# Gift from nature: Potential Energy Minimization for explainable dataset distillation

Zijia Wang[1], Wenbin Yang[1], Zhisong Liu[1], Qiang Chen[1], Jiacheng Ni[1], and Zhen Jia[1]

Dell Technologies OCTO Research Office

**Abstract.** Dataset distillation aims to reduce the dataset size by capturing important information from original dataset. It can significantly improve the feature extraction effectiveness, storage efficiency and training robustness. Furthermore, we study the features from the data distillation and found unique discriminative properties that can be exploited. Therefore, based on Potential Energy Minimization, we propose a generalized and explainable dataset distillation algorithm, called Potential Energy Minimization Dataset Distillation(PEMDD). The motivation is that when the distribution for each class is regular (that is, almost a compact high-dimensional ball in the feature space) and has minimal potential energy in its location, the mixed-distributions of all classes should be stable. In this stable state, Unscented Transform (UT) can be implemented to distill the data and reconstruct the stable distribution using these distilled data. Moreover, a simple but efficient framework of using the distilled data to fuse different datasets is proposed, where only a lightweight finetune is required. To demonstrate the superior performance over other works, we first visualize the classification results in terms of storage cost and performance. We then report quantitative improvement by comparing our proposed method with other state-of-the-art methods on several datasets. Finally, we conduct experiments on few-shot learning, and show the efficiency of our proposed methods with significant improvement in terms of the storage size requirement.

**Keywords:** dataset distillation, potential energy

## 1 Introduction

Energy consumption in deep learning model training is a big concern in real-world applications [29], one of the solutions for this problem is knowledge distillation [10]. It transfers the knowledge from a deep complex model to a simple one, hence people can save the running time. Another direction is dataset distillation [38, 30, 9]. It tries to synthesize some data samples or features to summarize information in the original huge dataset and use these synthesized data to train models more efficiently.

However, most dataset distillation algorithms mainly focus on the information in the models and try to reproduce the ability by utilizing small models or

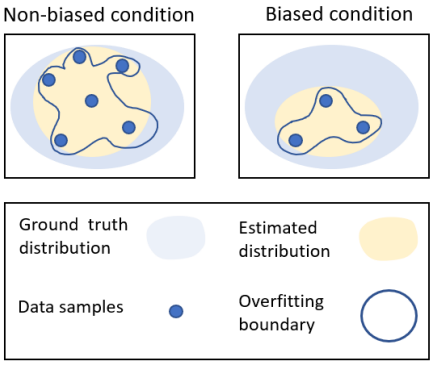



Fig. 1: If the distilled data is not biased (top left), even the overfitting boundary generated by deep learning models, which is common in few-shot learning, can better represent the ground truth distribution than the biased one(top right).For properly estimated distribution (yellow circle), the condition is the same.

small datasets. The properties of dataset are ignored. In the naive dataset distillation[39], the dataset properties are explored by "black box" meta-training process, which is not robust and explainable. It should be noticed that, in the setting of dataset distillation, which is essentially a few-shot learning scenario, models tend to overfit such small dataset[42]. The obtained distilled dataset is only adequate for networks used in the training iterations of the distillation process.

Therefore, to avoid overfitting and the rigid training constraints, we hope to distill the data with the consideration of inherent dataset statistical properties. Noticed that the dimension in feature space is much lower, so it's easier to be calibrated [41]. Also, mean and variance of Gaussian distribution could be transferred across similar classes [26], all the transformation are performed in the feature space (Figure 2 left), so we use Gaussian distribution to calibrate the feature distribution. In the final calibrated space, the features of different classes are expected to be far enough while features of similar classes should be in a distribution with low variance. The second one could be achieved in the distribution calibration stage, the ideal distance in the first requirement, however, is hard to determine. A large distance could make classification more accurate, but it makes the feature space sparse. On the other side, a small distance would cause the decrease in accuracy. Therefore, we choose to adapt the idea of potential energy stable equilibrium, this equilibrium exists if the net force is zero, any changes in the system would increase the potential energy [8]. This stable equilibrium describes the perfect distance between the center of each class(Figure2 right). In the stable system, the features from different classes could be easily classified using a simple classifier. What's more, the centers and edge points can be the distilled data which is the best subset of the original

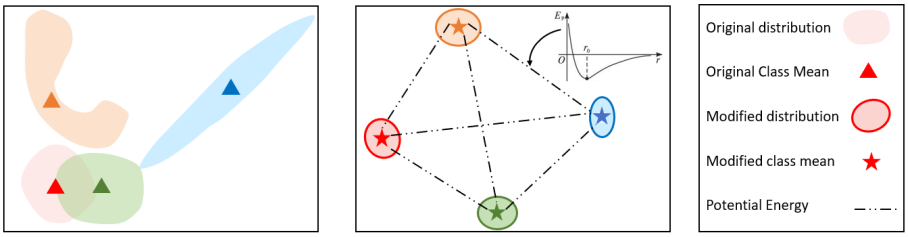



Fig. 2: In original feature space (left), the distributions are random and hard to classify. After the distribution calibration and transformation, a stable state is achieved(right) in which the distribution is tight and all the classes are 'stable' according to the potential energy stable equilibrium. Distilled data could be easily chosen based on this stable system, these distilled data, along with the transformation matrix, could be used to fuse dataset and evaluate data quality.

