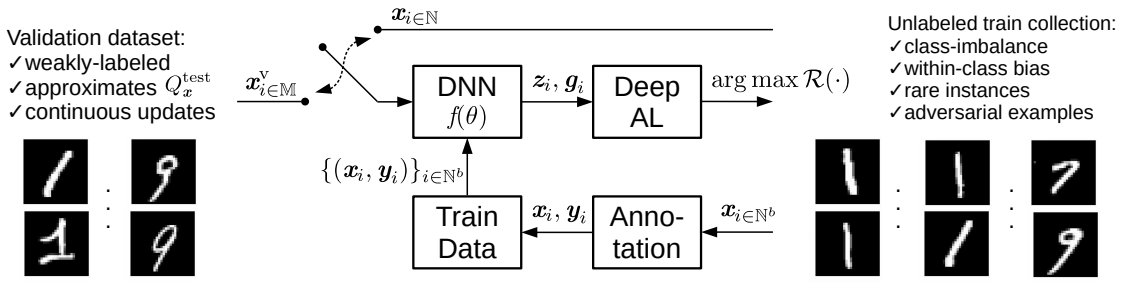

Figure 2. General setup for our semi-supervised AL: validation dataset is selected to approximate test data and can be continuously updated by the newly discovered misclassified examples. Unlabeled collection of train data is subject to the specified distortions. AL algorithm finds relevant train examples for annotation by maximizing acquisition function $\mathcal{R}(\cdot)$ every bth iteration.

Due to unknown density $q(\boldsymbol{x})$, the expectation over $Q_{\boldsymbol{x}}$ in (2) is usually replaced by *empirical distribution* $\hat{Q}_{\boldsymbol{x}}$ as

$$\mathbb{E}_{\hat{Q}_{\boldsymbol{x}}}[D_{KL}(Q_{\boldsymbol{y}|\boldsymbol{x}}\|P_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{\theta}))] = \frac{1}{|\mathcal{D}|}\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}}[D_{KL}(Q_{\boldsymbol{y}|\boldsymbol{x}}\|P_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{\theta}))]. \tag{3}$$

By rewriting loss $L(\cdot)$ in (1) using D_{KL} from (3), objective function $\mathcal{L}(\boldsymbol{\theta})$ can be rewritten as negative log of conditional probability

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{N^b}\sum\nolimits_{i\in\mathbb{N}^b}\log p(\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{\theta}). \tag{4}$$

However, the *actual task* is to minimize objective (2) for test data $\mathcal{D}_{\text{test}}$ with expectation over $Q_{\boldsymbol{x}}^{\text{test}}$ distribution. This contradiction is usually resolved in AL literature by assuming $Q_{\boldsymbol{x}}^{\text{test}}$ and $Q_{\boldsymbol{x}}$ equality. In practice, the deployed systems struggle to deal with underrepresented test cases in the train distribution $Q_{\boldsymbol{x}}$. The examples include autonomous vehicles in rare traffic situations or facial recognition systems with gender and race biases [6]. This is schematically illustrated in Figure 1.

We argue that the *key requirement* for effective AL *in the wild* is to collect a validation dataset $\mathcal{D}_{\text{v}}$ with distribution $Q_{\boldsymbol{x}}^{\text{v}}$, which approximates $Q_{\boldsymbol{x}}^{\text{test}}$. To be specific, we approximate distribution of a representative collection of *test cases* in $\mathcal{D}_{\text{v}}$ and continuously update it by newly discovered misclassified data. This can be done iteratively after conducting field trials for deployed systems. The assumptions about $\mathcal{D}_{\text{v}}$ and $\mathcal{D}$ are summarized in Figure 2.

It follows from (2) that an optimal acquisition function $\mathcal{R}_{opt}(\cdot)$ for AL minimizes distribution shift between $\mathcal{D}_{\text{test}}$ and $\mathcal{D}$, where the former is approximated by empirical $\mathcal{D}_{\text{v}}$. This can be expressed using KL divergence as

$$\mathcal{R}_{opt}(b,P) = \arg\min_{\mathcal{R}(b,P)} D_{KL}(Q_{\boldsymbol{x}}^{\text{test}}\|Q_{\boldsymbol{x}}) \approx \arg\min_{\mathcal{R}(b,P)} D_{KL}(\hat{Q}_{\boldsymbol{x}}^{\text{v}}\|\hat{Q}_{\boldsymbol{x}}), \tag{5}$$

where, in practice, (5) can be replaced by locally optimal steps for every iteration $b = 1\dots B$ and pool size P.

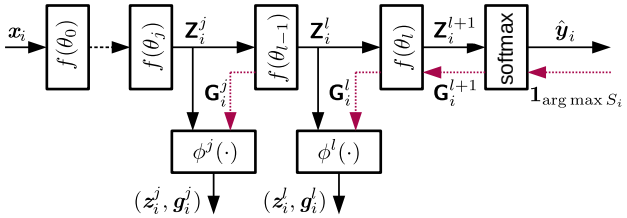

Figure 3. Conventional multi-scale feature extraction and the proposed FK extension (dashed). Descriptors $\boldsymbol{z}_i$ and Fisher score vectors $\boldsymbol{g}_i$ are used for density matching by our AL method.

4. The proposed method

4.1. Conventional feature descriptors for AL

High dimensionality of input $\boldsymbol{x}$ causes computational difficulties in minimizing (5). Then, $\boldsymbol{x}$ is usually replaced by a low-dimensional feature *descriptor* in image retrieval [30]. Such descriptors are pooled from DNN intermediate representations $\boldsymbol{z}$, which are found to be effective [3]. Then, (5) can be reformulated as empirical distribution matching between $\hat{P}_{\boldsymbol{z}}^{\text{v}}$ and $\hat{P}_{\boldsymbol{z}}$. This can be done using various methods [10], but, practically, a greedy k-center clustering for density estimation with a similarity measure is the most used method for the large train dataset size N.

