BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 06/13 pp1060–1074
DOI: 10.26599/BDMA.2025.9020011
Volume 8, Number 5, October 2025

SPOT: An Active Learning Algorithm for Efficient Deep Neural Network Training

Luyang Fang, Cheng Meng, Lin Zhao, Tao Wang, Tianming Liu, Wenxuan Zhong*, and Ping Ma*

Abstract: Recent advancements in deep neural networks heavily rely on large-scale labeled datasets. However, acquiring annotations for large datasets can be challenging due to annotation constraints. Active learning offers a promising solution to this problem by selectively labeling a small, strategically chosen subset of the unlabeled dataset. However, current active learning methods struggle with data that are unevenly distributed, which leads to the selection of subsets that fail to represent the entire dataset. To overcome this challenge, we introduce a novel active learning algorithm that integrates SPace-filling (SP) designs with the Optimal Transport (OT) technique (SPOT). SPOT technique utilizes optimal transport to effectively manage data from complex manifolds by mapping them to a uniformly distributed hypercube. Additionally, the space-filling design ensures a better asymptotic convergence rate, ensuring that the selected subset encompasses the entire dataset more effectively than other sampling methods, such as random sampling. Our extensive experiments across various image datasets and models demonstrate the superiority of SPOT over existing baselines.

Key words: Active Learning (AL); Optimal Transport (OT); SPace-filling (SP); sampling; deep learning

1 Introduction

Deep Neural Networks (DNNs) have achieved significant advancements in various domains, including image recognition and natural language processing[1–6]. The training of these large-scale DNNs typically requires extensive labeled datasets. For example, the optimization of the Vision Transformer relies on the JFT-300M dataset, which contains 303 million labeled samples[2]. However, acquiring annotations for such extensive training sets presents challenges due to cost, privacy, and the need for specialized expertise[7–9]. Active Learning (AL) has recently emerged as a promising strategy to address these challenges by efficiently selecting a subset from the unlabeled pool for annotation, thereby optimizing the construction of training datasets[10–13]. Unlike random sampling, which regards all data points as equally important, AL assumes some unlabeled data points are more critical for model optimization. The goal is to develop a learning algorithm that can identify and select these pivotal data points for annotation.

Current AL strategies for DNNs fall into two primary categories: uncertainty-based and diversity-based methods. Uncertainty-based methods focus on querying data points with high uncertainty, yet they risk selecting similar or duplicate samples. Conversely, diversity-based methods aim to encompass a

• Luyang Fang, Tao Wang, Wenxuan Zhong, and Ping Ma are with Department of Statistics, University of Georgia, Athens, GA 30602, USA. E-mail: luyang.fang@uga.edu; tw95546@uga.edu; wenxuan@uga.edu; pingma@uga.edu.
• Cheng Meng is with Institute of Statistics and Big Data, Renmin University of China, Beijing 100080, China. E-mail: chengmeng@ruc.edu.cn.
• Lin Zhao and Tianming Liu are with Department of Computer Science, University of Georgia, Athens, GA 30602, USA. E-mail: lin.zhao@uga.edu; tliu@uga.edu.
* To whom correspondence should be addressed.
Manuscript received: 2024-06-16 ; revised: 2024-12-27; accepted: 2025-01-20

comprehensive range of the sample space by selecting data points that maximize diversity based on their distances. A notable approach within this category is to select a subset from a coreset perspective[14, 15], which aims to represent the distribution of the entire dataset effectively[16–18]. For instance, Savarese[16] addressed the coreset selection challenge by formulating it as a minimax-based k-center problem[19]. The goal here is to determine k center points that cover the entire space by minimizing the maximum distance between any data point and the center point closest to that data point.

However, current coreset-based methods exhibit limitations in dealing with the data points that are unevenly distributed on the sample space, primarily because these methods do not estimate or account for distribution density. For example, coreset-based methods tend to select subsets that overly represent sparse areas in order to cover the entire sample space, consequently overlooking substantial information. As shown in Fig. 1a, points selected by the distance-based methods, represented in blue, result in a distorted representation of the original dataset. This can lead to subsets that do not accurately reflect the original dataset. Furthermore, minimax or maximin[20] distance designs are ineffective in projecting the selected design points onto subspaces. As illustrated in Fig. 1b, the representative data points from the original high-dimensional space tend to cluster and overlap when projected onto subspaces, leading to inefficient resource allocation at the subspace level. Given the principle of effect sparsity[21], which suggests that only a few dimensions in the data are statistically significant, it is crucial to project data accurately onto subspaces defined by these key dimensions. However, since the significant factors are not known in advance, ensuring an effective projection across all potential subspaces is important.

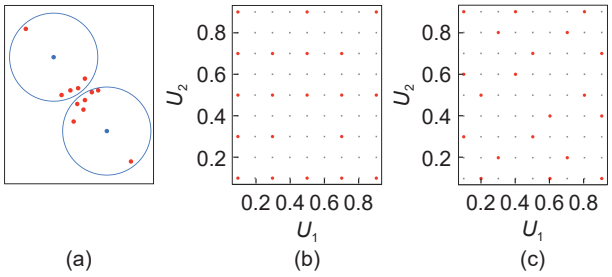


Fig. 1 A toy example illustrating the limitations of current coreset-based methods in handling data points that are unevenly distributed across the sample space. (a) Unevenly distributed sample space. Distance-based methods tend to select the points from the sparse areas (e.g., blue points) to cover the entire space, which ignores a lot of information (red points). (b) Data points with coreset design. When projected to the U_1 dimension, 20 points are collapsed into only 5 points because of overlap. (c) Data points with MaxPro design. In total 9 points are kept after the projection.

To address the aforementioned limitations, we introduce a novel diversity-based AL algorithm, named SPOT, which combines SPace-filling (SP) designs with Optimal Transport (OT) techniques. OT techniques[22–25] efficiently manage data points unevenly distributed on complex manifolds by mapping them onto a dataset uniformly distributed on a hypercube. This transformation relieves the difficulty of selecting a representative subset on the manifold to a more manageable task of choosing a subset within a hypercube. To ensure the coverage of the design points across lower-dimensional projections[21], we employ a space-filling design strategy based on Maximum Projection (MaxPro)[26]. The MaxPro design guarantees that the projection of selected design points onto any subspace maximizes space-filling properties, effectively countering the impact of effect sparsity, and thus improving the performance and robustness of the algorithm.

Furthermore, in scenarios involving the fine-tuning of pre-trained models, the unlabeled data pool may include data from classes that are not recognized by the pre-trained model. In such instances, it is critical to effectively select data from both known and unknown classes to ensure optimal performance. To tackle this challenge, we introduce a re-weighting strategy. This strategy assigns sampling probabilities that reflect not only the distribution of the unlabeled pool, but also an updated distribution incorporating insights from the labeled data. By considering both distributions, our approach enables a more informed and effective selection of data from both known and unknown classes, thereby enhancing overall model performance.

We evaluate the SPOT algorithm on three different datasets, specifically targeting the image classification task. The experimental findings demonstrate consistent improvements over baseline methods across these varied datasets and models. In summary, the key contributions of our work are as follows:

- We introduce a novel diversity-based active learning algorithm, named SPOT, which integrates SP

designs with the OT technique. OT efficiently handles data distributed on complex manifolds, while SP ensures coverage of the design points on lower-dimensional projections.

- We develop a re-weighting strategy designed to enhance the fine-tuning performance by effectively selecting data points from both the known and unknown classes of the pre-trained model.
- We conduct comprehensive experiments across various datasets and models, demonstrating that our SPOT algorithm consistently surpasses the baseline methods. These results provide new perspectives and insights into active learning methods.

2 Related Work

2.1 AL

AL algorithms are generally divided into three main categories: stream-based methods, synthesis-based methods, and pool-based methods. Stream-based AL methods[27–30] are designed to quickly decide whether to query incoming instances within a data stream. Synthesis-based algorithms[31–33] generate new instances for querying, rather than selecting from an existing dataset. Pool-based AL methods focus on selecting a specific number of unlabeled instances from an existing pool to optimize learning accuracy. Our study concentrates on pool-based AL methods, which are particularly relevant for DNNs that have access to extensive pools of unlabeled data but limited labeled data. In such scenarios, the importance of each data point for DNNs can be assessed using two main approaches: uncertainty-based and diversity-based methods.

2.2 Uncertainty-based methods

Uncertainty-based methods[34–43] aim to query data points with high uncertainty, which suggests that these points are not effectively represented by the pre-trained model. For example, Shannon[35] selected top-k instances with the highest entropy for querying. However, these methods can overlook the structural information of the unlabeled data. As a result, data points belonging to the same category often receive similar uncertainty scores from DNNs, leading to sample bias and the selection of redundant data points[44, 45].