dataset considering the data distribution, then if new models are trained on these distilled data, the accuracy of models trained on original dataset could be recovered. In experiments, besides the original ability of dataset distillation (accuracy recovery), visualization results show that our the chosen images are explainable with good diversity, this explainability can be further exploited to make more use of these distilled samples. We also use experiment results to show that simple classifier with our dataset distillation strategy can perfectly handle data fusion and data quality evaluation tasks. As an extension to current dataset distillation algorithm, our algorithm can also outperform SOTA results in few-shot learning. Overall, our contributions are:

- Potential Energy Minimization based Dataset Distillation (PEMDD). Applying the concept of PEM and distribution calibration to the feature vectors to find the stable state in feature space. Therefore, dataset could be distilled while, to the maximum extent, avoiding harming the distribution reproduction. In our model, only few parameters are added. What's more, this strategy is also shown to be useful in few-shot learning.
- Unscented Transformation (UT) for dataset distillation. UT is used to distill the data which could be used to reconstruct the stable distribution from distilled samples. Then the reconstructed distribution could be used to perform the up-sampling and other downstream tasks.
- Framework for the applications using dataset distillation and our stable system. Based on the distilled data and the transformation matrix of stable system, we propose frameworks to fuse dataset in three scenarios: (i) 2 datasets share same classes; (ii) new dataset contains new classes; (iii) 2 datasets share some same classes while new dataset contains new classes. What's more, PEM solution for few-shot learning is also tested in this paper and the results are good.

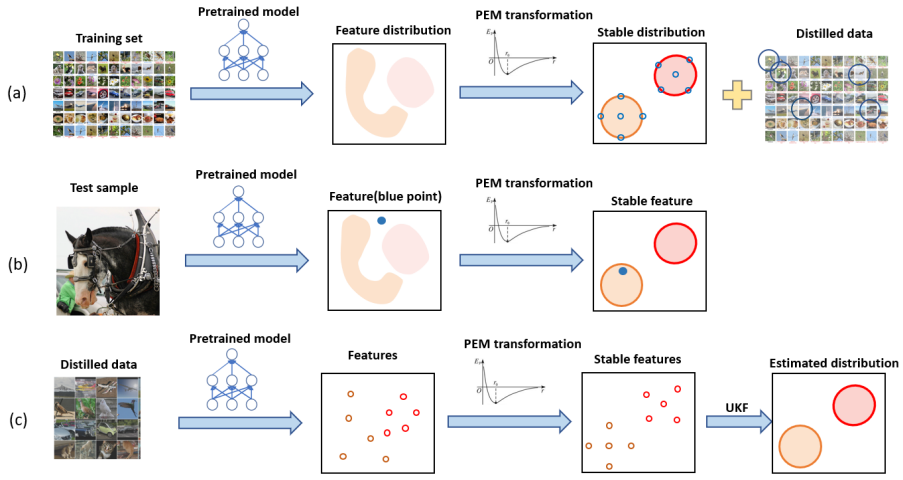



Fig. 3: Workflow for PEM-based dataset distillation. **(a) Training process.** This part shows the training process in which the PEM transformation model is trained, then the stable distribution is derived and the distilled data is selected from the original dataset based on the sampling strategy in stable distribution system. **(b) Testing process.** This part illustrates the testing process, the test sample is also first transformed into the feature space and then converted into the stable system using the trained PEM transformation model. **(c) Distilled data for applications.** This part demonstrates the basic strategy of using the distilled data, they could be used to reconstruct the distributions for each class while ideally, they are same to the distributions in the stable system. This strategy could be flexibly adapted into different scenarios.

In the following part of this paper, some preliminaries and related works are listed in section 2, then the algorithm details of PEM-based dataset distillation is demonstrated in the beginning of section 3, then the rest of that section shows the solution for few-shot learning and dataset fusion. Finally, experiment details are described in section 4. Concretely, in the experiment, we use few-shot learning, where our algorithm could achieve competitive results compared with SOTA results, to show the power of stable state derived by PEMDD, then the dataset fusion experiment shows the advantages of our distilled data by almost recovering the classification accuracy with only few data samples kept.

## 2 Related work

### 2.1 Dataset distillation

Computational cost in deep learning becomes more and more expensive, model compression starts to attract much attention of researchers [1, 23, 11]. Dataset

distillation[38] is one of them and was first introduced in the inspiration of network distillation [10], some theoretical works illustrate the intuition of dataset distillation [31] and extend the initial dataset distillation algorithm [30]. Besides these works, many works have shown impressive result in generating or selecting a small number of data samples [7, 32, 33, 2, 27, 3] utilizing active learning, core set selection, etc.