Let $\mathbf{Z}_i^j \in \mathbb{R}^{C\times H\times W}$ be the output of jth layer of *task DNN model* for input image $\boldsymbol{x}_i$ as shown in Figure 3 for image classification, where C, H, and W are the number of channels, the height, and the width, respectively. Then, a feature vector or descriptor of length L can be defined as $\boldsymbol{z}_i = \phi(\mathbf{Z}_i) \in \mathbb{R}^L$, where function $\phi(\cdot)$ is a conventional average pooling operation. In a multi-scale case, descriptor $\boldsymbol{z}_i$ is a list of multiple feature vectors $\boldsymbol{z}_i^j$.

A descriptor matrix for the validation dataset $\boldsymbol{Z}_{\text{v}} \in \mathbb{R}^{L\times M}$ and training dataset $\boldsymbol{Z} \in \mathbb{R}^{L\times N}$ can be efficiently calculated using DNN forward passes. Practically, descriptors can be further compressed for storage efficiency reasons using PCA, quantization, etc. Pearson correlation (PCC) is a common match kernel, which is an accurate measure of *linear* correlation. By preprocessing vectors $\boldsymbol{z}_i$

to have zero mean and unit variance, the similarity (cross-covariance) matrix for multi-scale case is simply

$$\boldsymbol{R}_{\boldsymbol{z}} = \sum\nolimits_j (\boldsymbol{Z}_{\text{v}}^j)^T \boldsymbol{Z}^j. \tag{6}$$

Using information theory [9], this framework assumes representation $\boldsymbol{z}$ to have the following properties about the task: *minimality* ($\min I(\boldsymbol{z};\boldsymbol{x})$) and *sufficiency* ($I(\boldsymbol{y};\boldsymbol{z}) = I(\boldsymbol{y};\boldsymbol{x})$), where $I()$ is mutual information quantity. Indeed, Achille *et al*. [1] analytically shows that a DNN trained by stochastic gradient descent (SGD) discards non-informative features and retains only the ones to minimize objective function (2). However, these properties are applicable only for a fully trained model without bias in train data.

An alternative approach is to use an autoencoder [12] or, similarly to VAAL [28], probabilistic VAE [18] to compress $\boldsymbol{x}$ to $\boldsymbol{z}$. Those alternatives require to train another model using a new set of hyperparameters and reconstruction loss rather than task-specific objective (2). However, the learned representation $\boldsymbol{z}$ is subject to biased train data pitfall shown in Figure 1. Fortunately, this pitfall can be resolved for *task model* by AL itself, if it minimizes distribution shift in (5). Hence, we choose to pool features $\boldsymbol{z}$ from the *task model* in our framework to avoid data bias, additional complexity, and hyperparameter search issues. We address the sufficiency property discussed above by using unsupervised pretraining followed by a more powerful match kernel.

4.2. Self-supervised Fisher kernel

Recent works [19, 17] revived interest in influence functions and Fisher kernels used in pre-DNN era [24]. They are able to identify the most influential training points for a given test data. Though attractive, these methods are computationally expensive for large-scale data and DNN models because FK is typically calculated with respect to high-dimensional parameter vector $\boldsymbol{\theta}$.

Using the sufficiency property [1], we approximate our optimal acquisition function (5) using the distributions of learned representations $\boldsymbol{z}$ as

$$\mathcal{R}_{opt}(b, P) = \underset{\mathcal{R}(b,P)}{\arg\min}\, D_{KL}(\hat{P}_{\boldsymbol{z}}^{\text{v}} \| \hat{P}_{\boldsymbol{z}}), \tag{7}$$

Then, a *connection* between the main task (2) and $D_{KL}(P_{\boldsymbol{z}}^{\text{v}} \| P_{\boldsymbol{z}})$ minimization in (7) via Fisher information can be derived with respect to small perturbations in $\boldsymbol{\theta}$. Assuming that the task model minimizes distribution shift in (2) every backward pass as

$$p^{\text{v}}(\boldsymbol{z}|\boldsymbol{\theta}) = p(\boldsymbol{z}|\boldsymbol{\theta}) + \Delta p, \tag{8}$$

where $\Delta p = \Delta\boldsymbol{\theta}\partial p(\boldsymbol{z}|\boldsymbol{\theta})/\partial\boldsymbol{\theta}$ and $\Delta \to 0$.

By substituting (8), the expanded form of (7) can be simplified using Taylor series of natural logarithm as

$$\mathcal{R}_{opt}(b, P) \approx \underset{\mathcal{R}(b,P)}{\arg\min}\, \Delta\boldsymbol{\theta}^T \mathcal{I} \Delta\boldsymbol{\theta}, \tag{9}$$

where $\mathcal{I} = \mathbb{E}_{P_{\boldsymbol{z}}}\left[\boldsymbol{g}(\boldsymbol{\theta})\boldsymbol{g}(\boldsymbol{\theta})^T\right]$ is a Fisher information matrix and $\boldsymbol{g}(\boldsymbol{\theta}) = \partial \log p(\boldsymbol{z}|\boldsymbol{\theta})/\partial\boldsymbol{\theta}$ is a Fisher score with respect to $\boldsymbol{\theta}$. The detailed derivation is given in Appendix.