2.3 Diversity-based methods

This paper primarily focuses on the diversity-based method, which distinguishes itself from uncertainty-based methods by emphasizing the selection of diverse samples that cover the entire sample space, considering distances between all samples. A notable example of this approach is the coreset method, which selects a representative subset by choosing data points that effectively approximate the full dataset's diversity and distribution within a reduced sample space[16–18]. Despite its strengths, there are situations where the coreset method is outperformed by uncertainty-based methods[46]. One possible explanation is that the coreset method treats each data point equally within the sample space, ignoring the inherent uneven distribution of data across a complex, high-dimensional manifold. Consequently, this method may favor points located in sparse areas, potentially overlooking more critical data points in order to achieve effective coverage. Another limitation of the coreset method arises from the projection of high-dimensional spaces, which can lead to overlapping points in low-dimensional spaces, resulting in information loss and reduced representativeness of the sampled points. These limitations are addressed in our proposed SPOT framework, which enhances the effectiveness of the coreset method.

3 Methodology

We develop a novel AL algorithm named SPOT, which integrates the space-filling design with optimal transport mapping to select a representative subsample. SPOT comprises two main steps. The first step involves linking the feature space to the unit hypercube $[0, 1]^p$, where p is the dimension of data, using the optimal transport technique, and enabling the mapping of data points from the complex feature space to a hypercube. In the second step, we employ a space-filling strategy to select the representative subsample that evenly and efficiently covers the hypercube $[0, 1]^p$. The workflow of SPOT is shown in Fig. 2.

3.1 Problem setup

Using a pre-trained model, an active learning algorithm identifies and selects the most informative data points from a large pool of unlabeled data. These selected data points are then labeled by experts. The newly labeled data are subsequently used to update and refine the model, resulting in enhanced performance.

Mathematically, consider a pre-trained base model M and an unlabeled pool $\mathcal{D}^U$ containing n unlabeled

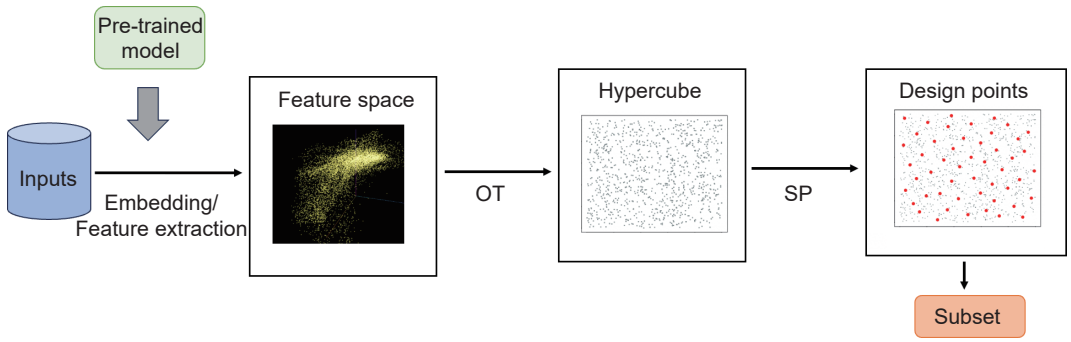


Fig. 2 Workflow of the SPOT algorithm. The inputs are first embedded into the feature space, after which they are mapped into a hypercube via the optimal transport technique. A space-filling strategy is then applied to select the representative subsample (red points) within the hypercube.

data points, denoted as $\mathcal{D}^U = \{z_i \in \mathbf{R}^p\}_{i=1}^n$, where each $z_i \in \mathbf{R}^p$ represents the p-dimensional covariates of data point i. The objective is to select a fixed-size subset $\mathcal{D}^l = \{z_j \in \mathbf{R}^p\}_{j=1}^r$ with of size r and acquire corresponding labels $\{y_j\}_{j=1}^r$. This subset is chosen to maximize the performance of model M when fine-tuned on $\mathcal{D}^l$. In classification tasks, each y_i is an integer from set $\{1, 2, \ldots, C\}$, representing the class label, where C denotes the number of classes. In regression problems, y_i is a real value.

3.2 SPOT algorithm

3.2.1 SP

To select the subset $\mathcal{D}^l$ that can best represent the whole dataset $\mathcal{D}^U$, we prefer data points that spread evenly in the dataset rather than cluster together. We use star discrepancy, a commonly employed measure, to assess the deviation of a given point set from the uniform distribution. Assuming, without loss of generality, that the unlabeled data points are distributed within the hypercube $[0, 1]^p$, our goal is to select a discrete set of data points, $\mathcal{D}^l$, which has the lowest discrepancy.

Given a p-dimensional unit hypercube $[0, 1]^p$, let $[0, a) = \prod_{j=1}^p [0, a_j)$ be a hyper-rectangle and $\mathcal{U}_r = \{\boldsymbol{u}_i\}_{i=1}^r$ be a set of r data points in $[0, 1]^p$, the star discrepancy is defined as

$$D^*(\mathcal{U}_r) = \sup_{a \in [0,1]^p} \left| \frac{1}{r} \sum_{i=1}^r 1\{\boldsymbol{u}_i \in [0, a)\} - \prod_{j=1}^p a_j \right| \quad (1)$$

The subset $\mathcal{U}_r$ that minimizes D^* is optimal for representing the hypercube space effectively. Several uniform design methods[47] have been proposed to generate such $\mathcal{U}_r$. However, these methods are computationally intensive and challenging to apply to datasets with large sample sizes. To reduce the computational load, we employ space-filling design strategies[21, 48], which create $\mathcal{U}_r$ with low star discrepancy.

We utilize the MaxPro, a space-filling strategy, to select a representative subset in $[0, 1]^p$. MaxPro helps avoid the suboptimal projections encountered in minimax or maximin distance designs, as illustrated in Fig. 1a. In this approach, when data are projected onto a subspace defined by several original dimensions, the distance between points $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ is calculated using the weighted Euclidean distance, defined as

$$d(\boldsymbol{u}_i, \boldsymbol{u}_j; \boldsymbol{\delta}) = \left\{ \sum_{l=1}^p \delta_l (u_{il} - u_{jl})^2 \right\}^{1/2} \quad (2)$$

where $\boldsymbol{u}_i = (u_{i1}, u_{i2}, \ldots, u_{ip})^{\mathrm{T}}$, $\boldsymbol{\delta} = \{\delta_1, \delta_2, \ldots, \delta_p\}^{\mathrm{T}}$, $i \in \{1, 2, \ldots, r\}$. $\delta_l = 1$ if dimension l participating in forming the subspace, otherwise $\delta_l = 0$ for $l \in \{1, 2, \ldots, p\}$. We aim to select a subset $\mathcal{U}_r$ that minimizes the projection error across all subspaces, defined as

$$E\{\phi_k(\mathcal{U}_r; \boldsymbol{\delta})\} = \int_{S_{p-1}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{d^k(\boldsymbol{u}_i, \boldsymbol{u}_j; \boldsymbol{\delta})} p(\boldsymbol{\delta}) \, \mathrm{d}\boldsymbol{\delta} \quad (3)$$

where $S_{p-1} = \left\{\theta: \delta_1, \delta_2, \ldots, \delta_{p-1} \geqslant 0, \sum_{i=1}^{p-1} \delta_i \leqslant 1\right\}$, and $k > 0$ is a constant. This ensures optimal representation in each considered subspace. We refer to Ref. [26] for more details.

We propose Algorithm 1 to select the representative subset $\mathcal{U}_r$ within the unit hypercube $[0, 1]^p$. This approach integrates the space-filling design with a 1-nearest neighbor method similar to that of Zhang et al.[49] First, we scale the original sample $\mathcal{D}^U$ to $\mathcal{X}^U$, ensuring it is distributed within $[0, 1]^p$. We then generate MaxPro design points within this space. For

Algorithm 1 SP algorithm

Input: Observed sample $\mathcal{D}^U = \{z_i \in \mathbf{R}^p\}_{i=1}^n$ and budget r

Step 1: Scale $\mathcal{D}^U = \{z_i \in \mathbf{R}^p\}_{i=1}^n$ to $\mathcal{X}^U = \{\boldsymbol{x}_i \in [0, 1]^p\}_{i=1}^n$;

Step 2: Generate a set of MaxPro space-filling design points $\{\boldsymbol{u}_j\}_{j=1}^r \in [0, 1]^p$;

Step 3: For $j = 1$ to r, select the nearest neighbor $\boldsymbol{x}_j$ for $\boldsymbol{u}_j$ from $\mathcal{X}^U$ using the Euclidean distance;

Output: Selected subset $\{\boldsymbol{x}_j\}_{j=1}^r$

each design point, denoted as $\boldsymbol{u}_j \in [0, 1]^p$, we identify its nearest neighbor $\boldsymbol{x}_j \in \mathcal{X}^U$. This neighboring point $\boldsymbol{x}_j$ is the data point we select to fine-tune the model.

