Feature Distribution Matching by Optimal Transport for Effective and Robust Coreset Selection

Weiwei Xiao¹,², Yongyong Chen¹, Qiben Shan², Yaowei Wang², Jingyong Su¹,²*
¹School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen
²Pengcheng laboratory, Shenzhen
xiaowwhit@gmail.com, YongyongChen.cn@gmail.com, {shanqb,wangyw}@pcl.ac.cn, sujingyong@hit.edu.cn

Abstract

Training neural networks with good generalization requires large computational costs in many deep learning methods due to large-scale datasets and over-parameterized models. Despite the emergence of a number of coreset selection methods to reduce the computational costs, the problem of coreset distribution bias, i.e., the skewed distribution between the coreset and the entire dataset, has not been well studied. In this paper, we find that the closer the feature distribution of the coreset is to that of the entire dataset, the better the generalization performance of the coreset, particularly under extreme pruning. This motivates us to propose a simple yet effective method for coreset selection to alleviate the distribution bias between the coreset and the entire dataset, called feature distribution matching (FDMat). Unlike gradient-based methods, which selects samples with larger gradient values or approximates gradient values of the entire dataset, FDMat aims to select the coreset that is closest to the feature distribution of the entire dataset. Specifically, FDMat transfers coreset selection as an optimal transport problem from the coreset to the entire dataset in feature embedding spaces. Moreover, our method shows strong robustness due to the removal of samples far from the distribution, especially for the entire dataset containing noisy and class-imbalanced samples. Extensive experiments on multiple benchmarks show that FDMat can improve the performance of coreset selection than existing coreset methods. The code is available at https://github.com/successhaha/FDMat.

Introduction

Recently, foundation models in the visual and multi-modal domains have achieved unprecedented success (Ouyang et al. 2022). However, these high-performing models rely on hundreds of millions of data for training, resulting in significant computational costs. To explore the potential of achieving comparable performance for models on smaller dataset, there has been a growing interest in coreset selection methods. The objective of the coreset selection is to identify a subset of samples that serves as an effectively representative of the entire dataset.

To identify samples that contribute to model learning, gradient-based methods (Killamsetty et al. 2021b; Paul, Ganguli, and Dziugaite 2021) typically assess the contribution of individual samples according to gradient values or gradient matching method (Killamsetty et al. 2021a) finds the coreset that is closest to the gradients of the entire dataset. Although gradient-based methods offer sufficient theoretical proofs, robustness is rarely guaranteed for datasets containing gradient-abnormal noise samples, due to the reliance on gradient judgments. As a result, gradient-based methods tend to exclude some samples with a high gradient to minimize perturbations from noise samples (Paul, Ganguli, and Dziugaite 2021). If the diversity of samples in coreset is not enough to represent the entire dataset or contains outlier samples that deviate from the true distribution, the coreset exhibits distribution bias, also known as sample selection bias (Cortes et al. 2008). Furthermore, the performance of the gradient-based method deteriorates significantly under extreme pruning conditions (such as retaining less than 40% of training samples) due to insufficient diversity of samples (Killamsetty et al. 2021b; Paul, Ganguli, and Dziugaite 2021).

In this paper, we delve into the distribution bias of coreset selection, which refers to the bias of feature distribution in feature embedding space. Intuitively, the coreset is a highly condensed version of the entire dataset, and its distribution should closely resemble that of the entire dataset. When the distribution of the coreset deviates significantly from that of the entire dataset, the generalization performance of the classifier may suffer. These phenomena are shown in Fig. 1(a). The distribution bias of the coreset could cause the decision boundary of the model to deviate, which in turn affects the generalization ability of the model on testing samples.

To validate our intuition, we randomly choose 5 classes from the MINIST (Lecun and Bottou 1998) as the same class and train a binary classification network. Then we train the network using two coresets: randomly sampling 10% and skwed sampling 10% of the entire dataset. In Fig. 1(b), we visualize the feature distribution of coresets and the entire dataset, and then evaluate the accuracy of coresets with different feature distributions. As expected, the less distribution bias and the more similar distribution between the coreset and the entire dataset in feature embedding spaces, the higher test accuracy of the coreset.