Using result in (9), Jaakkola and Hauusler [15] proposed the popular Fisher kernel expressed by

$$R_{z,g}(\boldsymbol{z}_m, \boldsymbol{z}_n) = \boldsymbol{g}_m(\boldsymbol{\theta})^T \mathcal{I}^{-1} \boldsymbol{g}_n(\boldsymbol{\theta}). \tag{10}$$

To make (10) computationally tractable, we use *practical* FK (PFK) where $\mathcal{I}^{-1}$ is replaced by identity matrix. Such a common approach decreases quadratic storage requirements. Next, we rewrite Fisher scores $\boldsymbol{g}_i(\boldsymbol{\theta})$ using a more compact form $\boldsymbol{g}_i(\boldsymbol{\theta}) = \text{vec}(\boldsymbol{g}_i \boldsymbol{z}_i^T)$, where $\boldsymbol{g}_i$ is computed with respect to features as $\boldsymbol{g}_i = \partial L(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i)/\partial \tilde{\boldsymbol{z}}_i$, $L(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i)$ is log-likelihood loss function from (4), and $\tilde{\boldsymbol{z}}_i$ is a vector before applying nonlinearity $\sigma(\cdot)$. The latter follows from the chain rule when computing loss function for a DNN layers ($\tilde{\boldsymbol{z}}_i^j = \boldsymbol{\theta}^T \boldsymbol{z}_i^j = \boldsymbol{\theta}^T \sigma(\tilde{\boldsymbol{z}}_i^{j-1})$) as derived in Appendix. Then, the tractable PFK can be rewritten for DNNs as

$$R_{z,g}(\boldsymbol{z}_m, \boldsymbol{z}_n) = \boldsymbol{g}_m(\boldsymbol{\theta})^T \boldsymbol{g}_n(\boldsymbol{\theta}) = \boldsymbol{z}_m^T \boldsymbol{z}_n \boldsymbol{g}_m^T \boldsymbol{g}_n. \tag{11}$$

Fisher scores in (11) are also related to visual explanation methods [22]. If replace $\boldsymbol{z}$ by $\boldsymbol{x}$ in $\boldsymbol{g}(\boldsymbol{\theta})$ calculation, the result estimates popular importance heatmaps in the input space. In our case, kernel (11) shows the model sensitivity to changes in parameters caused by distribution shift $D_{KL}(\hat{P}_{\boldsymbol{z}}^{\text{v}} \| \hat{P}_{\boldsymbol{z}})$. Then, PFK matrix $\boldsymbol{R}_{\boldsymbol{z},\boldsymbol{g}} \in \mathbb{R}^{M \times N}$ can be efficiently calculated using a series of forward-backward passes. By analogy to feature similarity (6), the Fisher scores $\boldsymbol{g}_i$ for images are calculated with respect to tensors $\mathbf{Z}_i$ and pooled by the same $\phi(\cdot)$ such that $\boldsymbol{g}_i = \phi(\partial L_i/\partial \mathbf{Z}_i) \in \mathbb{R}^L$. Finally, we minimize the distribution shift in (7) by maximizing PFK as

$$\mathcal{R}_{opt}(b, P) = \underset{\mathcal{R}(b,P)}{\arg\max}\, \boldsymbol{R}_{\boldsymbol{z},\boldsymbol{g}}, \tag{12}$$

where $\boldsymbol{R}_{\boldsymbol{z},\boldsymbol{g}} = \boldsymbol{R}_{\boldsymbol{z}} \circ \boldsymbol{R}_{\boldsymbol{g}} = \sum_j (\boldsymbol{Z}_{\text{v}}^j)^T \boldsymbol{Z}^j \circ (\boldsymbol{G}_{\text{v}}^j)^T \boldsymbol{G}^j$. Our PFK matrix $\boldsymbol{R}_{\boldsymbol{z},\boldsymbol{g}}$ is an element-wise multiplication of feature similarity from (6) and gradient similarity matrices.

4.3. The proposed pseudo-label estimators

The main drawback of (12) is lack of labels $\boldsymbol{y}$ in the unlabeled collection of train data. The common pseudo-labeling ($\mathbf{1}_{\arg\max_d S}$) metric $S(\cdot)$ assigns hard-label to the dth class with maximum predicted probability: $S = \hat{\boldsymbol{y}}$. That leads to incorrect estimates during first AL iterations, particularly, for rare examples. To overcome this limitation, we propose several novel metrics to estimate pseudo-labels.

First, we introduce estimation metrics using Monte Carlo (MC) sampling. Consider a DNN input $\boldsymbol{x}$ being sampled near its local neighborhood. That produces inputs $\boldsymbol{x}_k$, feature samples $\boldsymbol{z}_k$, and a corresponding per-class Fisher scores $\boldsymbol{g}_k(d) = \partial L(\mathbf{1}_d, \hat{\boldsymbol{y}}_k)/\partial \boldsymbol{z}_k$, where class $d = 1 \dots D$.

The sampling can include small rotations, translations or color distortions for image inputs [4]. The simplest MC label estimation maximizes linear correlation between features and Fisher scores as $S = \mathrm{tr}\,(\boldsymbol{C}_{\boldsymbol{z},\boldsymbol{g}})$, where $\boldsymbol{C}_{\boldsymbol{z},\boldsymbol{g}}$ is cross-covariance matrix between feature descriptors and Fisher scores. Theoretically, a better metric is maximization of mutual information $I(\boldsymbol{z};\boldsymbol{g})$ to capture nonlinear dependency. Classic result [9] shows that for random vectors $\boldsymbol{z}$ and $\boldsymbol{g}$ that follow Gaussian probability model, average mutual information can be estimated as $S = I(\boldsymbol{z};\boldsymbol{g}) = 0.5 \log\left(|\boldsymbol{C}_{\boldsymbol{z},\boldsymbol{z}}|\,|\boldsymbol{C}_{\boldsymbol{g},\boldsymbol{g}}| \,/\, |\boldsymbol{C}_{\boldsymbol{z}\boldsymbol{g},\boldsymbol{z}\boldsymbol{g}}|\right)$, where $|\boldsymbol{C}|$ is the determinant of cross-covariance matrix. This can be efficiently calculated using LU or Cholesky decomposition implemented in modern ML frameworks [23].