Although our idea is also to find the core dataset, we borrow the idea of dataset distillation to deal with data depending on the network information and select the core dataset based on the hyper-ball (calibrated distribution) we generated in the stable feature space. What's more, the advantages of the stable system make it possible to use distilled data and transformation matrix to perform more downstream tasks.

### 2.2 Distribution calibration

As shown in figure 1, some metrics could be used to estimate the distribution based on the distilled data [13, 37], but most of them assume the kind of distribution is known. To better reconstruct the distribution based on distilled data, the data distribution should follow a specific distribution (Gaussian in this paper). Many papers tried to calibrate the distribution of the data for different purposes [42, 28, 24], the main idea is to calibrate the data distribution into a regular and tight distribution.

However, the calibrated distribution in these algorithms cannot be directly used in the dataset distillation setting. In dataset distillation, we want the distilled data to maximally contain the information in original dataset without being affected by other classes. Furthermore, in the application phase, the addition of new data samples would make the system unstable if the distance of each class if unstable, so we use the concept of potential energy to avoid such risk to the greatest extent possible.

### 2.3 Potential Energy

Potential energy is a simple concept in physics. In this theory, there exists an distance between two particles ($r_0$). If the distance becomes closer, the resultant force is attractive while the resultant force changes to be repulsive force. Only when the resultant force is zero, the potential energy is lowest, which means that the system is stable [4, 18].

In this paper we choose to adapt this concept to find the perfect distance between 2 feature vectors. We use molecular potential energy to optimize the position of centroids to make them easy to classify while not being too far, then atomic potential energy is used to optimize the position of features of same class to make them close enough.

# 3 Method

In this section, the basic problem for dataset distillation is first defined in section 3.1 and the solution is revealed in section 3.2, then a concrete scenario is shown to demonstrate the application framework in section 3.3. Finally, our algorithm is applied into the few-shot learning problems and a thorough analysis is demonstrated.

## 3.1 Problem Definition

Given a labelled dataset $\mathcal{D} = \{x_i, y_i\}$ where $x_i \in \mathbb{R}$ is the raw data sample and $y_i \in \mathbb{C}$ is the corresponding labels with $\mathbb{C}$ denoting the set of classes. Then assume that a pretrained model $\mathcal{M}$ which could extract features $\mathcal{F} = \{f_i\}$ from $\mathcal{D}$ where $f_i$ is the feature vector of $x_i$. The goal of dataset distillation is to select a few data samples which capture the most important distribution properties of $\mathcal{D}$.

To realize the dataset distillation, we suggest to learn a transformation $\mathcal{T}$ to transfer $\mathcal{F}$ into an new embedding space where features from same class become more compact while features from different classes are in a 'moderate' distance. Then, based on the transferred distribution, a subset of original dataset $\mathcal{D}$ are collected as distilled dataset $\mathcal{S}$.

## 3.2 PEM-based transformation

**Tukey's Ladder of Powers Transformation** To make the feature distributions more regular, i.e. be more like Gaussian, the first step in PEM-based transformation is adopting the Tukey's Ladder of Powers Transformation [34] to reduce the skewness of distributions.

The function of Tukey's Ladder of Powers Transformation can be varied based on the configuration of the power. The formulation of this transformation can be expressed as:

$$\hat{x} = \begin{cases} x^P & if\ P \neq 0 \\ log(x) & if\ P = 0 \end{cases} \tag{1}$$

where $P$ is the hyper-parameter which could control the way of distribution regularization. To recover the feature distribution, $P$ should be set to 1. If the $P$ decreases, the distribution becomes less positively skewed and vice versa.

**Potential Energy minimization** Considering a linear transformation with weight $\mathbf{W}_T$ as

$$\mathcal{F}_s = \mathbf{W}_T \mathcal{F} \tag{2}$$

where $\mathcal{F}_s$ is the desired feature, $\mathcal{F}$ is the input feature. we hope to find a suitable distance among classes to ensure the diversity of the latent feature. To achieve this goal, recall the potential energy expression [4], which may have different forms. In this paper we use the following formula:

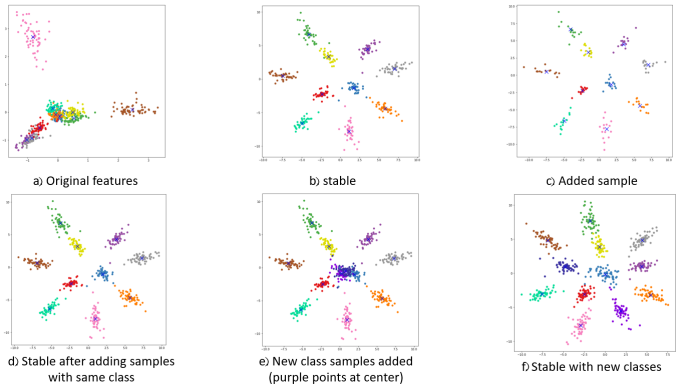



Fig. 4: Visualization for dataset fusion with PEM in CUB-200. a) shows the original feature distribution, after PEM transformation, they are stable in b). In c), some new samples whose class are same with the classes in b) are added, and in d) they are perfectly handled using PEM. In e), some samples of new classes (purple points) are added, then the PEM transformation results are shown in f) after fine-tune, they are in the stable equilibrium again.