Our method is simple yet effective and exhibits strong robustness, especially when the entire dataset contains class-
imbanced and noisy samples, which removes samples far away from the distribution. Our contributions are summarized as follows:

- We propose an efficient coreset method to discover important samples by alleviating the distribution bias between the coreset and the entire dataset through optimal transport in deep learning scenarios.
- We propose the supremum of maximum mean discrepancy (MMD) between coreset and entire dataset can be solved by minimizing 1-Wasserstein distance in feature embedding space.
- Extensive experiments show that FDMat outperforms existing methods under extreme pruning and noise label datasets, which guarantees closest distribution of coreset to the entire dataset by feature distribution matching.

**Related Work**

Most current foundation models are computationally expensive due to training on large datasets, resulting in increased energy costs. As a result, there has been a growing interest in improving data-efficiency. In this paper, we primarily focus on coreset selection methods to reduce computational costs. Initially introduced in computational geometry (Agarwal et al. 2005), coresets have been quickly adopted later by machine learning community to address classical problems, such as Bayesian inference (Huggins, Campbell, and Broderick 2016), K-means (Har-Peled and Mazumdar 2004) and more.

The traditional coreset methods can be roughly divided into two groups, one is sensitivity-based important sampling coresets, and the other is distribution-based coresets under certain conditions. Sensitivity-based methods, such as the k-clustering problems take into account the importance of samples by approximate probability (Bachem, Lucic, and Lattanzi 2018; Bateni et al. 2014). Distribution-based methods typically require consideration of the underlying data distribution, such as designing the coreset based on Reproducing Kernel Hilbert Space (RKHS) theory (Chen, Welling, and Smola 2012) or utilizing the integral probability metric in the context of optimal transport theory (Claici, Genevay, and Solomon 2018). However, these traditional coreset methods (Feldman, Faulkner, and Krause 2011; Bachem, Lucic, and Krause 2015; Zhang et al. 2023) face challenges due to their high computational complexity and reliance on fixed data representations (which are seldom suited for image data). As a result, doubts have arisen regarding their effectiveness in deep learning scenarios.

With the increasing prominence of data-driven methods, coreset selection has emerged as a focal point of research in deep learning scenarios. In this context, coreset selection can be roughly categorized into two main groups: gradient-based and decision boundary-based methods.

**Gradient-based** methods aim to find the coreset in which the gradient aligns with minimal error to the gradient of the entire dataset, thereby ensuring that the model exhibits comparable generalization performance on the coreset. Some recent works include Craig (Mirzasoleiman, Bilmes, and Leskovec 2020), GradMatch (Killamsetty et al. 2021a), Glister (Killamsetty et al. 2021b), AdaCore (Pooladzandi, Davini, and Mirzasoleiman 2022), LCMat (Shin et al. 2023), etc. Craig (Mirzasoleiman, Bilmes, and Leskovec 2020) identifies the coreset by formulating gradient matching as an optimization problem of a monotonic submodular function (Fujishige 2005) under a first-order gradient error bound. Similar to Craig, GradMatch (Killamsetty et al. 2021a) incorporates L2 regularization on the basis of first-order error to reduce reliance on specific samples, and leverages orthogonal matching pursuit (OMP) (Elenberg et al. 2016) to select samples. Curr (Mirzasoleiman, Cao, and Leskovec 2020) extends the coreset selection method to handle noisy label scenarios while maintaining the first-order error. Building upon first-order error, AdaCore (Pooladzandi, Davini, and Mirzasoleiman 2022) extends the gradient error range to second-order, resulting in improved gradient matching through Hessian matrix computations. LCMat (Shin et al. 2023) further refines the second-order gradient error by introducing sharpness-aware minimization (Foret P. 2020). Gradient-based methods have a strong theoretical basis, and their convergence has been analyzed. However, the diversity of samples in coreset under extreme pruning conditions is not well guaranteed, which ignores the relationship between samples.