The second proposed metric explicitly estimates $\hat{p}(\boldsymbol{y},\boldsymbol{z}) = \hat{p}(\boldsymbol{y}|\boldsymbol{z})p(\boldsymbol{z})$, for which it is necessary to have a trusted annotated dataset to obtain $\hat{p}(\boldsymbol{y}|\boldsymbol{z})$. In our case, it can be validation dataset $\mathcal{D}_\mathrm{v}$ or its subset. Since $p(\boldsymbol{y}|\boldsymbol{z}) = p^\mathrm{v}(\boldsymbol{y}|\boldsymbol{z})$, the estimate $\hat{p}(\boldsymbol{y},\boldsymbol{z})$ can be found from trusted conditional density $p^\mathrm{v}(\boldsymbol{y}|\boldsymbol{z})$ and marginal $p(\boldsymbol{z})$. We propose to reuse the described above framework to find the most similar data points in $\mathcal{D}_\mathrm{v}$ to examples in $\mathcal{D}$ using $\boldsymbol{R}_{\boldsymbol{z}}$ kernel. Then, we assign given trusted labels $\boldsymbol{y}^\mathrm{v}$ from $p^\mathrm{v}(\boldsymbol{z})$ to train labels from $p(\boldsymbol{z})$ for which $\boldsymbol{R}_{\boldsymbol{z}}$ is maximized. This results in a low-complexity non-parametric method.

To summarize, we experiment with the following label estimation metrics: a) $S = \boldsymbol{y}$ for ablation study with true labels, b) common $S = \hat{\boldsymbol{y}}$, as well as the proposed c) MC $S = \mathrm{tr}\,(\boldsymbol{C}_{\boldsymbol{z},\boldsymbol{g}})$, d) MC $S = I(\boldsymbol{z};\boldsymbol{g})$ and e) $S = \hat{p}(\boldsymbol{y},\boldsymbol{z})$.

4.4. Complexity of weakly-supervised algorithm

While FK finds the most similar data points using discriminative representation, our AL needs to identify validation points for distribution matching using (12). However, even inexpensive greedy k-center clustering might be prohibitive ($\mathcal{O}(PM)$) for relatively small $\mathcal{D}_\mathrm{v}$. To address this, we propose to use *weak supervision* (correct or incorrect prediction) to find subset of misclassified validation examples $\{\mathbf{1}_{\arg\max_d \hat{\boldsymbol{y}}_i^\mathrm{v}} \neq \boldsymbol{y}_i^\mathrm{v}\}_{i\in\acute{\mathbb{M}}}$, where $\acute{M} < M$. Then, this subset is clustered using k-centers, and P validation points are selected to maximize PFK in (12). Weak supervision assumption typically holds because, often, $\mathcal{D}_\mathrm{v}$ is already fully-labeled to know how model is performing. Variant of our weakly-supervised method is fully described in Alg. 1.

Computational complexity of PFK is estimated in Table 1 in terms of forward and backward DNN passes. Note that the complexity of greedy clustering, finding cross-covariance matrices is not shown because it is negligible compared to DNN passes. For comparison with AL phase (lines 3-10 in Alg. 1), we report complexity of retraining phase (line 11) using I epochs and N^b labeled train data.

Since the number of unlabeled data $\acute{N}^b$ ($\acute{N}^b = N - N^{b-1}$ in line 8) is much bigger than validation data M, our method is $EK/2$ times less complex than uncertainty methods [7, 5] with K stochastic passes and E ensembles.

Algorithm 1 Variant with weakly-supervised $\mathcal{D}_\mathrm{v}$.

1: **Initialize:** $\mathbb{N}^0 = \{\}$, $\boldsymbol{\theta}^0$ random or pretrained by [8]
2: **for** $b = 1, 2 \ldots B$ **do**
3: find misclassified subset $\{\mathbf{1}_{\arg\max_d \hat{\boldsymbol{y}}_i^\mathrm{v}} \neq \boldsymbol{y}_i^\mathrm{v}\}_{i\in\acute{\mathbb{M}}}$
4: pool matrices $(\boldsymbol{Z}_\mathrm{v}, \boldsymbol{G}_\mathrm{v}) \in \mathbb{R}^{L\times\acute{M}}$
5: **if** $\acute{M} > P$ **then**
6: find P centers in $\acute{\mathbb{M}}$ using k-center clustering
7: subsample matrices $(\boldsymbol{Z}_\mathrm{v}, \boldsymbol{G}_\mathrm{v}) \in \mathbb{R}^{L\times P}$
8: pool matrices $(\boldsymbol{Z}, \boldsymbol{G}) \in \mathbb{R}^{L\times\acute{N}^b}$, $\acute{N}^b = N - N^{b-1}$
9: calculate PFK matrix $\boldsymbol{R}_{\boldsymbol{z},\boldsymbol{g}} = \boldsymbol{R}_{\boldsymbol{z}} \circ \boldsymbol{R}_{\boldsymbol{g}}$
10: add P points to $\mathbb{N}^b$ as $\arg\max_p \boldsymbol{R}_{\boldsymbol{z},\boldsymbol{g}}$
11: update $\boldsymbol{\theta}^b = \arg\min_{\boldsymbol{\theta}} \sum_{i\in\mathbb{N}^b} L(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i)/N^b$

Table 1. Complexity estimates per AL iteration. Assuming $\acute{N}^b >> M$, our method has the lowest complexity in terms of forward and backward DNN passes during AL phase.