3.2.2 OT

For any $\mathcal{D}^U = \{z_j \in \mathbf{R}^p\}_{j=1}^r$, Algorithm 1 can be applied following a simple scaling step. Nonetheless, challenges arise when the data points are non-uniformly distributed across the sample space. Employing the MaxPro space-filling design method under these conditions often leads to suboptimal outcomes. Firstly, as illustrated in Fig. 3a, Algorithm 1 tends to select the subset that overly represents data points from sparse areas. Secondly, for data points that are non-uniformly distributed in the sample space, utilizing a uniformly distributed space-filling design set to locate the nearest neighbor may not be reasonable. This is because even its nearest neighbor can still be significantly distant, making this approach ineffective.

We apply the OT technique[22, 50] to transfer the dataset $\mathcal{D}^U$, which is unevenly distributed on a complex manifold, into a uniformly distributed dataset within a unit hypercube $[0, 1]^p$. The transformation

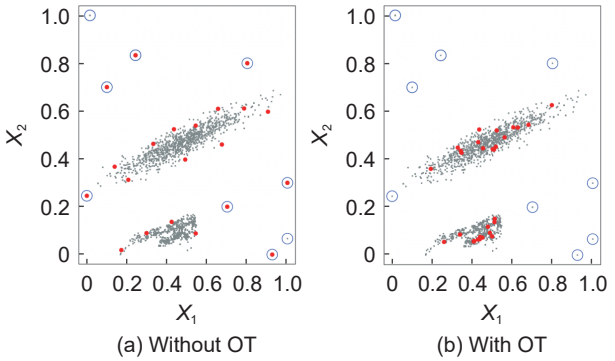


Fig. 3 Power of OT on the unevenly distributed sample (grey points). The points from the sparse areas are considered as outliers (circled in blue). (a) Subset (red points) selected by applying Algorithm 1 directly, and (b) subset (red points) selected by applying Algorithm 1 after the optimal transport method.

simplifies the challenging task of selecting a representative subset from the manifold to selecting one from a dataset uniformly distributed in a hypercube. Consequently, the effectiveness of Algorithm 1 is fully demonstrated, as shown in Fig. 3b. We observe that the selected data points are more concentrated to the true distribution, and it is robust to this non-uniformly distribution with outliers.

Assume μ is the probability measure on space $\mathcal{X} \in \mathbf{R}^p$, the domain of the random variable, and ν is the uniform probability measure on $\mathcal{Y} = [0, 1]^p$. Let $T: \mathcal{X} \to \mathcal{Y}$ be a transport map that transports $\mu \in \mathcal{P}(X)$ to $\nu \in \mathcal{P}(\mathcal{Y})$, where $\mathcal{P}(\cdot)$ is the set of probabilities. T is defined as

$$\nu(\mathcal{B}) = \mu\left(T^{-1}(\mathcal{B})\right) \tag{4}$$

for all ν-measurable sets $\mathcal{B}$. As shorthand we write $\nu = T_{\#}\mu$ if Eq. (4) is satisfied. The focus is primarily on the cost of transporting μ to ν. Specifically, let $c: \mathcal{X} \times \mathcal{Y} \to [0, +\infty]$ be a cost function, where $c(\boldsymbol{x}, \boldsymbol{y})$ measures the cost of transporting one unit of mass from $\boldsymbol{x} \in \mathcal{X}$ to $\boldsymbol{y} \in \mathcal{Y}$. The objective is to search the optimal transport map T^* that minimizes

$$h(T) = \int_X c(\boldsymbol{x}, T(\boldsymbol{x}))\, \mathrm{d}\mu(\boldsymbol{x}) \tag{5}$$

over μ-measurable maps $T: \mathcal{X} \to \mathcal{Y}$ subject to $\nu = T_{\#}\mu$.

To obtain the desired optimal transport map that maps the observed sample to be uniformly distributed on $[0, 1]^p$, a synthetic sample, $\mathcal{U}_n = \{\boldsymbol{u}_i\}_{i=1}^n$, uniformly distributed on $[0, 1]^p$ is first generated. Subsequently, T^*, mapping from the observed sample to $\mathcal{U}_n$ is calculated. This mapping can be approximated using projection-based methods[25, 51], which simplify the estimation of a p-dimensional optimal transport map by addressing it through a sequence of one-dimensional subproblems. These subproblems, involving the calculation of one-dimensional optimal transport maps between projected samples, are readily solved using sorting algorithms. The set $\mathcal{U}_r$ is then selected according to Eq. (1) based on space-filling designs. The observed samples transported to $\mathcal{U}_r$ by T^* form the targeted subsample. This procedure is outlined in Algorithm 2. The selected subset is subsequently annotated with expert knowledge and utilized to refine the current model M.

We further illustrate Algorithm 2 using a toy example as shown in Fig. 4. We generate two distinct classes of random samples, each consisting of 3000 points. The first class is sampled from a

Algorithm 2 Naive-SPOT algorithm

Input: $\mathcal{D}^U$, $\mathcal{D}^L$, and budget r

Step 1: Generate a synthetic sample $\mathcal{U}_n = \{\boldsymbol{u}_i\}_{i=1}^n$ uniformly distributed on the unit hypercube $[0, 1]^p$;

Step 2: Calculate the optimal transport map T^* that maps $\mathcal{D}^U = \{z_i \in \mathbf{R}^p\}_{i=1}^n$ to $\mathcal{U}_n$, denote the transformed sample as $\{T^*(z_i)\}_{i=1}^n$;

Step 3: Generate the MaxPro space-filling design points $\{\boldsymbol{u}_j\}_{j=1}^r$ = SP $(\mathcal{U}_n, r)$;

Step 4: Achieve the subset $\mathcal{D}^l = \{z_j\}_{j=1}^r$ mapped to $\{\boldsymbol{u}_j\}_{k=1}^r$ by T^*;

Output: Selected subset $\mathcal{D}^l$

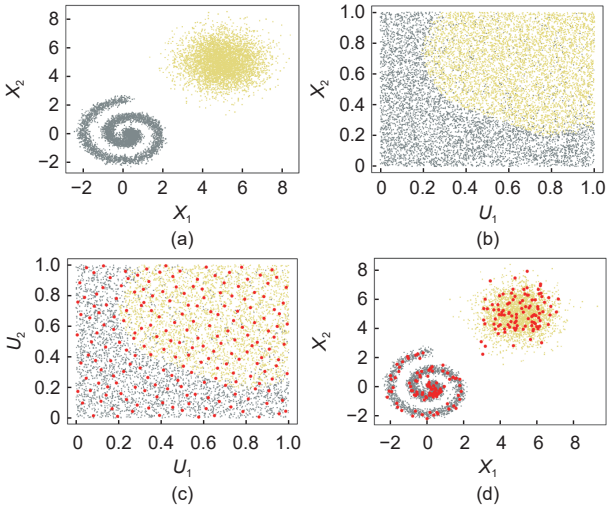


Fig. 4 Toy example of the proposed SPOT algorithm. (a) Original data consisting of two classes, distinguished by different colors, (b) optimal transport maps the original data to the synthetic data uniformly distributed on the 2D unit hypercube $[0, 1]^2$, (c) generated space-filling design points (red points) covering the unit hypercube $[0, 1]^2$, and (d) subset of the original data (red points) mapped to the selected synthetic data.

normal distribution $\mathcal{N}\left(\begin{pmatrix} a\cdot\sin(2a) \\ a\cdot\cos(2a)\end{pmatrix}, \begin{pmatrix} 0.4^2 & 0 \\ 0 & 0.4^2\end{pmatrix}\right)$, where $a \sim \text{Unif}(0, 2\pi)$, and the second from $\mathcal{N}\left(\begin{pmatrix} 5 \\ 5\end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1\end{pmatrix}\right)$, as displayed in Fig. 4a. Following this, we map the generated data to a synthetic dataset uniformly distributed on $[0, 1]^2$ as per Step 2 in Algorithm 2. Figure 4b confirms that data from the same class remain spatially close even after the OT step, validating the logic of the subsequent space-filling selection procedure. The SP design points are marked in red in Fig. 4c. The subsample corresponding to these design points, marked in red in Fig. 4d, form the desired subsample.

3.2.3 Down-weight

Algorithm 2 is designed to select a small subset of data that effectively represents the entire unlabeled dataset. This is particularly useful when the unlabeled data comes from classes that differ from those in the base dataset $\mathcal{D}_0^L$, which is used to train the base model. However, in practice, $\mathcal{D}^U$ may also contain data points belonging to the same classes as $\mathcal{D}_0^L$, which the base model already distinguishes well. In these cases, we prefer to reduce the probability of selecting such data points. To address this issue, we introduce a down-weighting method that adjusts the input to our selection procedure.