$$E(r) = \frac{1}{r^3} - \frac{1}{r^2} \tag{3}$$

where $r$ is the distance between two particles. Here, $r_0$ is the optimal distance for minimal potential energy. Then we adapt this equation and derive our loss function to learn linear transformation $\mathbf{W}_T$ that minimizes the 'potential energy' between the extracted features of every pair of data points:

$$L = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left[ \frac{1}{(\gamma_{ij} d_{ij} + b_0)^3} - \frac{1}{(\gamma_{ij} d_{ij} + b_0)^2} \right] \tag{4}$$

where $d_{ij} = dis(\mathbf{W}_T f_i, \mathbf{W}_T f_j)$ with $dis(\cdot, \cdot)$ representing the Euclidean distance, $f_i, f_j$ are the input features, $N$ is the number of data samples. The hyperparameter $b_0$ $(0 < b_0 < r_0)$ is introduced to improve the numerical stability of PE. $\gamma_{ij}$ is the function to control the properties of loss, which can be defined as:

$$\gamma_{ij} = \begin{cases} \tau_0 & if\ y_i = y_j \\ \tau_1 & if\ y_i \neq y_j \end{cases} \tag{5}$$

where $\tau_0$, $\tau_1$ are the inter class and inner class weights, respectively. In this paper, we set $0 < \tau_1 < \tau_0$.

**Process of Dataset Distillation** In figure 3 (a), a co-stable system with desired distribution is derived after the optimized transformation. Then some data

points can be sampled based on the stable equilibrium. Considering the inner-class Gaussian-like properties, we use the data sampling strategy in UT [36, 37] to get the distilled data points $\mathcal{S}$ (sigma points in UT) and their corresponding weights $\omega$ for distribution reconstruction. Here there are two sets of weights $\omega_m$ and $\omega_c$, $\omega_m$ is used to recover the means of the original distribution while the $\omega_c$ is for the reconstruction of the covariance matrix.

For each class, the first sample in sigma points set is $\mathcal{S}[0] = \mu$ with $\mu$ representing the mean, then the other samples are sampled as following:

$$\mathcal{S}[i] = \begin{cases} \mu + V_i & for\ i = 1, ..., d-1 \\ \mu - V_{i-d} & for\ i = d, ..., 2d \end{cases} \tag{6}$$

where variance $V_i = \sqrt{(d+\lambda)\Sigma(:,i)}$, $d$ stands for the dimension of the features, $\lambda$ is the scaling parameter and $\Sigma(:,i)$ is the $i_{th}$ column of the covariance matrix while the covariance matrix could be easily derived with data samples.

For this sequence $\mathcal{S}$, the corresponding weights $\omega_m$ for mean estimation with sigma set can be calculated as:

$$\omega_m^{[i]} = \begin{cases} \frac{\lambda}{d+\lambda} & if\ i = 0 \\ \frac{1}{2(d+\lambda)} & if\ i = 1, ..., 2d \end{cases} \tag{7}$$

while $\omega_m^{[i]}$ is the weight for the $i_{th}$ element in $\mathcal{S}$, then the equation of the weight $\omega_c$ for calculating the covariance is:

$$\omega_c^{[i]} = \begin{cases} \omega_m^{[0]} + H & if\ i = 0 \\ \frac{1}{2(d+\lambda)} & if\ i = 1, ..., 2d \end{cases} \tag{8}$$

In equations 7 and 8, $\lambda = \alpha^2(d+k) - d$ and $H = 1 - \alpha^2 + \beta$. To control the distance between the sigma points and the mean, we could adjust $\alpha \in (0, 1]$ and $k \geq 0$. In some literature [36, 37], $\beta = 2$ is an optimal choice for Gaussian.

It should be noticed that, the sigma points $\mathcal{S}$ are just sampled in the latent feature space. Finally, for data sample selection, we suggest modeling the dataset distillation as an assignment problem and select the data samples according to the distance (such as the Euclidean distance) between real data features and the sigma points. When considering only one-to-one correspondences modeled as bipartite graph matching, Hungarian algorithm[12] can be used to solve the assignment problem in polynomial time. After the bipartite graph matching, sigma points will be assigned to real data samples.

**Classification for new samples** As shown in figure 3 (b), when a new test sample comes in, it is first transformed by the pretrained model (feature extractor) and our PEM model, then a simple classifier like Logistic classifier[21] could be used to classify this sample effectively and robustly.

### 3.3 How to fuse datasets

In this part, we extend the basic problem defined in section 3.1 to demonstrate the solution in some more concrete application scenarios in dataset fusion.

**Dataset fusion problem definition** Recall the problem defined in section 3.1, a distilled dataset $\mathcal{S}$ is derived from the labelled dataset $\mathcal{D} = \{x_i, y_i\}$. Now assuming a new dataset $\mathcal{D}^{new} = \{x_i^{new}, y_i^{new}\}$ appears, where $x_i^{new} \in \mathbb{R}$ is the raw data and $y_i^{new} \in \mathbf{C}^{new}$ is the data label with $\mathbf{C}^{new}$ representing the new class categories.