Method	AL	Train
Uncert. [7]	$K\acute{N}^b$	$2IN^b$
Ens. uncert. [5]	$EK\acute{N}^b$	$2EIN^b$
VAAL [28]	$\acute{N}^b + 2NI_{\text{VAE,D}}$	$2IN^b$
PCC (6): $\boldsymbol{R}_{\boldsymbol{z}}$	$M + \acute{N}^b$	$2IN^b$
PFK (12): $\boldsymbol{R}_{\boldsymbol{z},\boldsymbol{g}}$ (**ours**)	$2(M + \acute{N}^b)$	$2IN^b$
PFK_MC(12): $\boldsymbol{R}_{\boldsymbol{z},\boldsymbol{g}}$ (**ours**)	$KD(M + \acute{N}^b)$	$2IN^b$

VAAL [28] consists of sampling phase with $\acute{N}^b$ forward passes and retraining phase of VAE and discriminator models using $I_{\text{VAE,D}}$ epochs. Assuming that VAE, discriminator and task model $f(\boldsymbol{x},\boldsymbol{\theta})$ have roughly the same complexity, our method is $I_{\text{VAE,D}}$ times less complex than VAAL.

The method with PCC kernel (6) is $2\times$ less complex than ours with PFK. The variant of our method with MC pseudo-labeling ($S = \mathrm{tr}\,(\boldsymbol{C}_{\boldsymbol{z},\boldsymbol{g}})$ or $I(\boldsymbol{z};\boldsymbol{g})$) is $KD/2$ times more complex than PFK with inexpensive metrics ($S = \hat{\boldsymbol{y}}$ or $\hat{p}(\boldsymbol{y},\boldsymbol{z})$), where D is number of classes. MC metrics have potentially better accuracy compared to $S = \hat{\boldsymbol{y}}$ without reliance on a trusted labeled dataset as in $S = \hat{p}(\boldsymbol{y},\boldsymbol{z})$.

5. Experiments

We apply our framework to MNIST, SVHN and ImageNet classification. We evaluate AL not only with the original training data, but also their biased versions. Hence, we introduce a *class imbalance* which scales down number of available train images for subset of classes. Class imbalance is defined as the ratio of $\{0 \ldots 4\}$ digits to $\{5 \ldots 9\}$ digits for MNIST and SVHN. We randomly select 500 out of 1,000 classes for ImageNet. Train examples for the selected 500 classes are decimated by the class imbalance ratio, while the other 500 classes keep the original train data.

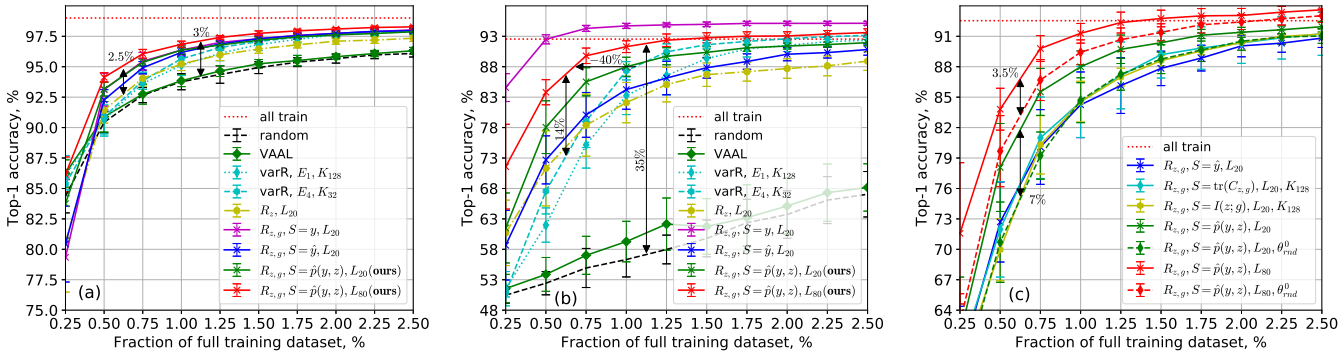

Figure 4. MNIST test accuracy: (a) no class imbalance, (b) $100\times$ class imbalance, and (c) ablation study of pseudo-labeling and unsupervised pretraining ($100\times$ class imbalance). Our method decreases labeling by 40% compared to prior works for biased data.

The code is written in PyTorch [23] with reproducible experiments and is publicly available.

The following experimental configurations are defined: baseline when *all train* data is used, *random* sampling, and methods from Table 1. We reimplemented all uncertainty methods [7, 5]: variation ratio (*varR*), maximum entropy and BALD. Only results of the best-performing varR method are reported. We use official code for VAAL [28] experiments. We use the following notation in figures: number of ensembles is specified by the E, samples by K, and descriptor size by L.

We run each experiment $10\times$ for MNIST, $5\times$ for SVHN and once for large-scale ImageNet on V100 GPUs. We report mean accuracy and standard deviation for MNIST and SVHN test dataset. Due to lack of test labels for ImageNet, we use validation dataset for testing. Each AL experiment consists of 10 iterations ($B = 10$). With the exception of last fully-connected layer, initial network parameters are from unsupervised pretraining using rotation method [8] or, if specified, randomly initialized. Large batch sizes may underperform with class-imbalanced data and, therefore, we select mini-batch size by cross-validation. The used DNN models are LeNet, ResNet-10, and ResNet-18 for MNIST, SVHN, and ImageNet, respectively. The dropout configurations are the same or similar to [7, 5] setups.