To illustrate our method, consider a scenario involving two classes. Assume the unlabeled dataset $\mathcal{D}^U = \{z_i \in \mathbf{R}^p\}_{i=1}^n$ contains two classes: $C_1^U = \{z_j : j \in I_1\}$ and $C_2^U = \{z_k : k \in I_2\}$, where $I_1 \cup I_2 = \{1, 2, \ldots, n\}$, $I_1 \cap I_2 = \varnothing$, $|I_1| = m_1$, and $|I_2| = m_2$. Furthermore, suppose that C_1^U contains the same classes as the base labeled dataset $\mathcal{D}_0^L = \{\boldsymbol{x}_i\}_{i=1}^N$, where N is the sample size of $\mathcal{D}_0^L$. We denote the sampling probabilities for data points in C_1^U, C_2^U, and $\mathcal{D}_0^L$ by π_{1j} (for $j \in I_1$), π_{2k} (for $k \in I_2$), and π_{0i} (for $i = 1, 2, \ldots, N$), respectively.

According to the principle of OT and SP, the proportion of the selected subset from each class should be proportional to the sample size of each class. Therefore, when incorporating the base dataset $\mathcal{D}_0^L$, the probability of selecting each data point into the subset is given by

$$\hat{\pi}_{0i} = \frac{n}{n+m_1+m_2} \times \frac{\pi_{0i}}{\sum_{i=1}^{n}\pi_{0i} + \sum_{i=1}^{m_1}\pi_{1i}} \tag{6}$$

$$\hat{\pi}_{1j} = \frac{m_1}{n+m_1+m_2} \times \frac{\pi_{1j}}{\sum_{i=1}^{n}\pi_{0i} + \sum_{i=1}^{m_1}\pi_{1i}} \tag{7}$$

$$\hat{\pi}_{2k} = \frac{m_2}{n+m_1+m_2} \times \frac{\pi_{2k}}{\sum_{i=1}^{m_2}\pi_{2i}} \tag{8}$$

where $\hat{\pi}_{0i}$, $\hat{\pi}_{1j}$, and $\hat{\pi}_{2k}$ represent the adjusted sampling probability for each data point in $\mathcal{D}_0^L$, C_1^U, and C_2^U, respectively. Without including the base dataset, the adjusted probabilities are as follows:

$$\tilde{\pi}_{1j} = \frac{m_1}{m_1+m_2} \times \frac{\pi_{1j}}{\sum_{i=1}^{m_1}\pi_{1i}} \tag{9}$$

$$\tilde{\pi}_{2k} = \frac{m_2}{m_1 + m_2} \times \frac{\pi_{2k}}{\sum_{i=1}^{m_2} \pi_{2i}} \tag{10}$$

For any $z_j \in C_1^U$ and $z_k \in C_2^U$, without the base dataset, the ratio of the selection probability is given by

$$k_0 = \frac{\hat{\pi}_{1j}}{\hat{\pi}_{2k}} = \frac{\pi_{1j} \sum_{i=1}^{m_2} \pi_{2i}}{\pi_{2k} \sum_{i=1}^{m_1} \pi_{1i}} \tag{11}$$

However, when including the base dataset, this ratio becomes

$$k_1 = \frac{\tilde{\pi}_{1j}}{\tilde{\pi}_{2k}} = \frac{\pi_{1j} \sum_{i=1}^{m_2} \pi_{2i}}{\pi_{2k} \left(\sum_{i=1}^{n} \pi_{0i} + \sum_{i=1}^{m_1} \pi_{1i} \right)} \tag{12}$$

which is lower than k_0. Thus, we effectively decrease the probability of selecting data points from classes that are already well represented. Further details of this method under general conditions are outlined in the Algorithm 3.

In the first step of the SPOT Algorithm 3, simple random sampling is employed to select a subset from the labeled base dataset, $\mathcal{D}_0^L$. To enhance the performance of the SPOT algorithm, the potential for incorporating more advanced sampling techniques[52–54] can be further investigated.

4 Experiment

This section provides an overview of the datasets and algorithms to be employed in our experiments, followed by an experimental analysis. We perform a thorough evaluation of SPOT across multiple classification tasks utilizing various models. Furthermore, we conduct a sensitivity analysis to assess the effects of several critical parameters on the performance of the SPOT algorithm.

Algorithm 3 SPOT algorithm

Input: $\mathcal{D}^U$, $\mathcal{D}_0^L$, and budget r

Step 1: Randomly select a subset $\mathcal{D}_{\text{sub}}^L$ from $\mathcal{D}_0^L$ with the size of the sample order with $\mathcal{D}^U$;

Step 2: Form $\mathcal{D}_{\text{new}} = \mathcal{D}^U \cup \mathcal{D}_{\text{sub}}^L$;

Step 3: Pass $\mathcal{D}_{\text{new}}$ into the Naive-SPOT algorithm to select a subset for labeling:

$$\mathcal{D}^l = \text{Naive-SPOT}\,(\mathcal{D}_{\text{new}}, r),$$

- If the selected subset contains samples from $\mathcal{D}_0^L$, no budget is used for these alreadylabeled data points;
- The saved budget can either be preserved or used to annotate additional samples;

Output: Selected subset $\mathcal{D}^l$

4.1 Baselines

To validate the performance of our approach, we compare it against a number of baselines:

- **Coreset:** Selecting the subset $\mathcal{D}^l$ using the K-center algorithm (K is equal to the budget) developed in Sener and Savarese[16].
- **Batch Active learning by Diverse Gradient Embeddings (BADGE):** A sampling strategy that incorporates both predictive uncertainty and sample diversity, as proposed by Ash et al.[55]
- **K-means:** Partitioning $\mathcal{D}^U$ into K clusters (K is equal to the budget) according to Sculley[56] and take the cluster centroids as $\mathcal{D}^l$.
- **Random:** Selecting the subset $\mathcal{D}^l$ uniformly at random from $\mathcal{D}^U$.
- **Least Confidence (LC) :** Selecting $\mathcal{D}^l$ for which the pre-trained model M is least confident in class assignment.
- **Active Learning By Learning (ALBL):** A bandit-style meta-active learning algorithm that selects between Coreset and LC at every round[57].
- **Generalized Empirical F-Discrepancy (GEFD):** It is a low GEFD data-driven subsampling method according to Zhang et al.[58]

4.2 Size of the budget

Different from many previous AL studies that allocate a large budget for their experiments, we focus on the scenarios where budget r is very limited. This focus mirrors situations where labeling is extremely expensive, as is often the case in fields such as medical imaging. Specifically, for the situation that $\mathcal{D}^U$ contains tens of thousands of data points, we limit the size of the budget to the order of tens, i.e., the few-shot scenario[59, 60].

4.3 Dataset

Agri-ImageNet: The Agri-ImageNet dataset[61] contains two parent classes including fruits (with 11 sub-classes) and vegetables (with 4 sub-classes).

MNIST: MNIST[62] is a dataset of handwritten digit images with a training set of 60 000 examples and a test set of 10 000 examples. Each example is a 28 pixel × 28 pixel grayscale image, associated with a label of 10 classes.

CIFAR-10: The CIFAR-10 dataset[63] consists of a training set of 50 000 examples and a test set of 10 000 examples. Each example in the dataset is a 32 pixel × 32 pixel color image, spanning 10 different classes of objects such as animals and vehicles. These classes include airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks, each equally represented in the dataset.

4.4 Implementation details

We briefly introduce some important implementation details of our experiments. Detailed information of our experiments are presented in the Appendix.

Dataset settings: For all datasets, we randomly separate them into the base dataset and the novel dataset. Model M is pre-trained on the base set, and the active learning algorithms are applied to the novel set. The base dataset is randomly divided into training (80%) and test (20%) splits. For the novel dataset, all samples except those actively selected for fine-tuning are used as the test split. Image pre-processing steps are also applied. Specifically, for the training dataset, Rand-Augment[64], Random Erasing[65], and RandomResizeCrop are applied for data augmentation. For the test dataset, images are only resized and center-cropped. In Table 1, we list some basic information about the three datasets, where m_1 and m_2 denote the numbers of classes in $\mathcal{D}_o^L$ and $\mathcal{D}^U$, respectively, n_1 and n_2 denote the numbers of images in $\mathcal{D}_o^L$ and $\mathcal{D}^U$, respectively.

Model settings: We consider two different model structures. For Agri-ImageNet and CIFAR-10 dataset, we apply the Vision Transformer (ViT)[2] model in the experiments. The ImageNet-1k pre-trained model is firstly trained on the base dataset with the vanilla ViT. We adopt an AdamW optimizer with 300 epochs using a cosine decay learning rate scheduler and 5 epochs of linear warm-up. For the MNIST dataset, a Convolutional Neural Network (CNN) with two sequential layers and one fully connected layer is applied.

Feature extraction: For the distance-based methods (Coreset and K-means), we follow the instructions in Ref. [16] to define the distance metric. Specifically, we use the l_2 distance between the final fully connected layers as the distance. For SPOT, since the properties of space-filling designs are restricted to a relatively low dimension, we further apply a simple Autoencoder and Principal Component Analysis (PCA) step to reduce the dimension. This feature extraction procedure is solely used for selecting $\mathcal{D}^I$ and will not be applied in the subsequent model fine-tuning step.