**Decision boundary-based** methods consider that samples that are difficult to distinguish near the decision boundary are beneficial to construct the coreset. Deepfool (Moosavi-Dezfooli, Fawzi, and Frossard 2016) constructs coreset based on the minimum perturbation required for samples, which applying perturbation until the predicted labels change. Similarly, Cal (Margatina et al. 2021)
constructs the coreset by comparing the maximum deviation between the predicted likelihood of samples and their neighbors. Additionally, some methods implicitly use decision boundaries to select coreset in feature embedding spaces. Herding (Welling 2009) constructs the coreset using a distance-based approach that greedily chooses samples to minimize the distance between the centers of the coreset and the entire dataset in feature space. K-Center Greedy (Sener and Savarese 2017) further defines coreset selection as minimizing the maximum distance between the coreset and the entire dataset. However, these methods are slower to run on models with over-parameters, some methods use a proxy model to speed up the process of coreset selection (Coleman et al. 2020; Sachdeva, Wu, and McAuley 2021). SVP (Coleman et al. 2020) accelerates the coreset selection process by using a lighter model as a proxy, and selecting coreset on the proxy model using metrics such as decision boundary (margin) or uncertainty. Although the gradient-based method has sufficient theoretical support, it is susceptible to noise samples and labels, leading to reduced robustness of the coreset. On the other hand, the decision boundary-based method can find samples far away from the distribution, but it lacks sufficient theoretical support. Therefore, we revisit the traditional distribution-based coreset methods (Claici, Genevay, and Solomon 2018; Chen, Welling, and Smola 2012) and design FDmat based on optimized transport theory for deep learning scenarios. In contrast to (Claici, Genevay, and Solomon 2018), we re-define the maximum mean discrepancy with the 1-Lipschitz constraint instead of RKHS, enabling it to be applied to feature space for deep learning methods. We then formulate it as a dual problem to solve for the 1-Wasserstein distance. Additionally, we expedite the computation of the 1-Wasserstein distance using the Sinkhorn (Cuturi 2013) method instead of stochastic gradient descent.

**Methodology**

In this section, we introduce the coreset selection preliminaries and the details of our method.

**Preliminaries**

We focus on dataset selection for classification task, which is a widely studied scenario in machine learning community. We are given a training set $\mathcal{U} = \{(x_i, y_i)\}_{i=1}^n$ and a coreset $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^m$ from unknown Borel probability distribution $\mathcal{P}$ and $\mathcal{Q}$, where $x_i \in \mathcal{X}$ represents the input, $y_i \in \mathcal{Y}$ represents the label of $x_i$, $m$ and $n$ are sample numbers. Let $f_\theta$ be the feature extractor and $f_\theta^S$ be a classifier of $c$ classes. The goal of coreset selection is to find the most representative coreset $S$ with the constraint $S \subset \mathcal{U}$, so that the model $\theta^S$ trained on $S$ has closer generalization performance to the model $\theta^U$ trained on the entire dataset $\mathcal{U}$.

**Feature-Distribution Matching (FDMat)**

As shown in Fig. 1(b), the more similar distribution between the coreset and the entire dataset, the higher accuracy of the coreset. Therefore, we devote to developing a coreset method that has a distribution closest to that of the entire dataset while maintaining strong robustness. In this section, we introduce the optimal objective of feature distribution matching and prove its supremum. Additionally, we elaborate how to select a coreset by iteratively solving feature distribution matching.

**Optimal Objective of Feature Distribution Matching**

This section introduces an objective for coreset selection, called FDMat, which matches the feature distribution of $\mathcal{U}$ and $\mathcal{S}$ based on feature extractor $f_\theta$ pre-trained with $\mathcal{U}$. The discrepancy between the feature distribution of $\mathcal{U}$ and $\mathcal{S}$ is our primary focus.

**Definition 3.1** Let the shorthand notation $E_{x \sim \mathcal{P}}[f_\theta(x)]$ and $E_{x \sim \mathcal{Q}}[f_\theta(x)]$ to denote expectation with respect to $\mathcal{P}$ and $\mathcal{Q}$, respectively.

The goal of feature distribution matching is to minimize the maximum mean discrepancy between the coreset $\mathcal{S}$ and the entire dataset $\mathcal{U}$ in feature embedding spaces. Following the optimization scheme, we formulate the primary objective as follows:

$$\min_{\mathcal{S}} \max_{\mathcal{U}} (E_{x \sim \mathcal{P}}[f_\theta(x)] - E_{x \sim \mathcal{Q}}[f_\theta(x)])$$  (1)

In Eq. (1), our goal is first to find the supremum of the maximum mean discrepancy between $\mathcal{U}$ and $\mathcal{S}$, and then minimize this supremum on coreset $\mathcal{S}$.