5.1. MNIST

The dataset split $|\mathcal{D}|$, $|\mathcal{D}_\mathrm{v}|$ and $|\mathcal{D}_\mathrm{test}|$ has 50, 10 and 10 thousand images, respectively. The following hyperparameters are used: SGD, epochs=50, batch-size=25, lr=0.05, lr-decay=0.1 every 15 epochs. Descriptor length L is 20 for single-scale (after *conv2* output) and 80 for three-scale descriptor (*conv1,2* and *fc1* outputs). The selected pool size P is 125 images or 0.25% of $|\mathcal{D}|$.

Figure 4(a) shows the case when the unlabeled train dataset approximates test distribution. In this setting, the uncertainty method varR performs relatively well with only 2.5% decrease in accuracy compared to our best method ($\boldsymbol{R}_{\boldsymbol{z},\boldsymbol{g}}$, $S = \hat{p}(\boldsymbol{y}, \boldsymbol{z})$, L_{80}) at first iterations and almost on par when $b > 5$. Random sampling accuracy is only 3% lower due to nearly uniform train distribution. VAAL [28] results are similar to random sampling.

A practical case with $100\times$ class imbalance is illustrated in Figure 4(b). Our FK-based methods from (12) outperform PCC feature-only method from (6) with the increase of descriptors size L and use of a better label estimation metric: $S = \hat{p}(\boldsymbol{y}, \boldsymbol{z})$ vs. common $S = \hat{\boldsymbol{y}}$. The gap between the best FK and the best uncertainty method with ensembles reaches 14% or, equivalently, 40% less labels is needed for the same accuracy. Furthermore, our method requires $EK/2 = 64\times$ less processing according to Table 1.

As part of ablation study, we plot in Figure 4(a,b) a FK setup with all-true labels ($S = \boldsymbol{y}$). It shows the *theoretical limit* of FK: no accuracy is gained without class-imbalance, while significant (3-10%) improvement is achieved with the data bias compared to pseudo-labeling using $S = \hat{p}(\boldsymbol{y}, \boldsymbol{z})$. In fact, such setup *exceeds* performance of the full train dataset accuracy at the second AL iteration. Task model pretrained by rotation method is able to separate digits without supervision with exception of the last randomly initialized fully-connected layer. Hence, a single AL iteration is needed to achieve baseline result.

A set of ablation studies is presented in Figure 4(c). First, unsupervised pretraining using rotations [8] adds 7% in accuracy when $L = 20$ and 3.5% when $L = 80$ compared to random-weight initialization ($\boldsymbol{\theta}^0_{rnd}$). Second, we compare pseudo-label estimation metrics proposed in Section 4.3. The common $S = \hat{\boldsymbol{y}}$ metric performs only 1% inferior compared to MC metrics ($S = \mathrm{tr}\,(\boldsymbol{C}_{\boldsymbol{z},\boldsymbol{g}})$ and $S = I(\boldsymbol{z}; \boldsymbol{g})$) when $b > 4$, while it requires $KD/2\times$ less processing. In our setup, MC metrics employ uniform $\pm 5^\circ$ image rotations and Gaussian additive noise for sampling. They may require larger K, other sampling or go beyond the Gaussian assumption to achieve better results. For example, Kay *et al*. [16] show a tractable solution for elliptically symmetric probability model and Bachman *et al*. [4] propose to measure mutual information across multiple scales of features. Our best metric with $S = \hat{p}(\boldsymbol{y}, \boldsymbol{z})$ outperforms others by

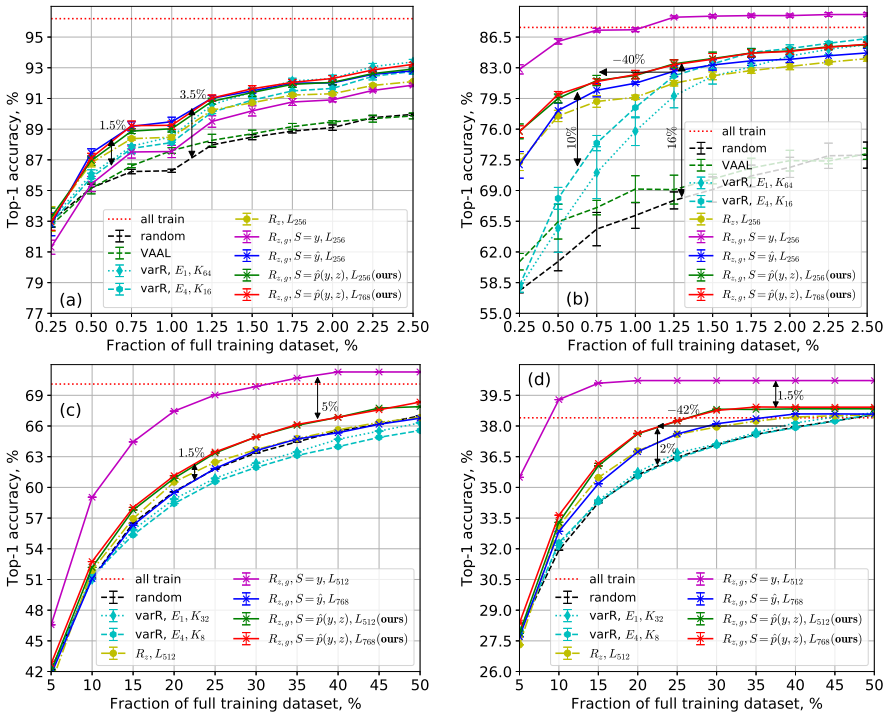

Figure 5. SVHN test (top) and ImageNet val (bottom) accuracy: (a,c) no class imbalance and (b,d) with 100× class imbalance.