The goal in this section is to find a fine-tuned PEM transformation $\mathcal{T}^{new}$ to realize a new stable state for fused dataset $\mathcal{D}^{fuse} = \mathcal{S} \cup \mathcal{D}^{new}$. There are mainly 2 settings in this problem. The first setting is that two datasets share exactly same classes, i.e. $\mathbf{C}^{new} = \mathbf{C}$. The other one setting considers the $\mathbf{C}^{new} \neq \mathbf{C}$.

**Distribution fusion** In both setting, the fusion process is similar to the process described in section 3.2. At first, to estimate the fused statistics more accurately ,we will up-sampling some feature-points. Considering the distribution for each class after PEM based transformation are Gaussian-like, features can be easily generated with re-parameter trick. Then a PEM training process as shown in figure 3 (a) is performed on $\mathcal{D}^{new}$ to get the stable distribution.

### 3.4 Few-shot learning

In PEM, the inner-class compactness and inter-class diversity of the latent features is the foundation of our success in dataset distillation. Therefore, we extend the PEM strategy in few-shot learning setting to show our advantages in data property exploitation.

**Few-shot learning problem definition** A few-shot learning problems can be a simple extension of the problem defined in previous section, where the samples in $\mathcal{D}^{new}$ are quite few. Tasks in few-shot learning could be called N-way-K-shot [35] ,where there are $N$ classes in $\mathbf{C}^{new}$ and $K$ labelled samples for each class.

**Few-shot learning solution** The solution for few-shot learning can be modeled as an simplified version of PEMDD. At first, the PEM transformation is trained using very few samples to get the stable state, then this transformation is used to transform the test features into the latent feature space. Then, a simple classifier, such as logistic regression, can be utilized for label prediction.

We will show more details about our implementation of few-shots learning in section 4.

### 3.5 Analysis

Intuitively, the proposed PEM framework share some common ideas with the well-known Fisher Discriminant Analysis (FDA) [19]. Both methods try to increase the diversity between classes and reduce the diversity within classes. However, PEM is constructed based on the steady state through the PE function, which is discovered in physics and makes the separation between classes more balanced. Meanwhile, our PEM also guarantees the existence of an stable state for all systems. All these reasons above allow the PEM-based method to maintain good performance on few-sample data set compared with works like FDA.

Our method is also different from the dataset distillation in [39]. For the dataset distillation in [39], its aim is to replicate the performance of the entire dataset from the synthetic points, but our method is trying to select some informative samples and their corresponding linear transformation weights. Our method allows more flexible selection of data sample, while the dataset distillation must predefine the synthesized number before training. Also a Hungarian algorithm was used section 3.2, one can also increase the sample to be selected by an unbalanced Hungarian algorithm. For a classical Hungarian problem for data sample selection, the complexity is just $O(d^3)$.

Last but not least, our method can also be adopted as a first principle to learn **diverse and discriminative** features for down-stream tasks.

## 4 Experiment and discussion

### 4.1 Experiment setup

**Datasets** In this paper, miniImageNet [22] and CUB-200 [40] are used to evaluate our algorithm.

For miniImageNet, in few-shot learning validation experiment, all classes are split into 64 base classes, 16 validation classes and 20 new classes as [22] did in their work. However, in data fusion experiment and visualization part, only 10 classes are selected from the miniImageNet because of the reality and visualization simplicity, the train-test-split conditions are illustrated in the experiment part (section 4.3).

CUB-200 is a fine-grained benchmark for few-shot learning. There are 11,788 images with size $84 \times 84 \times 3$ for 200 different classes of birds. These classes are spilt into 100 base classes, 50 validation classes and 50 novel classes [5].

**Evaluation metric** For different tasks, we adopted different metrics. In few shot learning setting, top-1 accuracy is used to evaluate our strategy in 5way1shot and 5way5shot settings for both 2 datasets [42]. In Data fusion setting, all experiments are variants of the image classification task, so top 1 accuracy is used to evaluate the performance of different strategy.

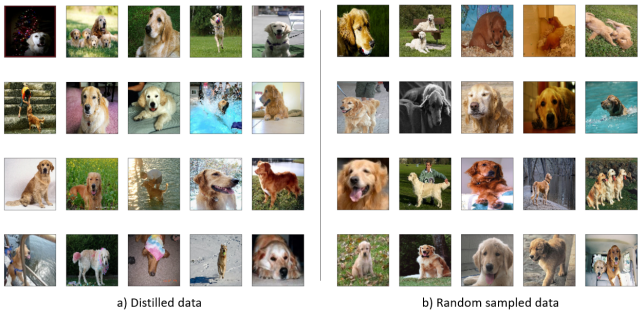



Fig. 5: Visualization of dataset distillation for dog.