**Theorem 3.2** If $f_\theta$ is the 1-Lipschitz function, then the supremum of MMD can be obtained from the Kantorovich-Rubinstein duality (Hörmander, Totaro, and Waldschmidt 2006). (Proof in Appendix A.1 of the supplementary material)

$$W_1(\mathcal{P}, \mathcal{Q}) = \sup_{\|f_\theta\|_1 \leq 1} (E_{x \sim \mathcal{P}}[f_\theta(x)] - E_{x \sim \mathcal{Q}}[f_\theta(x)])$$  (2)

where $W_1$ is Wasserstein distance. Therefore, the goal can be defined as the optimal transport problem: finding a mapping route that minimizes the cost of distributed transmission from the coreset $\mathcal{S}$ to the entire dataset $\mathcal{U}$.

**Wasserstein distance** Let $\mathcal{P} = \sum_{k=1}^n p_i \delta_{x_i}, \mathcal{Q} = \sum_{k=1}^m q_i \delta_{u_i}$ be two discrete distributions, described by their supports $(S_i)_{i=1}^n \subset \mathcal{R}^{n \times d}$ and $(U_i)_{i=1}^m \subset \mathcal{R}^{m \times d}$ and weight vectors $p \in \Delta_n$ and $q \in \Delta_m$.

According to Theorem 3.2, we get the supremum of MMD. Therefore, the optimal solution can be defined as the case where Wasserstein distance corresponds to the first-order ground cost.

$$W_1(\mathcal{P}, \mathcal{Q}) = \min_{P \in \mathcal{U}(p,q)} \langle P, M \rangle = \sum_{i,j} P_{i,j} M_{i,j}$$  (3)

where $\mathcal{U}(p,q) = \{P \in \mathcal{R}^{n \times m} : P_{1\cdot} = p, P_{\cdot 1} = q \}$ represents the mapping transport from distribution $p$ to distribution $q$. $P_{1\cdot} = \{\sum_{i,j} P_{i,j}\}$ is $\mathcal{R}^{n \times m}$ and $P_{\cdot 1} = \{\sum_{i,j} P_{i,j}\}$ is $\mathcal{R}^{m}$ are the matrix-vectors, $M = (|S_i - U_j|)_{i,j} \in \mathcal{R}^{n \times m}$ is the matrix of pairwise Euclidean distances between the supports, $m$ and $n$ are sample numbers.
Algorithm 1: FDMat: Feature Distribution Matching

**Input:** Training data $\mathcal{U} = \{x_i, y_i\}^n$, coreset size $k$, feature extractor $f_\theta$, class $w$, hyperparameter $\lambda$

**Output:** coreset $\mathcal{S}$

**Training Variables:** feature extractor $f_\theta$ pre-trained on $\mathcal{U}$

for $(x_i, y_i) \in \mathcal{U}$ do
  Transform $x_i$ with Tukey’s Ladder of Powers by Eq. (4)
  Approximate feature distribution $c_j$ by Eq. (5)
  Calculate the distance matrix $M_{ij}$ by Eq. (6)
end

Initialize edge distribution $: p \leftarrow 1/n$, $q \leftarrow |c_j|/w$

Optimal transport distance:

$$d^\lambda_{M_{ij}}(p, q) = \text{Sinkhorn}(M(p, q, \lambda))$$  \hspace{1cm} (8)

where $H(P) = -\sum_{i,j} P_{i,j}(\log(P_{i,j}) - 1)$ represents the entropy regularization, $\lambda > 0$ represents the regularization parameter used to control the degree of entropy regularization. Sinkhorn is used to iteratively solve Eq. (7) as follows:

$$d^\lambda_{M_{ij}}(p, q) = \text{Sinkhorn}(M(p, q, \lambda))$$  \hspace{1cm} (8)

Through Sinkhorn algorithm (Cuturi 2013), we get the $W_1$ distance of optimal transport.

The entire algorithm is shown in Algorithm 1. FDMat mainly contains three steps:

- Step 1: Using Tukey’s Ladder to reduce the skewed distribution and then approximate the distribution of entire dataset $\mathcal{U}$.
- Step 2: Calculating the optimal transport distance from the training samples to the approximate distribution of entire dataset $\mathcal{U}$ by Sinkhorn.
- Step 3: Obtaining the coreset $\mathcal{S}$ based on the size of the coreset and the transport cost of samples.