For all the active learning algorithms with randomness, we run them with three random seeds and use the median accuracy as a metric.

Table 1 Datasets used in the experiments.

Dataset	m_1	m_2	n_1	n_2	Model
CIFAR-10	7	6	30 500	19 500	ViT
Agri-ImageNet	3	8	3149	1491	ViT
MNIST	7	6	36 781	23 219	CNN

4.5 Results

Figures 5–7 show the results of classification accuracies versus different budget r. Three significant observations can be made from these results. First, it is observed that as the budget increases, the accuracy of all methods generally exhibits an upward trend. Although there may be slight drops in accuracy at certain points, such as when $r = 50$ for the SPOT

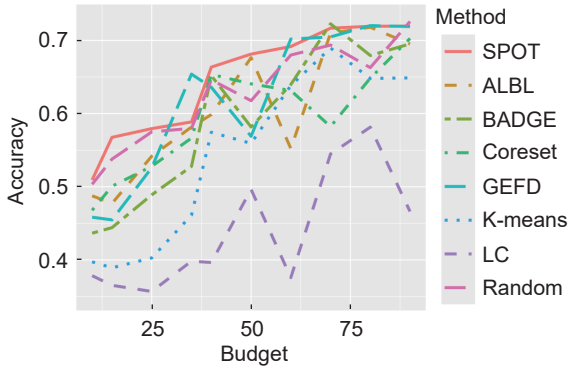


Fig. 5 Image classification accuracy given the budgets (number of training samples) on the CIFAR-10 dataset with the ViT model.

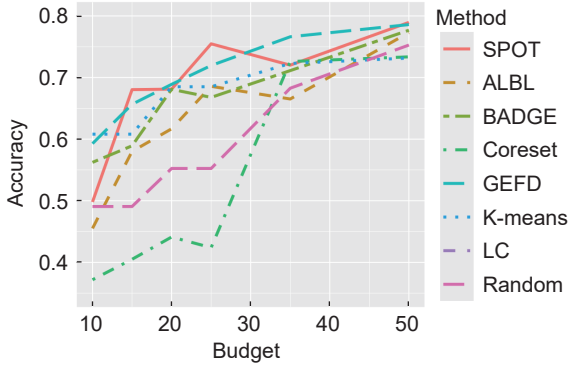


Fig. 6 Image classification accuracy given the budgets (number of training samples) on the Agri-ImageNet dataset with the ViT model.

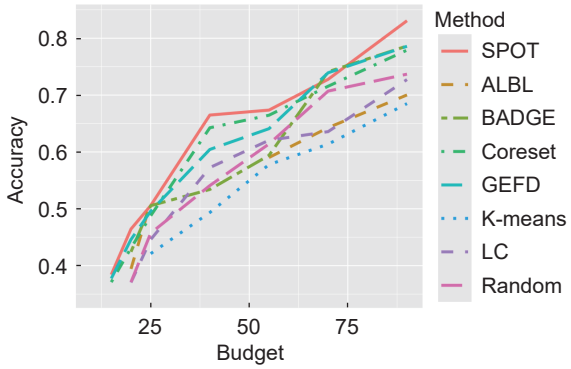


Fig. 7 Image classification accuracy given the budgets (number of training samples) on the MNIST dataset with the CNN model.

algorithm, the overall trend remains positive. Notably, the proposed SPOT algorithm consistently outperforms the other methods for both datasets in most cases with a few exceptions that GEFD achieves marginally better accuracies. These findings align with the statements and demonstrations provided in the methodology section and the accompanying toy examples. They reinforce the notion that the subset selected by the SPOT algorithm better represents the observed sample space compared to the subsets selected by the other four methods.

Second, we note that even with a significantly limited budget (specifically, a budget controlled to be under 100), DNNs can still achieve good performance by taking advantage of active learning algorithms. Utilizing the SPOT algorithm, the classification accuracy on both datasets reaches 0.7. This highlights the efficacy of SPOT in maximizing performance even under resource constraints.

Third, we observe that the accuracy of all methods gradually approaches a fixed value, differing only in their convergence rates. This behavior is expected since, as the training sample size increases, the distinctions between various active learning methods diminish until they become negligible. For instance, the convergence rate of the star discrepancy for space-filling design points is of the order $O(\log(r)^p/r)$, while the convergence rate for uniformly random sampling is of the order $O(\log(\log(r))/\sqrt{r})$[66], which is significantly slower than $O(\log(r)^p/r)$. However, as the budget r goes to infinity, even uniformly random sampling will perform well. One exception is the performance of the entropy-based method LC in the Agri-ImageNet dataset. As r increases, its accuracy barely changes. This may be due to DNNs giving similar uncertainty estimates to the data points belonging to the same class.

4.6 Computational time

Although the expensive labeling procedure constitutes a significant cost in active learning algorithms, it is also essential to consider the computational time required by the proposed SPOT algorithm. Typically, the pre-trained model used in active learning is not counted as part of the computational cost, since it is trained on large benchmark datasets like ImageNet. Thus, the computational time for the model-building procedure consists of two main steps: (1) selecting the subset $\mathcal{D}^l$ for annotation, and (2) the fine-tuning process to adapt the pre-trained model to the novel dataset. The subset selection step is performed on a Mac with a 10-Core M1 Max processor and 32 GB memory, utilizing the CPU. On the other hand, the fine-tuning process is executed on an NVIDIA Tesla V100 Tensor Core. We list the computational time of Step 1 in Table 2 and the computational time for Step 2 in Table 3.

Table 2 shows that the computational time required by different subset selection methods varies greatly, but overall time for this step can usually be completed within several minutes. A running time of zero here indicates that the execution is completed in less than one second. While the fine-tuning step is more time-consuming, requiring several hours to fine-tune the ViT model.

4.7 Parameter sensitivity

To assess the impact of parameterization changes in the

Table 2 Median computational time for subset selection.

(s)

Dataset	Method							
	SPOT	Coreset	K-means	Random	LC	ALBL	BADGE	GEFD
Agri-ImageNet	23	50	416	0	119	201	223	1
MNIST	174	6	1300	0	39	205	556	0
CIFAR-10	132	83	1081	0	54	371	368	1

Table 3 Median computational time for fine-tuning.

(h)

Dataset	Method							
	SPOT	Coreset	K-means	Random	LC	ALBL	BADGE	GEFD
Agri-ImageNet	11.57	12.25	14.55	11.38	14.37	19.07	19.06	16.52
MNIST	0.17	0.18	0.15	0.18	0.08	0.07	0.13	0.12
CIFAR-10	3.08	3.89	3.43	3.05	2.55	1.71	3.51	3.20

dimension reduction phase on classification performance, we conduct experiments using the CIFAR-10 dataset. Specifically, we analyze the effects of modifying the number of the first principal components, i.e., the dimension p. The results, as depicted in Fig. 8, consistently demonstrate stable classification performance across various values of p. This finding suggests that the performance remains robust and unaffected by changes in the specific values of p. More details of this part can be found in the Appendix.

4.8 Ablation study

We conduct experiments to assess the influence of space-filling designs and OT individually. Using the MNIST dataset as an example, we compare SPOT with the following methods: (1) OT, which applies optimal transport with a simple random Latin hypercube design[67] instead of the proposed MaxPro space-filling design; and (2) SP, which uses the MaxPro space-filling design without OT. Figure 9 shows the classification accuracies of these three methods at varying budget levels r. The results demonstrate that SPOT consistently achieves higher classification accuracy compared to both OT and SP, with the performance gap increasing at higher budget levels.

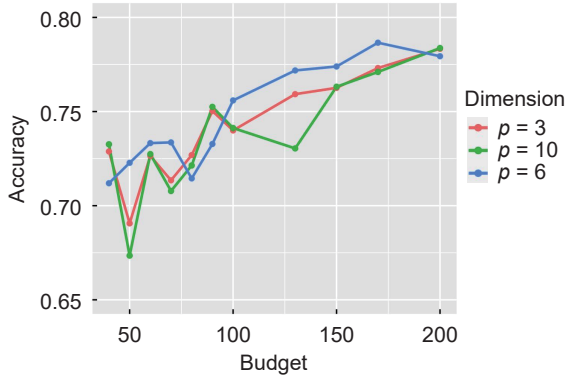


Fig. 8 Results of the parameter sensitivity test for the dimension reduction part.

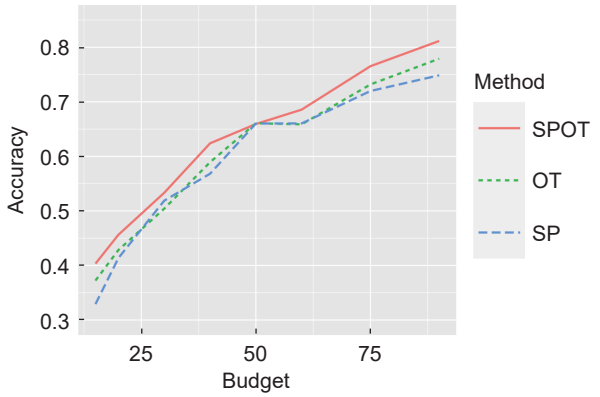


Fig. 9 Image classification accuracy given the budgets (number of training samples) on the MNIST dataset.