**Implementation details** In all experiments, we use pretrained WideResNet [17] to extract the features, features are extracted from the layer before fully connected layer with a ReLU activation function which makes the output non-negative [42]. The parameters of $\tau_0$,$\tau_1$,$b_0$ are set to 1, 0.1, 0.3, respectively. The embedding dimension size is set to 12, therefore, the sigma points number for each class will be 25. We also follow the Tukey transformation settings in [42].

In few-shot learning part, we follow the solution described in previous section to deal with the extracted features, nearest center is used to be the classifier.

In data fusion part, we choose logistic classifier to represent the current SOTA results and compare our results with them. We use the logistic classifier implementation of scikit-learn[20] with the default settings.

### 4.2 Empirical Understanding of PEMDD

Table 1a shows the experiment results in a regular setting. In this setting, for each class, 450 samples are used to train the PEM model and 150 samples are used as test set. As shown in table 1a, when the training samples are reduced to 25%, no matter what strategy is used to choose samples, a huge decline in the accuracy occurs. Then in our strategy, we adapt Logistic [20] as the classifier. With our strategy (PEM with sampling), we sample 1000 points for each class in test phase, the results are almost recovered while a fine-tune PEM could enhance the performance (from 0.943 to 0.988).

Then to further illustrate the advantages of our results, we visualize the 20 “selected” images selected from MiniImageNet dataset for the ”dog” classes (figure 5). We observe that the selected images are much more diverse and representative than those selected randomly from the dataset (with random selection program), indicating such PEM-based distilled images can be used as a good “summary” of the dataset.

At last, we compare to the state of the art dataset distillation method[39]. We test our method on CIFAR-10 dataset. The model is identical to the ones used in[39], which can achieve about 80% test accuracy on CIFAR-10 in a fully

| Method | CUB-200 | miniImageNet |
|---|---|---|
| Full data | 1.000 | 0.938 |
| Class-based random (10%) | 0.378 | 0.258 |
| Most remote (10%) | 0.463 | 0.314 |
| **PEM** | 0.981 | 0.923 |
| **PEM+sampling** | 0.943 | 0.911 |
| **PEM+sampling+PEM** | 0.988 | 0.941 |

(a) Experiment results for basic classification task.

| Method | 50 | 100 | 150 | 200 |
|---|---|---|---|---|
| All random | 0.091 | 0.132 | 0.158 | 0.165 |
| Class-based random | 0.092 | 0.143 | 0.174 | 0.220 |
| K-means | 0.105 | 0.184 | 0.223 | 0.347 |
| dataset distillation(random init) | - | 0.368 | - | - |
| dataset distillation(fixed init) | - | 0.540 | - | - |
| **PEMDD** | 0.247 | 0.519 | 0.614 | 0.719 |

(b) Experiment results compared with original dataset distillation algorithm.

Table 1: Basic experiments of PEMDD.

supervised setting. The dataset distillation will synthesis 100 pictures(10 pictures for each class). The PEM based method, K-means and Random selection will select 50-200 pictures. The embedding dimension size of PEM is set to 10. All the result is shown in Tab.1b.

### 4.3 Data Efficiency Application

In this section, we perform a series of experiments to test our strategy on different settings.

In table 2a, we equally split the 10-th class. 300 samples are used to train the PEM and others are used as new samples. Other settings are same as previous one. In our setting, these two sets have different distribution on the 10-th class, therefore conventional method which builds on the i.i.d assumption may suffer from the performance decreasing. As shown in table 2a, a simple fine-tune after adding new data, the PEM based method could get a good test accuracy performance (0.980).

Then, we remove the 8-th class in the training dataset and treat it as a totally new class. As shown in table 2b, the PEM based with fine-tune could also give good result (0.981) in this setting.

| Method | CUB-200 | miniImageNet |
|---|---|---|
| Full data | 1.000 | 0.82 |
| Class-based random (10%) | 0.176 | 0.238 |
| **PEM+Sampling** | 0.516 | 0.610 |
| **PEM+Sampling+PEM** | 0.980 | 0.932 |

(a) Experiment results for adding new samples with same classes.

| Method | CUB-200 | miniImageNet |
|---|---|---|
| Full data | 1.000 | 0.938 |
| Class-based random (10%) | 0.168 | 0.185 |
| **PEM + Sampling** | 0.588 | 0.681 |
| **PEM + Sampling+PEM** | 0.981 | 0.940 |

(b) Experiment results for adding new classes.

Table 2: Data efficiency of PEMDD.

To illustrate the effectiveness of PEM in above three tasks, we visualize the feature on CUB-200 dataset with t-SNE[16] before and after the PEM optimization. In figure 4, (a),(c),(e) show the distribution condition before the PEM

Table 3: Experiment results of our few-shot learning solution.

| Methods | miniImageNet | | CUB | |
|---|---|---|---|---|
| | 5way1shot | 5way5shot | 5way1shot | 5way5shot |
| LEO[25] | 63.80 | 77.59 | - | - |
| Negative-Cosine[14] | 62.33 | 80.94 | 72.66 | 89.40 |
| TriNet [6] | 58.12 | 76.92 | 69.61 | 84.10 |
| E3BM [15] | 63.80 | 80.29 | - | - |
| LR with DC [42] | 68.57 | 82.88 | 79.56 | 90.67 |
| S2M2_R [17] | 64.93 | 83.18 | 80.68 | 90.85 |
| **PEM-S** | **58.48** | **82.75** | **72.81** | **90.55** |

transformation. In (a), the distributions are hard to classify while in (b), features among classes are diverse and easy to classify. Similarly, in (c) and (e), when the new samples come in, the clusters become unstable and PEM could stabilize them again, as shown in (d) and (f). Overall, in the most stable state, all classes have a "safe" distance.