Since FDMat selects the coreset that is closest to the class center, it can effectively eliminate noise samples that far away from the distribution.

### Experiments

This section examines the effectiveness and robustness of FDMat method through various experiments.

### Implementation Details and Baselines

**Implementation details** Most current coreset selection methods have been evaluated in different experimental settings, such as datasets, model architectures, coreset sizes, augmentations, training strategies. These differences may lead to unfair comparisons between different methods and unconvincing results. To investigate the effectiveness of coreset selection methods under a fair and unified framework, we follow the coreset selection scenario of these methods (Shin et al. 2023; Guo, Zhao, and Bai 2022).

We evaluate FDMat on three widely-used datasets and one medical dataset for coreset selection: CIFAR10, CIFAR100 (Krizhevsky and Hinton 2009), Tiny-ImageNet (Russakovsky et al. 2015) and Path-MNIST (Yang, Shi, and Ni 2021). Datasets with varying levels of granularity, such as CIFAR100 and Tiny-ImageNet which respectively contain 100 and 200 classes of samples, may exhibit different feature spatial distributions (Liang and Zou 2022). The objective of our evaluation is to show the effectiveness and universality of our method across datasets with different distributions.

**Baselines** To demonstrate the effectiveness of FDMat under extreme pruning conditions, we compare various coreset methods in recent years. The comparison baselines can be divided into two primary categories: the first category is based on the forward-output of the model, including layer feature vector and softmax output, such as C-Div (Agarwal et al. 2020), Herding (Welling 2009), K-Center (Sener and Savarese 2017), Least Confidence (Coleman et al. 2020), Entropy (Coleman et al. 2020), and Margin (Coleman et al. 2020).
Table 1: The accuracy of coreset selection methods on CIFAR-10 and CIFAR-100 using ResNet-18 over 5 different random seeds. The best and second-best results for each setting are highlighted in bold and underlined, respectively.

<table>
<thead>
<tr>
<th>Fraction</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>100%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>77.45±1.0</td>
<td>87.36±0.4</td>
<td>90.67±0.2</td>
<td>92.10±0.3</td>
<td>34.09±2.4</td>
<td>55.98±0.7</td>
<td>64.59±0.1</td>
<td>68.65±0.1</td>
<td>78.91±0.2</td>
<td>80.29±2.1</td>
</tr>
<tr>
<td>C-Div</td>
<td>56.85±1.7</td>
<td>81.30±2.5</td>
<td>90.93±0.5</td>
<td>93.30±0.4</td>
<td>20.53±0.6</td>
<td>44.91±1.1</td>
<td>58.60±2.7</td>
<td>66.21±1.5</td>
<td>59.21±0.4</td>
<td>32.23±0.5</td>
</tr>
<tr>
<td>Herding</td>
<td>63.04±2.5</td>
<td>74.10±2.5</td>
<td>79.93±1.5</td>
<td>85.20±0.9</td>
<td>26.47±0.2</td>
<td>42.83±1.9</td>
<td>52.14±1.4</td>
<td>60.99±0.5</td>
<td>24.07±0.4</td>
<td>44.42±1.4</td>
</tr>
<tr>
<td>K-Center</td>
<td>72.12±1.7</td>
<td>87.10±0.3</td>
<td>90.83±0.3</td>
<td>92.80±0.1</td>
<td>27.37±1.5</td>
<td>52.10±0.8</td>
<td>63.74±0.7</td>
<td>68.88±0.2</td>
<td>17.63±2.1</td>
<td>41.29±1.1</td>
</tr>
<tr>
<td>L-Conf</td>
<td>58.43±3.0</td>
<td>81.90±2.2</td>
<td>91.21±0.1</td>
<td>93.10±0.5</td>
<td>16.94±0.9</td>
<td>41.88±1.3</td>
<td>57.45±2.0</td>
<td>63.74±0.4</td>
<td>59.45±2.2</td>
<td>65.56±0.3</td>
</tr>
<tr>
<td>Entropy</td>
<td>57.45±3.6</td>
<td>81.90±0.4</td>
<td>91.06±0.7</td>
<td>93.20±0.2</td>
<td>36.66±1.0</td>
<td>56.66±0.6</td>
<td>64.81±0.9</td>
<td>68.30±1.1</td>
<td>38.25±1.2</td>
<td>57.73±1.6</td>
</tr>
<tr>
<td>Margin</td>
<td>59.90±6.7</td>
<td>81.70±3.2</td>
<td>90.92±0.4</td>
<td>92.90±0.2</td>
<td>20.70±1.1</td>
<td>46.36±2.7</td>
<td>59.45±2.2</td>
<td>66.56±0.3</td>
<td>78.91±0.2</td>
<td></td>
</tr>
</tbody>
</table>