6-7%. Therefore, we conclude that $\boldsymbol{R}_{\boldsymbol{z},\boldsymbol{g}}$ with $S = \hat{p}(\boldsymbol{y}, \boldsymbol{z})$ is a preferable approach.

5.2. SVHN

The dataset split $|\mathcal{D}|$, $|\mathcal{D}_{\text{v}}|$ and $|\mathcal{D}_{\text{test}}|$ contains 500, 104 and 26 thousand images, respectively. Training dataset is obtained from concatenation of the original *train* and *extra train* datasets with total of 604,388 images. The following hyperparameters are used: SGD, epochs=35, batch-size=128, lr=0.1, lr-decay=0.1 every 15 epochs. Descriptor length L is 256 for single-scale (*resblock3* output) and 768 for two-scale descriptor (*resblock3,4* outputs). The selected pool size P is 1,250 images or 0.25% of $|\mathcal{D}|$.

The gap between random sampling and our method is 3.5% for the original and 16% for the biased SVHN with the same amount of training data in Figures 5(a,b). Uncertainty varR method lacks 1.5% and 10% in accuracy compared to ours during first AL iterations and perform on par when $b > 4$. Hence, approximately 40% of labeling can be avoided for the biased train data. Moreover, computational complexity of uncertainty methods is 32× higher.

The method with PCC ($\boldsymbol{R}_{\boldsymbol{z}}$) in Figure 5(b) achieves 2% and 4% less accuracy compared to PFK ($\boldsymbol{R}_{\boldsymbol{z},\boldsymbol{g}}$) with the simplest pseudo-label estimation metric ($S = \hat{\boldsymbol{y}}$) and our best metric $S = \hat{p}(\boldsymbol{y}, \boldsymbol{z})$, respectively.

The larger descriptor size L does not significantly improve accuracy in this setup. This points to importance of multi-scale extraction when, for example, spatially-localized features can be more relevant than global ones or vice versa. A parametric aggregation of feature hierarchy can lead to better results [2, 14]. The latter is not trivial without labeled data, unlike our non-parametric approach.

5.3. ImageNet

The original dataset split $|\mathcal{D}|$ and $|\mathcal{D}_{\text{v}}|$ has 1,200 and 50 thousand images, respectively. The following hyperparameters are used: SGD, epochs=60, batch-size=128, lr=0.1, lr-decay=0.1 at [30, 50, 57] epoch. The descriptor configuration is the same as for SVHN. The selected pool size P is 64,000 images or 5% of $|\mathcal{D}|$.

Figures 5(c,d) show results for large-scale ImageNet. Uncertainty varR method underperforms without class imbalance and only a fraction of percent better than random sampling with 100× class imbalance. This could be related to lower number of samples K compared to setup in [5], dropout setting heuristics or large number of classes. Unfortunately, it is almost infeasible to increase K due to high complexity of varR, which is 16× more than for our method during AL phase and E× more during retraining. For instance, the ImageNet experiment took 2.5 days for our method and 12 days for varR on a single V100 GPU.

Our best method ($\boldsymbol{R}_{\boldsymbol{z},\boldsymbol{g}}$, $S = \hat{p}(\boldsymbol{y}, \boldsymbol{z})$, L_{768}) increases accuracy compared to prior works by 1.5% without class imbalance and by 2% with 100× class imbalance. The configurations with the simplest pseudo-label estimation metric ($S = \hat{\boldsymbol{y}}$) or the ones without FK supervision gain only 1%

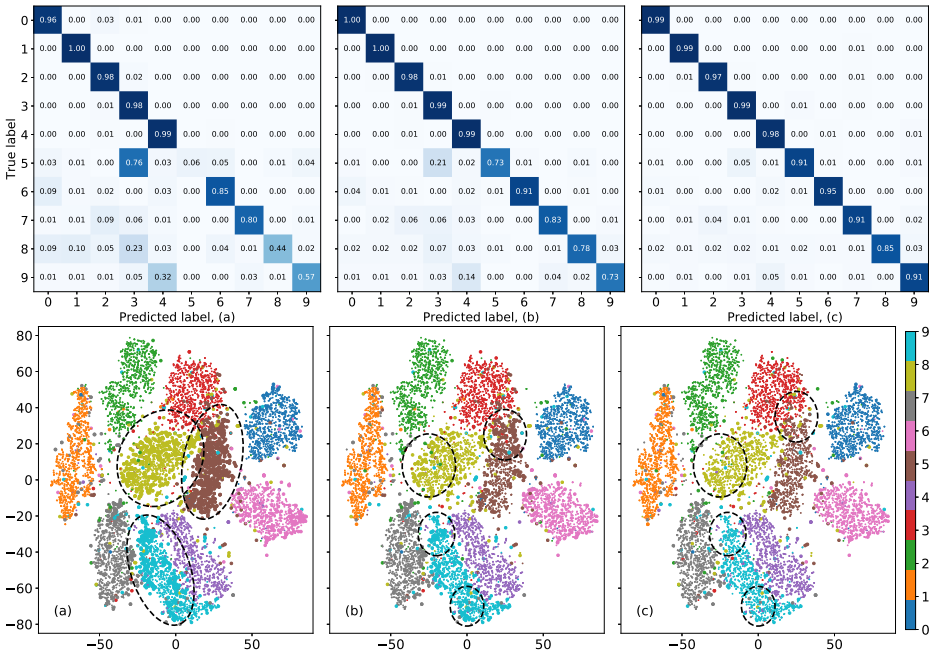

Figure 6. Confusion matrix (top) and t-SNE (bottom) of MNIST test data at AL iteration $b = 3$ with 100× class imbalance for: (a) varR with E_1, K_{128}, (b) $\boldsymbol{R}_{\boldsymbol{z},\boldsymbol{g}}$, $S = \hat{p}(\boldsymbol{y}, \boldsymbol{z})$, L_{80} (**ours**), and (c) $\boldsymbol{R}_{\boldsymbol{z},\boldsymbol{g}}$, $S = y$, L_{80}. Dots and balls represent correspondingly correctly and incorrectly classified images for t-SNE visualizations. The underrepresented classes $\{5, 8, 9\}$ have on average 36% accuracy for prior work (a), while our method (b) increases their accuracy to 75%. The ablation configuration (c) shows 89% theoretical limit of our method.