5 Conclusion and Discussion

In this paper, we introduce a novel active learning framework that combines SP designs and OT to effectively select representative subsets that capture the underlying distribution of the entire dataset. In particular, our design remedies the limitations in coreset-based methods from the uneven distribution density of data points and ineffective projection onto sub-spaces. Through extensive experiments on three diverse datasets using various models, we demonstrate the superiority of our proposed methods compared to the baseline approaches. The results highlight the effectiveness and robustness of our framework. As part of our future work, we aim to apply the SPOT framework to other scenarios, including medical imaging applications and other data modalities such as text, time series, and videos. This will allow us to explore the potential benefits and practicality of our approach in broader domains.

The computational cost of SPOT depends on both the OT step and the SP step. Traditional linear programming algorithms for solving OT problems have a computational complexity of $O\left(n^3 \log(n)\right)$. Additionally, the MaxPro design step has a complexity of $O\left(n^2 \cdot p\right)$, where n is the sample size and p is the

data dimension. For large-scale datasets, such as medical imaging data, the high computational cost of OT poses a significant challenge to implementing SPOT. Fortunately, efficient OT algorithms, such as the Sinkhorn algorithm, have been developed to significantly reduce computational time. Empirical studies demonstrate that the Sinkhorn algorithm, with a complexity of $O\left(n^2 \log(n)\right)$, can solve OT problems reliably and efficiently for datasets with $n \approx 10^4$[24]. Furthermore, under sparsity assumptions, the computational cost can be further reduced, with efficiency demonstrated on datasets as large as $n \approx 10^6$[68]. Thus, the SPOT algorithm remains feasible and practical for most applications, even with large-scale datasets.

Appendix

In this section, we provide detailed information of our experiments.

A1 Dataset Setting

A1.1 Splitting the dataset

We partition each dataset into two subsets: the base dataset and the novel dataset. The pre-trained model is trained using the base dataset, while the active learning algorithm is applied to the novel dataset. In the case of the CIFAR-10 and MNIST datasets, the novel dataset comprises both classes that are already present in the base dataset and additional classes that are not included in the base dataset. For the Agri-ImageNet dataset, all the classes in the novel dataset are entirely new.

CIFAR-10: We design all data samples belonging to four classes (airplane, automobile, bird, and cat) and randomly allocate 70% of the data from three classes (deer, dog, and frog) to form the base dataset. Subsequently, we assign the remaining 30% of the three classes (deer, dog, and frog) along with all data samples from three classes (horse, ship, and truck) as the novel dataset.

Agri-ImageNet: The base dataset contains three classes (Chinee apple, maize, and tomato), while the novel dataset contains 12 classes (apple, fuji apple, golden delicious apple, melrose apple, apple tree, avocado, capsicum, lettuce, mango, orange, rockmelon, and strawberry).

MNIST: Similar to the CIFAR-10 dataset, we set all data from four classes (digit 0–3) and randomly select 70% of the data from three classes (digit 4–6) as the base dataset. We then set the rest of the data, i.e. 30% of three classes (digit 4–6) and all data from three classes (digit 7–9), as the novel dataset.

A1.2 Dataset settings

For all datasets, the base dataset is randomly split into training and testing subsets with an 80%/20% ratio. The novel dataset's test split consists of the remaining data after excluding the actively selected few-shot samples. Image preprocessing steps are applied as follows: for the training dataset, data augmentation techniques, such as Rand-Augment[64], Random Erasing[65], and RandomResizeCrop, are applied. Specifically, images are resized and cropped to 32 pixel × 32 pixel for CIFAR-10, 224 pixel × 224 pixel for Agri-ImageNet, and 28 pixel × 28 pixel for MNIST. For the test dataset, images undergo only resizing and center-cropping to 32 pixel × 32 pixel (CIFAR-10), 224 pixel × 224 pixel (Agri-ImageNet), and 28 pixel × 28 pixel (MNIST).

A2 Model Setting

ViT model for CIFAR10: We use the ViT[2] model in the experiments. The ImageNet-1k pre-trained model is firstly trained on the base dataset with the vanilla ViT. We adopt an AdamW optimizer with 300 epochs using a cosine decay learning rate scheduler and 5 epochs of linear warm-up. Then, we fine-tune the model on the few-shot samples in the novel dataset. We keep the same settings of regular training except for the epochs to 200.

ViT model for Agri-ImageNet: We use the ViT[2] model in the experiments. The ImageNet-1k pre-trained model is firstly trained on the base dataset with the vanilla ViT. We adopt an AdamW optimizer with 100 epochs using a cosine decay learning rate scheduler and 5 epochs of linear warm-up. Then, we fine-tune the model on the few-shot samples in the novel dataset. We keep the same settings as regular training.

CNN model for MNIST: We use a CNN with two sequential layers and three fully connected layers. The CNN model is first trained on the base dataset. We adopt an Adam optimizer with 100 epochs and 5 epochs of linear warm-up. Then, we fine-tune the model on the few-shot samples in the novel dataset. We keep the same settings of regular training except for the epochs to 300.

A3 Feature Extraction

For particularly high-dimensional data such as images,

they are not reliable or even feasible for us to use the original high-dimensional data for analysis. Thus, a feature extraction step, which has the ability to extract low-dimensional features that can preserve the most relevant information from the original dataset and discard the redundant information, is desired before applying the active learning algorithms. For the classification problems, since the pre-trained model itself has the ability to extract important features required to distinguish classes, we take advantage of it to finish the feature extraction step.

ViT model: For the distance-based methods (Coreset and KNN), we follow the instruction in Ref. [16] to extract the low-dimensional feature and define the distance metric. Specifically, take the output of the last block of the ViT model as the image features, and use the l_2 distance as the distance metric. For SPOT, since the properties of space-filling designs are restricted to a relatively low dimension, we further apply a simple Autoencoder with a three-layer encoder and a three-layer decoder to reduce the dimension. The principal component analysis is applied when needed.

CNN model: For the distance-based methods (Coreset and KNN), we take the output of the second fully connected layer of the CNN model as the image features, and use the l_2 distance as the distance metric. For SPOT, we use the principal component analysis to reduce the dimension further when needed.

A4 Parameter Sensitivity

In order to evaluate the robustness of the proposed SPOT algorithm over the parameters in the dimension reduction step, we take the benchmark dataset CIFAR-10 as an example to conduct experiments. Specifically, we test the influence of (1) the number of nodes d for the latent layer in autoencoder, and (2) the number of principal components p used in PCA.

Specifically, we first fix p to be 6 and vary d among 50, 100, and 150 to explore the influence of d. Results are shown in Fig. A1. We observe that the overall trend of accuracy is upward as the shot size increases for all scenarios. For different d, the increase in accuracy of the proposed SPOT algorithm is stable, while the increase of the random sampling method fluctuates greatly. Moreover, the performance of the proposed SPOT algorithm is stable across different values of d, and outperforms the random sampling algorithm for almost all scenarios.

Then we fix d to be 100 and vary p among 3, 6, and 10 to explore the influence of p. Results are shown in Fig. A2. Similar to the phenomenon in Fig. A1, the overall trend of accuracy is upward as the shot size increases for all scenarios. For different p, the increase in accuracy of the proposed SPOT algorithm is more stable than the random sampling method. Moreover, the performance of the proposed SPOT algorithm has better performance than random sampling in all scenarios and is stable across different values of p.

Acknowledgment

This work was supported by the U.S. National Science Foundation (Nos. DMS-1925066, DMS-1903226, DMS-2124493, DMS-2311297, DMS-2319279, and DMS-2318809) and the National Institutes of Health (No. NIH R01GM152814).

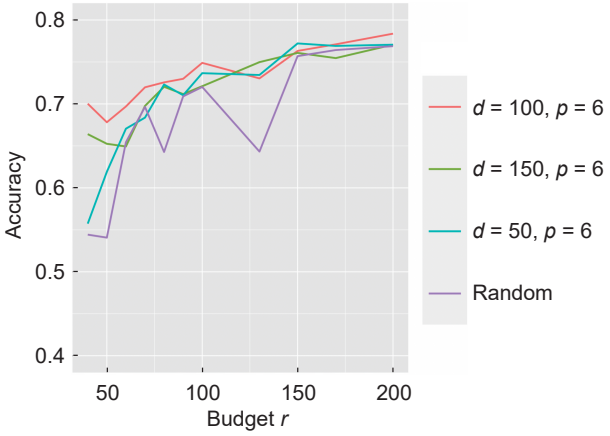


Fig. A1 Image classification accuracy given the budgets (number of training samples) on the CIFAR-10 dataset with various *d*.