Evaluation on Benchmarks

We first evaluate the performance of the coreset under different pruning ratios on CIFAR10 and CIFAR100. Table 1 reports accuracy of various coreset methods with ResNet-18 (He et al. 2016). Interestingly, the results in Table 1 show that the random selection achieves competitive performance compared to most existing coreset selection methods under extreme pruning (10% − 30%), indicating weakness in the robustness of current methods. However, FDMat performs significant improvement over other coreset methods, as it guarantees the similarity of distribution between the coreset and the entire dataset. Notably, our FDMat achieves 3.7% and 4.3% improvement over the second-best method in terms of accuracy on CIFAR10 and CIFAR100, respectively, at the 10% pruning ratio in Table 1. Moreover, to evaluate the performance of coreset methods with a larger number of classes, we conduct experiments on Tiny-ImageNet, and the results are presented in Table 2. Table 2 shows that FDMat consistently outperforms other coreset selection baselines on Tiny-ImageNet. It should be noted that most existing coreset selection methods tend to perform poorly when faced with large datasets containing a great number of classes, such as Craig, Glisters and GradMatch. To some extent, this experiment indicates that the existing coreset methods are weak in generalization and robustness.

Finally, we also evaluate the effectiveness of FDMat on the unnatural dataset, medical image dataset (Yang, Shi, and Ni 2021). Table 2 shows that FDMat method still has a good performance than other methods.

Robustness Analysis

In this section, we conduct robustness experiments of coreset methods in various situations including mislabeled samples, class imbalance samples, cross-architectures, and pre-trained feature extractors.

Mislabeled Samples Similar to (Killamsetty et al. 2021b), we artificially construct a mislabeled samples dataset on CIFAR10 to investigate the impact of mislabeled sam-

Figure 2: The performance of different coreset selection methods on the entire dataset $U$ with (a) mislabeled samples and (b) class-imbalanced samples.
samples. Fig. 2(a) reports the experimental results under mislabeled dataset. The results show that the performance of the gradient-based methods significantly decreases under extreme pruning, due to the abnormal gradient disturbance caused by the presence of noisy samples. In contrast, FDMat outperforms the other coreset methods by a significant margin, particularly under extreme pruning of a 10% fraction, where the performance is nearly 50% better than that of the second-best coreset method. Moreover, we find that FDMat can outperform the entire mislabeled dataset using only 60% of samples. This indicates that the gradient-based coreset selection methods could suffer from significant selection bias in the presence of noisy samples, whereas FDMat can effectively eliminate noisy samples that deviate from the underlying distribution. This experiment verifies the robustness of FDMat method from the empirical level to some extent.

Class imbalance samples We then study the effectiveness of coreset methods under class imbalance on CIFAR10. To this end, we manually conduct the class-imbalanced experiments, randomly selected three classes and save 50% of samples on CIFAR10. The experimental results are presented in Fig. 2(b). We observe that selecting a coreset using a feature extractor trained on a class-imbalanced dataset leads to more severe sample selection bias. Even effective random selection method exhibits significant performance degradation in multiple scenarios. However, our FDMat significantly outperforms other coreset methods. Overall, our results suggest that ensuring the similarity of distributions is more effective than considering the magnitude of gradient value when dealing with class-imbalanced datasets.

Robustness on cross-architecture We also study the generalization performance of FDMat under cross-architecture. Specifically, we use the ResNet-18 to select coreset samples, and then train the coreset samples on other architectures, including WRN-16-8 (Zagoruyko and Komodakis 2016), VGG-16 (Simonyan and Zisserman 2015) and Inception-v3 (Szegedy et al. 2016). Table 3 shows that the data selected by FDMat consistently achieve improved or competitive performance across different network structures compared with other coreset methods. Moreover, this experiment also reveals a certain degree of consistency in the performance of coreset samples selected by different architectures.