in accuracy. The gap between theoretically possible ImageNet result with true labels ($S = \boldsymbol{y}$) and our method with the estimated pseudo-labels is increasing compared to relatively small-scale 10-class MNIST in Figures 4(a,b) and SVHN in Figures 5(a,b). It indicates that a more accurate pseudo-label metric may improve results even more. While our absolute accuracy improvement is 2%, it leads to 42% less annotations with the same accuracy.

5.4. Qualitative visualizations

To demonstrate improvement of AL behavior, we calculate confusion matrices and t-SNE [29] clusters. We use the same experimental setup as in Figure 4(b) with class imbalance ratio of 100 and analyze MNIST test dataset after the third AL iteration ($b = 3$). Figure 6 presents results for the following configurations: (a) varR (E_1, K_{128}) and the proposed ($\boldsymbol{R}_{\boldsymbol{z},\boldsymbol{g}}, L_{80}$) with (b) pseudo-labels ($S = \hat{p}(\boldsymbol{y}, \boldsymbol{z})$) and (c) true-labels ($S = \boldsymbol{y}$) for ablation study.

The class-imbalanced digits $\{5 \dots 9\}$ are heavily misclassified in Figure 6(a). It visually confirms quantitative result from Section 5.1 that uncertainty methods fail to identify relevant training data clusters. Those methods can only capture so called *epistemic* uncertainty which is uncertainty over DNN parameters instead of uncertainty about data.

Figures 6(b,c) show results of the FK-supervised methods with the estimated pseudo-labels and true-labels. Compared to Figure 6(a) the class-imbalanced digits are significantly better classified, specifically, the centers of clusters "5", "8" and "9", whose average accuracy increased from only 36% to 75%. This result indicates the ability of self-supervised FK to find long-tails of distribution using our acquisition function (12).

The far edges of the imbalanced clusters that intersect with other digit clusters still experience some irregular densities of misclassified examples in Figure 6(b) due to imperfect pseudo-labeling. The t-SNE setup with all-true labels in Figure 6(c) improves on those edges and achieves 89% accuracy. Clearly, it is the most difficult to separate very similar intersecting examples from different classes. As a potential future direction, this problem might be addressed by a better feature separation or using adversarial training.

6. Conclusions

We formulated the optimal acquisition function for AL with realistic assumptions about data biases and continuous updates after field trials. We introduced low-complexity non-parametric AL method that minimizes distribution shift between train and validation datasets using self-supervised FK and several novel pseudo-label estimators. According to ablation studies, unsupervised pretraining further improved our approach. The conducted image classification experiments showed that our method results in at least 40% less labeling for biased data compared to prior works while requiring a factor of 10 less processing.

References

[1] Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, pages 1947–1980, 2018.

[2] Relja Arandjelović, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5297–5307, 2016.

[3] Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1269–1277, 2015.

[4] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, 2019.

[5] William H. Beluch, Tim Genewein, Andreas Nürnberger, and Jan M. Kohler. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9368–9377, 2018.

[6] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of the Conference on Fairness, Accountability and Transparency*, pages 77–91, 2018.

[7] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1183–1192, 2017.

[8] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[9] Robert M. Gray. *Entropy and Information Theory*. Springer-Verlag, 1990.

[10] A. Gretton, K. Borgwardt, M. Rasch, B. Scholkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, pages 723–773, 2012.

[11] Olivier J. Henaff, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv:1905.09272*, 2019.

[12] Geoffrey E Hinton and Richard S. Zemel. Autoencoders, minimum description length and Helmholtz free energy. In *Proceedings of the 6th Conference on Neural Information Processing Systems*, pages 3–10, 1994.

[13] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[14] Syed Husain and Miroslaw Bober. Remap: Multi-layer entropy-guided pooling of dense CNN features for image retrieval. *IEEE Transactions on Image Processing*, pages 5201–5213, 2019.

[15] Tommi Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. In M. J. Kearns, S. A. Solla, and D. A. Cohn, editors, *Proceedings of the 11th Conference on Neural Information Processing Systems*, pages 487–493, 1999.

[16] Jim Kay. Feature discovery under contextual supervision using mutual information. In *Proceedings of the International Joint Conference on Neural Networks*, 1992.

[17] Rajiv Khanna, Been Kim, Joydeep Ghosh, and Oluwasanmi Koyejo. Interpreting black box predictions using Fisher kernels. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 3382–3390, 2019.

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

[19] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1885–1894, 2017.

[20] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, 1994.

[21] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. *arXiv:1511.05644*, 2016.

[22] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, pages 211–222, 2017.

[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *Autodiff workshop at Advances in Neural Information Processing Systems*, 2017.

[24] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[25] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 4334–4343, 2018.

[26] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[27] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin–Madison, 2010.

[28] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[29] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, pages 2579–2605, 2008.

[30] Wengang Zhou, Houqiang Li, and Qi Tian. Recent advances in content-based image retrieval: A literature survey. *arXiv:1706.06064*, 2017.