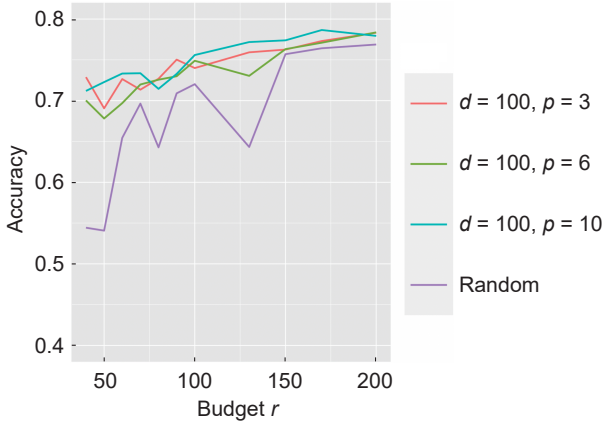


Fig. A2 Image classification accuracy given the budgets (number of training samples) on the CIFAR-10 dataset with various *p*.

References

[1] H. Jiang, Z. Diao, T. Shi, Y. Zhou, F. Wang, W. Hu, X. Zhu, S. Luo, G. Tong, and Y. D. Yao, A review of deep learning-based multiple-lesion recognition from medical images: Classification, detection and segmentation, *Comput. Biol. Med.*, vol. 157, p. 106726, 2023.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, in *Proc. 9th Int. Conf. Learning Representations*, Virtual Event, Austria,https://dblp.uni-trier.de/db/conf/iclr/iclr2021.html#DosovitskiyB0WZ21, 2007.

[3] L. Fang, Y. Chen, W. Zhong, and P. Ma, Bayesian knowledge distillation: A bayesian perspective of distillation with uncertainty quantification, in *Proc. 41st Int. Conf. Machine Learning*, Vienna, Austria, 2024. https://dblp.uni-trier.de/db/conf/icml/icml2024.html#FangCZM24.

[4] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in *Proc. 2019 Conf. North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*, Minneapolis, MN, USA, 2019, pp. 4171–4186.

[5] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, in *Proc. 34th Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2020, p. 159.

[6] OpenAI, Gpt-4 technical report, arXiv preprint arXiv: 2303.08774, 2023.

[7] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, The computational limits of deep learning, arXiv preprint arXiv: 2007.05558, 2020.

[8] B. R. Bartoldson, B. Kailkhura, and D. Blalock, Compute-efficient deep learning: Algorithmic trends and opportunities, *J. Mach. Learn. Res.*, vol. 24, no. 122, pp. 1–77, 2023.

[9] X. Liu, L. Xie, Y. Wang, J. Zou, J. Xiong, Z. Ying, and A. V. Vasilakos, Privacy and security issues in deep learning: A survey, *IEEE Access*, vol. 9, pp. 4566–4593, 2021.

[10] B. Settles, *Active Learning Literature Survey*. Madison, WI, USA: University of Wisconsin Madison, 2009.

[11] P. Ren, Y. Xiao, X. Chang, P. Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, A survey of deep active learning, *ACM Computing Surveys*, vol. 54, no. 9, pp. 1–40, 2022.

[12] C. Zhao, B. Qin, S. Feng, W. Zhu, W. Sun, W. Li, and X. Jia, Hyperspectral image classification with multi-attention transformer and adaptive superpixel segmentation-based active learning, *IEEE Trans. Image Process.*, vol. 32, pp. 3606–3621, 2023.

[13] S. Rezayi, H. Dai, Z. Liu, Z. Wu, A. Hebbar, A. H. Burns, L. Zhao, D. Zhu, Q. Li, W. Liu, et al., ClinicalRadioBERT: Knowledge-infused few shot learning for clinical notes named entity recognition, in *Proc. 13th Int. Workshop on Machine Learning in Medical Imaging*, Singapore, 2022, pp. 269–278.

[14] M. Langberg and L. J. Schulman, Universal ε-approximators for integrals, in *Proc. 21st Annu. ACM-SIAM Symp. Discrete Algorithms*, Austin, TX, USA, 2010, pp. 598–607.

[15] J. M. Phillips, Coresets and sketches, in *Handbook of Discrete and Computational Geometry*, C. D. Toth, J. O' Rourke, and J. E. Goodman, eds., 3rd ed. New York, NY, USA: Chapman and Hall, 2017, pp. 1269–1288.

[16] O. Sener and S. Savarese, Active learning for convolutional neural networks: A core-set approach, in *Proc. 6th Int. Conf. Learning Representations*, Vancouver, Canada,https://dblp.uni-trier.de/db/conf/iclr/iclr2018.html#SenerS18, 2018.

[17] R. Pinsler, J. Gordon, E. Nalisnick, and J. M. Hernández-Lobato, Bayesian batch active learning as sparse subset approximation, in *Proc. 33rd Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2019, p. 571.

[18] Z. Borsos, M. Mutny, and A. Krause, Coresets via bilevel optimization for continual learning and streaming, in *Proc. 34th Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2020, p. 1247.

[19] R. Z. Farahani and M. Hekmatfar, *Facility Location*: *Concepts, Models, Algorithms and Case Studies*. Berlin, Germany: Springer Sci. & Bu. Media, 2009.

[20] M. E. Johnson, L. M. Moore, and D. Ylvisaker, Minimax and maximin distance designs, *J. Stat. Plann. Inference*, vol. 26, no. 2, pp. 131–148, 1990.

[21] C. J. Wu and M. S. Hamada, *Experiments*: *Planning, Analysis, and Optimization*. Hoboken, NJ, USA: John Wiley & Sons, 2011.

[22] C. Villani, *Optimal Transport*: *Old and New*. Berlin, Germany: Springer, 2009.

[23] G. Peyré and M. Cuturi, Computational optimal transport: With applications to data science, *Found. Trends® Mach. Learn.*, vol. 11, nos. 5&6, pp. 355–607, 2019.

[24] J. Zhang, P. Ma, W. Zhong, and C. Meng, Projection-based techniques for high-dimensional optimal transport problems, *WIREs*: *Comput. Stat.*, vol. 15, no. 2, p. e1587, 2022.

[25] C. Meng, Y. Ke, J. Zhang, M. Zhang, W. Zhong, and P. Ma, Large-scale optimal transport map estimation using projection pursuit, in *Proc. 33rd Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2019, p. 729.

[26] V. R. Joseph, E. Gul, and S. Ba, Maximum projection designs for computer experiments, *Biometrika*, vol. 102, no. 2, pp. 371–380, 2015.

[27] A. Beygelzimer, S. Dasgupta, and J. Langford, Importance weighted active learning, in *Proc. 26th Annu. Int. Conf. Machine Learning*, Montreal, Canada, 2009, pp. 49–56.

[28] D. Cohn, L. Atlas, and R. Ladner, Improving generalization with active learning, *Mach. Learn.*, vol. 15, no. 2, pp. 201–221, 1994.

[29] S. Dasgupta, D. Hsu, and C. Monteleoni, A general agnostic active learning algorithm, in *Proc. 21st Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2007, pp. 353–360.

[30] J. Smailović, M. Grčar, N. Lavrač, and M. Žnidaršič, Stream-based active learning for sentiment analysis in the financial domain, *Inf. Sci.*, vol. 285, pp. 181–203, 2014.

[31] D. Angluin, Queries revisited, *Theor. Comput. Sci.*, vol. 313, no. 2, pp. 175–194, 2004.

[32] A. Radford, L. Metz, and S. Chintala, Unsupervised

representation learning with deep convolutional generative adversarial networks, in *Proc. 4th Int. Conf. Learning Representations*, San Juan, Puerto Rico, https://dblp.uni-trier.de/db/conf/iclr/iclr2016.html#RadfordMC15, 2016.

[33] M. Huijser and J. C. van Gemert, Active decision boundary annotation with deep generative models, in *Proc. IEEE Int. Conf. Computer Vision*, Venice, Italy, 2017, pp. 5296–5305.

[34] Y. Gal and Z. Ghahramani, Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, in *Proc. 33rd Int. Conf. Machine Learning*, New York, NY, USA, 2016, pp. 1050–1059.

[35] C. E. Shannon, A mathematical theory of communication, *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.

[36] T. Yuan, F. Wan, M. Fu, J. Liu, S. Xu, X. Ji, and Q. Ye, Multiple instance active learning for object detection, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Nashville, TN, USA, 2021, pp. 5326–5335.

[37] Y. Siddiqui, J. Valentin, and M. Niesner, ViewAL: Active learning with viewpoint entropy for semantic segmentation, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020, pp. 9430–9440.

[38] W. H. Beluch, T. Genewein, A. Nurnberger, and J. M. Kohler, The power of ensembles for active learning in image classification, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 9368–9377.

[39] C. Campbell, N. Cristianini, and A. J. Smola, Query learning with large margin classifiers, in *Proc. 17th Int. Conf. Machine Learning*, Stanford, CA, USA, 2000, pp. 111–118.