### 4.4 Few-shot Application

In the experiment, a PEM transformation is learned based on few-shot training samples to get the stable state, then this transformation is used in the testing phase to transform the test set features into stable state.. Our method, PEM-S,contains the up-sampling process mentioned before. The sampling number is 500. The experiment results are summarized in table 3.

As shown in the table 3, our results achieve SOTA in 5way5shot learning, though not has the best performance in 5way1shot learning. This is because our strategy partly rely on the inner class compactness and the 1-shot setting cannot provide such information for our PEM-S method.

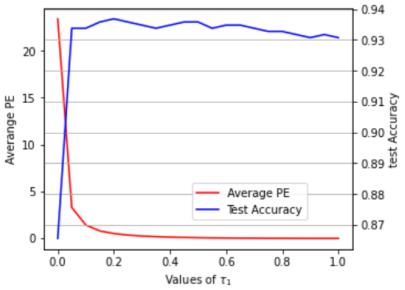
### 4.5 Hyper-parameters

To show the effeteness and robustness of the proposed method, we run some Hyper-parameters studies in the baseline problems with mini-Imagenet dataset.

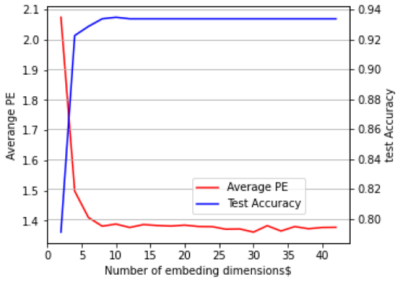
**Properties of PE function** The Properties of the PE function is mainly affected by the $\gamma_{ij}$. For simplicity, we fixed the value of $\tau_0$ to 1, and run a comprehensive study of the selection $\tau_1$. Figure 6a illustrates the average PE value $(1/N(N-1))$ of the PE of the whole stable distribution) at training dataset and test accuracy. It can be witnessed that appropriate values of $\tau_1$ can make the system have lower PE and better test accuracy.

**Embedding size** We test our method with different embedding size. From figure 6b, one can found that the embedding size may slightly affect the PE values when $> 7$.

**Different backbone networks** Table 5 shows the consistent performance on different feature extractors, i.e, five convolutional layers (conv4), AlexNet, vgg16, resnet18, WRN28(Baseline). It can be concluded that the PEM based




(a) The effect of different values of $\tau_1$.




(b) The effect of the number of embedding dimensions.

Fig. 6: Hyper-parameters testing

method can achieves at almost 10× accuracy improvement for conv4,AlexNet and vgg16. Moreover, the WRN28 achieve the best performance. It is because of that WRS28 is a semi-supervised method that considers the main-fold information of the dataset.

Table 4: Experiment results with different backbones.

| Backbones | Class-based random | PEM-based |
|---|---|---|
| conv4 | 0.089 | 0.858 |
| AlexNet | 0.098 | 0.913 |
| vgg16 | 0.106 | 0.905 |
| resnet18 | 0.151 | 0.921 |
| WRN28(Baseline) | 0.258 | 0.941 |

## 5 Conclusion and future work

In this paper, we propose a PEM-based framework for DD and few-shot learning settings. PEM can help features to achieve a stable state in the new embedding space. In the new embedding space, the features will represent inner class compactness and inter class diversity, which is the foundation of UT based DD. Experiments results in multi-scenarios reveals the superiority of our PEM strategy. our PEM-based framework shows that the limited data could be used to recover or even outperform the performance of original data while largely reducing the computation costs and storage costs. Future work will explore more application scenarios for distilled data. Moreover, statistical properties may also be used to generate samples, instead of choosing samples from the dataset. What's more, the PEM can be introduced as a first principle for machine learning problems instead of the conventional 'black box' models.