Robustness on the pre-training We also evaluate the robustness of the coreset methods under different training hyperparameters. Specifically, we conduct multiple experiments with combinations of selected epochs [1, 5, 10, 15, 20]; weight decay [1e-4, 5e-4, 1e-3]; optimizers [SGD, Adam]; and 3 seeds, which result in a total of 90 cases on Tiny-ImageNet. Fig. 3 shows the number of times that each method beats the others in these experiments. Notably, FDMat is superior to other methods in most cases and exhibits strong robustness across different training parameters.

Continual Learning with Memory Replay Memory-based continual learning methods store small representative instances and optimize their classifiers using the
samples stored in memory to alleviate catastrophic forgetting of previously observed tasks. As an application, we use coreset $S$ as a memory exemplar for previously seen classes under the class incremental setting of the method (Zhao, Mopuri, and Bilen 2020). In this setting, CIFAR-100 is divided into 5 sets of sub-classes, with a memory budget of 50 images per class. Each set of classes represents a separate task stage. The model is trained purely based on the latest memory at each task stage. Fig. 4 shows that FDMat outperforms existing coreset methods under memory replay setting, which represents that the sample selected by FDMat has an anti-catastrophic forgetting effect to a certain extent.

**Efficiency Analysis**

In Table 4, the selection efficiency of various coreset methods on CIFAR10 is shown. It can be seen from the experimental results that FDMat has certain advantages in selection efficiency compared with the method using submodular function, Gradmatch, Glíster, Adacore, LCMat-s. The AdaCore and LCMat-S, which leverage second-order gradients and involve the computation of Hessian matrix, exhibit longer selection times. GradMatch employs an OMP algorithm with substantial computing consumption, resulting in an extremely time-consuming process. Our FDMat method strikes a balance between performance and efficiency, offering a reliable compromise between the two factors.

**Conclusion**

We proposed a novel objective for coreset selection, called FDMat, which aims to reduce the maximum mean discrepancy between coreset and entire dataset in feature embedding spaces. By ensuring that the coreset distribution is closest to the entire dataset, FDMat significantly outperforms existing coreset methods under extreme pruning. Notably, FDMat exhibits strong robustness in handling noise and class-imbalanced samples by eliminating samples far from the distribution. Moreover, FDMat shows clear performance merits for continual learning.

<table>
<thead>
<tr>
<th>Fraction</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Herding</td>
<td>15.88</td>
<td>17.65</td>
<td>22.14</td>
</tr>
<tr>
<td>K-Center</td>
<td>48.52</td>
<td>49.11</td>
<td>54.04</td>
</tr>
<tr>
<td>L-Conf</td>
<td>22.51</td>
<td>22.55</td>
<td>22.68</td>
</tr>
<tr>
<td>Entropy</td>
<td>22.69</td>
<td>23.56</td>
<td>24.28</td>
</tr>
<tr>
<td>Margin</td>
<td>22.08</td>
<td>22.21</td>
<td>22.26</td>
</tr>
<tr>
<td>Craig</td>
<td>59.39</td>
<td>61.88</td>
<td>65.56</td>
</tr>
<tr>
<td>GradMatch</td>
<td>1395.73</td>
<td>2157.49</td>
<td>4085.69</td>
</tr>
<tr>
<td>Glíster</td>
<td>98.35</td>
<td>172.84</td>
<td>394.85</td>
</tr>
<tr>
<td>Adacore</td>
<td>762.08</td>
<td>1407.62</td>
<td>2150.39</td>
</tr>
<tr>
<td>LCMat-S</td>
<td>803.45</td>
<td>1543.99</td>
<td>2259.23</td>
</tr>
<tr>
<td>FDMat</td>
<td>31.96</td>
<td>33.70</td>
<td>44.04</td>
</tr>
</tbody>
</table>

Table 4: Selection time analyses of coreset methods on CIFAR10(seconds)
Acknowledgments
This work was supported by National Natural Science Foundation of China under Grants 62376068 and 62106063, by the Shenzhen Science and Technology Innovation Program under Grant JCYJ20220818102414031.

References
PMLR.


9204