[40] A. Kapoor, G. Hua, A. Akbarzadeh, and S. Baker, Which faces to tag: Adding prior constraints into active learning, in *Proc. 12th Int. Conf. Computer Vision*, Kyoto, Japan, 2009, pp. 1058–1065.

[41] B. Settles and M. Craven, An analysis of active learning strategies for sequence labeling tasks, in *Proc. Conf. Empirical Methods in Natural Language Processing*, Honolulu, HI, USA, 2008, pp. 1070–1079.

[42] S. Tong and D. Koller, Support vector machine active learning with applications to text classification, *J. Mach. Learn. Res.*, vol. 2, pp. 45–66, 2002.

[43] J. Kremer, K. S. Pedersen, and C. Igel, Active learning with support vector machines, *WIREs*: *Data Min. Knowl. Discov.*, vol. 4, no. 4, pp. 313–326, 2014.

[44] M. Bloodgood and K. Vijay-Shanker, Taking into account the differences between actively and passively acquired data: The case of active learning with support vector machines for imbalanced datasets, in *Proc. Human Language Technologies*: *The 2009 Annu. Conf. North American Chapter of the Association for Computational Linguistics*, Boulder, CO, USA, 2014, pp. 137–140.

[45] S. Dasgupta, Two faces of active learning, *Theor. Comput. Sci.*, vol. 412, no. 19, pp. 1767–1781, 2011.

[46] Y. Kim and B. Shin, In defense of core-set: A density-aware core-set selection for active learning, in *Proc. 28th ACM SIGKDD Conf. Knowledge Discovery and Data Mining*, Washington, DC, USA, 2022, pp. 804–812.

[47] K T. Fang, M. Q. Liu, H. Qin, and Y. D. Zhou, *Theory and Application of Uniform Experimental Designs*. Singapore: Springer, 2018.

[48] V. R. Joseph, Space-filling designs for computer experiments: A review, *Qual. Eng.*, vol. 28, no. 1, pp. 28–35, 2016.

[49] J. Zhang, C. Meng, J. Yu, M. Zhang, W. Zhong, and P. Ma, An optimal transport approach for selecting a representative subsample with application in efficient kernel density estimation, *J. Comput. Graph. Stat.*, vol. 32, no. 1, pp. 329–339, 2023.

[50] C. Villani, *Optimal Transport*: *Old and New*. Berlin, Germany: Springer, 2008

[51] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister, Sliced and radon Wasserstein barycenters of measures, *J. Math. Imaging Vis.*, vol. 51, no. 1, pp. 22–45, 2015.

[52] H. Wang, R. Zhu, and P. Ma, Optimal subsampling for large sample logistic regression, *J. Am. Stat. Assoc.*, vol. 113, no. 522, pp. 829–844, 2018.

[53] P. Ma, X. Zhang, X. Xing, J. Ma, and M. Mahoney, Asymptotic analysis of sampling estimators for randomized numerical linear algebra algorithms, in *Proc. the 23rd Int. Conf. Artificial Intelligence and Statistics*, Palermo, Italy, 2020, pp. 1026–1035.

[54] J. Guo, H. Gu, and M. Potkonjak, Efficient image sensor subsampling for DNN-based image classification, in *Proc. Int. Symp. Low Power Electronics and Design*, Seattle, WA, USA, 2018, p. 40.

[55] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal, Deep batch active learning by diverse, uncertain gradient lower bounds, in *Proc. 8th Int. Conf. Learning Representations*, Addis Ababa, Ethiopia, https://dblp.uni-trier.de/db/conf/iclr/iclr2020.html#AshZK0A20, 2020.

[56] D. Sculley, Web-scale k-means clustering, in *Proc. 19th Int. Conf. World Wide Web*, Raleigh, NC, USA, 2010, pp. 1177–1178.

[57] W. N. Hsu and H. T. Lin, Active learning by learning, in *Proc. AAAI Conf. Artificial Intelligence*, Austin, TX, USA, 2015, pp. 2659–2665.

[58] M. Zhang, Y. Zhou, Z. Zhou, and A. Zhang, Model-free subsampling method based on uniform designs, *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 3, pp. 1210–1220, 2024.

[59] F. F. Li, R. Fergus, and P. Perona, One-shot learning of object categories, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, 2006.

[60] M. Fink, Object classification from a single example utilizing class relevance metrics, in *Proc. 18th Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2004, pp. 449–456.

[61] Y. Chen, Z. Xiao, Y. Pan, L. Zhao, H. Dai, Z. Wu, C. Li, T. Zhang, C. Li, D. Zhu, et al., Mask-guided vision transformer for few-shot learning, *IEEE Trans. Neural Networks Learn. Syst*., 2024, doi: 10.1109/TNNLS. 2024.3418527.

[62] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE*, vol. 86, no. 11, pp. 2278−2324, 1998.

[63] A. Krizhevsky, *Learning Multiple Layers of Features from Tiny Images*. Toronto, Canada: University of Toronto, 2009.

[64] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, Randaugment: Practical automated data augmentation with a reduced search space, in *Proc. IEEE/CVF Conf.*

Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 2020, pp. 3008–3017.

[65] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, Random erasing data augmentation, in *Proc. AAAI Conf. Artificial Intelligence*, New York, NY, USA, 2020, pp. 13001–13008.

[66] K. L. Chung, An estimate concerning the kolmogroff limit distribution, *Trans. Am. Math. Soc.*, vol. 67, no. 1, pp. 36–50, 1949.

[67] M. Stein, Large sample properties of simulations using Latin hypercube sampling, *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.

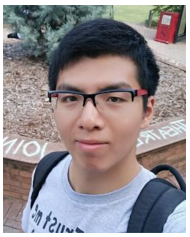
[68] J. Altschuler, F. Bach, A. Rudi, and J. Niles-Weed, Massively scalable sinkhorn distances via the nyström method, in *Proc. 33rd Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2019, p. 398.

Luyang Fang received the MEng degree in statistics from University of Wisconsin-Madison, USA in 2021. She is currently a PhD candidate in statistics at University of Georgia (UGA), USA. Her research interests include deep learning, non-parametric methods, and big data analytics.

Lin Zhao received the BEng degree from Northwestern Polytechnical University, China in 2017. He is currently a PhD candidate at Department of Computer Science, UGA, USA, under the supervision of Prof. Tianming Liu. His current research interests include deep learning and medical image analysis.

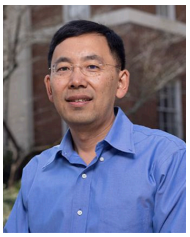
Tao Wang received the MEng degree in operations research from Georgia Institute of Technology, USA in 2021, where he is currently a PhD candidate in statistics. His current research interests include deep learning, functional regression, and large language model.

Tianming Liu is a distinguished research professor and a full professor of computer science at University of Georgia, USA. He has published 400+ research papers on these topics, his Google citation is over 16000+, and his H-index is 66. He is the recipient of NIH Career Award and NSF CAREER Award. He serves on the editorial boards of multiple international journals, including *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Medical Imaging*, *Medical Image Analysis*, *IEEE Transactions on Cognitive and Developmental Systems*, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *IEEE Reviews in Biomedical Engineering*, and *IEEE Journal of Biomedical and Health Informatics*. He is a fellow of American Institute of Medical and Biological Engineering (AIMBE) and was the general chair of MICCAI 2019. His research interests are brain imaging, computational neuroscience, brain-inspired artificial intelligence, and artificial general intelligence.



Cheng Meng received the PhD degree from UGA, USA in 2020. He is currently an assistant professor (tenure-track) at Institute of Statistics and Big Data, Renmin University of China. His research interests include numerical linear algebra, optimal transport problems, sufficient dimension reduction, nonparametric statistics, and machine learning.

Wenxuan Zhong received the BS degree in statistics from Nankai University, China, and the PhD degree from Purdue University, USA. After a postdoctoral fellowship in statistics and systems biology at Harvard, USA, she served as an assistant professor at University of Illinois Urbana-Champaign, USA from 2007 to 2013. Since 2013, she has been working at Department of Statistics, UGA, USA, where she is now the Georgia Athletic Association Professor. Her research focuses on the statistical methodology and theory development to face the striking new phenomena emerged under the big data regime. Over the past few years, she has established diverse extramurally funded research programs to overcome the computational and theoretical challenges arising from the big data analysis. The basic statistical researches are successfully applied in modern genomic, epigenetic, metagenomics, text-mining, and chemical sensing researches.



Ping Ma received the BS degree in statistics from Nankai University, China, and the PhD degree in statistics from Purdue University, USA. He then conducted postdoctoral research at Harvard University, USA. Since 2005, he has served as an assistant professor and then associate professor at University of Illinois at Urbana-Champaign, USA. He is currently a distinguished research professor at UGA, USA. His main research areas include the statistical theory and methods for very large samples, the development and application of data-driven nonparametric inverse modeling methods, and the development of gene regulatory network analysis.