## References


1. Ba, L.J., Caruana, R.: Do deep nets really need to be deep? arXiv preprint arXiv:1312.6184 (2013)
2. Bachem, O., Lucic, M., Krause, A.: Practical coreset constructions for machine learning. arXiv preprint arXiv:1703.06476 (2017)
3. Bezdek, J.C., Kuncheva, L.I.: Nearest prototype classifier designs: An experimental study. International journal of Intelligent systems **16**(12), 1445–1473 (2001)
4. C, J.: Textbook Of Engineering Physics. Prentice-Hall Of India Pvt. Limited (2009), https://books.google.com/books?id=DqZlU3RJTywC
5. Chen, W.Y., Liu, Y.C., Kira, Z., Wang, Y.C.F., Huang, J.B.: A closer look at few-shot classification. arXiv preprint arXiv:1904.04232 (2019)
6. Chen, Z., Fu, Y., Zhang, Y., Jiang, Y.G., Xue, X., Sigal, L.: Multi-level semantic feature augmentation for one-shot learning. IEEE Transactions on Image Processing **28**(9), 4594–4605 (2019)
7. Cohn, D.A., Ghahramani, Z., Jordan, M.I.: Active learning with statistical models. Journal of artificial intelligence research **4**, 129–145 (1996)
8. Dina, Z.e.a.: Force and potential energy (Jun 2019), https://chem.libretexts.org/@go/page/2063
9. Hariharan, B., Girshick, R.B.: Low-shot visual object recognition. CoRR **abs/1606.02819** (2016), http://arxiv.org/abs/1606.02819
10. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network (2015)
11. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
12. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)
13. Larrañaga, P., Lozano, J.A.: Estimation of distribution algorithms: A new tool for evolutionary computation, vol. 2. Springer Science & Business Media (2001)
14. Liu, B., Cao, Y., Lin, Y., Li, Q., Zhang, Z., Long, M., Hu, H.: Negative margin matters: Understanding margin in few-shot classification. In: European Conference on Computer Vision. pp. 438–455. Springer (2020)
15. Liu, Y., Schiele, B., Sun, Q.: An ensemble of epoch-wise empirical bayes for few-shot learning. In: European Conference on Computer Vision. pp. 404–421. Springer (2020)
16. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
17. Mangla, P., Kumari, N., Sinha, A., Singh, M., Krishnamurthy, B., Balasubramanian, V.N.: Charting the right manifold: Manifold mixup for few-shot learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2218–2227 (2020)
18. McCall, R.: Physics of the Human Body. Johns Hopkins University Press (2010), https://books.google.com/books?id=LSyC41h6CG8C
19. Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Mullers, K.R.: Fisher discriminant analysis with kernels. In: Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468). pp. 41–48. Ieee (1999)
20. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. the Journal of machine Learning research **12**, 2825–2830 (2011)

21. Peterson, L.E.: K-nearest neighbor. Scholarpedia **4**(2), 1883 (2009)
22. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning (2016)
23. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550 (2014)
24. Rueda, M., Martínez, S., Martínez, H., Arcos, A.: Estimation of the distribution function with calibration methods. Journal of statistical planning and inference **137**(2), 435–448 (2007)
25. Rusu, A.A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., Hadsell, R.: Meta-learning with latent embedding optimization. arXiv preprint arXiv:1807.05960 (2018)
26. Salakhutdinov, R., Tenenbaum, J., Torralba, A.: One-shot learning with a hierarchical nonparametric bayesian model. In: Proceedings of ICML Workshop on Unsupervised and Transfer Learning. pp. 195–206. JMLR Workshop and Conference Proceedings (2012)
27. Sener, O., Savarese, S.: Active learning for convolutional neural networks: A core-set approach. arXiv preprint arXiv:1708.00489 (2017)
28. Song, H., Diethe, T., Kull, M., Flach, P.: Distribution calibration for regression. In: International Conference on Machine Learning. pp. 5897–5906. PMLR (2019)
29. Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for deep learning in nlp (2019)
30. Sucholutsky, I., Schonlau, M.: Improving dataset distillation. CoRR **abs/1910.02551** (2019), http://arxiv.org/abs/1910.02551
31. Sucholutsky, I., Schonlau, M.: 'less than one'-shot learning: Learning n classes from m¡n samples (2020)
32. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. Journal of machine learning research **2**(Nov), 45–66 (2001)
33. Tsang, I.W., Kwok, J.T., Cheung, P.M., Cristianini, N.: Core vector machines: Fast svm training on very large data sets. Journal of Machine Learning Research **6**(4) (2005)
34. Tukey, J.W., et al.: Exploratory data analysis, vol. 2. Reading, Mass. (1977)
35. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. arXiv preprint arXiv:1606.04080 (2016)
36. Wan, E.A., Van Der Merwe, R.: The unscented kalman filter for nonlinear estimation. In: Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373). pp. 153–158. Ieee (2000)
37. Wan, E.A., Van Der Merwe, R., Haykin, S.: The unscented kalman filter. Kalman filtering and neural networks **5**(2007), 221–280 (2001)
38. Wang, T., Zhu, J., Torralba, A., Efros, A.A.: Dataset distillation. CoRR **abs/1811.10959** (2018), http://arxiv.org/abs/1811.10959
39. Wang, T., Zhu, J.Y., Torralba, A., Efros, A.A.: Dataset distillation. arXiv preprint arXiv:1811.10959 (2018)
40. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-ucsd birds 200 (2010)
41. Xian, Y., Lorenz, T., Schiele, B., Akata, Z.: Feature generating networks for zero-shot learning. CoRR **abs/1712.00981** (2017), http://arxiv.org/abs/1712.00981
42. Yang, S., Liu, L., Xu, M.: Free lunch for few-shot learning: Distribution calibration (2021)