ECE1512 - Project B: Dataset Distillation: A Data-Efficient Learning Framework

*Due Date: Friday, December 08, 6:00 PM EST

Lanzhi Xiao
Master Engineer of MIE
University of Toronto
Toronto, Canada
uid: 1004569362

Bocheng Zhang
Master Engineer of ECE
University of Toronto
Toronto, Canada
uid: 1009969517

***Abstract*—This document is a report for ECE1512 Project B, project use Data Distillation (DD) to reduce computation in cost-based model-space. The data set MNIST and MHIST will be used in this project**

I. Introduction

Dataset Distillation (DD) is an method to reduce the high computing source challenge of deep learning models trained on large-scale, by created a synthetic dataset. In this project there are two part. For the first part, we will use the dataset distillation with gradient matching to learn a synthetically small dataset for the MNIST and MHIST datasets, train networks from scratch on the condensed images, and then evaluate them on the real testing data. Based on our findings, it has been observed that with a reduced training duration, the model accuracy achieved using synthetic images is notably commendable. This technique holds significant potential for application across various domains within the field of machine learning.In the second phase of our study, we explored alternative approaches to dataset distillation, such as Difficulty-Aligned Trajectory Matching (DATM) and Distribution Matching (DM). We applied the same testing protocols to these methods to assess their advantages and disadvantages in comparison to the gradient matching technique. [1]

II. Task 1

1. Basic Concepts

a. Dataset distillation (DD) alleviates the storage and preprocessing tedium associated with large-scale datasets by learning a small set of information-rich images from a large amount of training data. Meanwhile, Zhao's team [1] mentioned that there are two advantages of using DD: i) It does not rely on heuristics, and can embrace any optimal solution for the task downstream. ii) It does not rely on representative samples. iii) It does not depend on the number of samples. iv) It does not depend on the number of samples. ii) It does not rely on representative samples. In [1], they used DD to obtain synthetic information samples and used the new samples to train neural networks for downstream tasks. The goal of in paper is to obtain the highest generalization performance with a model trained on a small set of synthetic images.

b. In this article, a new dataset compression method is proposed for learning small sets of ”compressed” synthetic samples. Training a deep neural network with synthetic samples allows the model to achieve performance similar to that of a model trained on a large ensemble. synthesizes a small set of highly informative samples, condensing large datasets into a more manageable size. This efficiency is crucial in scenarios where storing and processing large datasets is expensive and impractical. By using these condensed datasets, the methodology ensures that deep neural networks can be trained effectively without the need for extensive data. In addition, unlike traditional dataset reduction techniques that may select a subset of the original or approximate data distribution, Gradient Matching Approach synthesizes samples that mimic the gradient information of the original dataset. This approach ensures that the synthetic dataset is not just a smaller version of the original but is optimized for effective training, leading to potentially better model performance.The paper [1] also mentioned that traditional data selection methods often rely on heuristics and assume the presence of representative samples in the dataset. The proposed method overcomes these limitations by directly optimizing the synthetic data for the specific downstream task, rather than relying on potentially unrepresentative samples, the gradient matching technique circumvents the inefficient expansion of the recursive computation graph over prior parameters, significantly enhancing the speed of the optimization process.

c. The primary novelty of Zhao's team [**?**] lies in its efficient approach to creating condensed synthetic samples. In

[1] GitHub Link: [online] https://github.com/zbc0917/ECE1512-2023F-ProjectRepo-Lanzhi-Xiao-Bocheng-Zhang/tree/main/Project%20B

other words, they create a new method to do the dataset condensation. This method is based on gradient matching, where the gradients of neural network weights trained on the original and synthetic data are matched. By comparing the left (prior methods) and right (new method) Dataset Condensation methods, (a) and (b), the method on the right unlike Knowledge Distillation (KD) approach, the solution's proximity in the gradient space is closer and more consistent, thereby facilitating easier convergence of the network to a local minimum. Furthermore, optimization steps involving synthetic data are excessively resource-intensive for large-scale models with numerous parameters. This is due to the need for back-optimization on each parameter, which necessitates expanding the recursive computation graph over previous ones – a process that can be efficiently bypassed in gradient space. Therefore, our method, which does not generate synthetic data at the scale of real images, is more efficient.

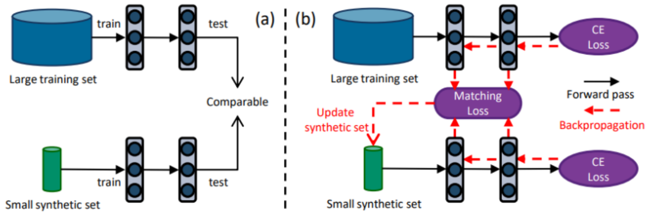


Fig. 1. Dataset Condensation introduction. Left figure shows the baseline method and right figure shows the Dataset Condensation with gradient match method [1].

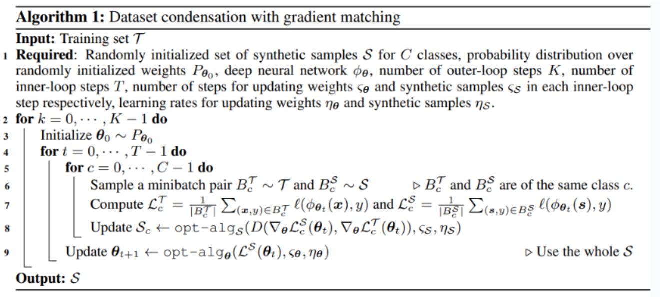
d. In the paper [1], the main idea is using a curriculum based condensation method to condense large datasets into a small set of informative synthetic samples that can effectively train deep neural networks from scratch. The curriculum gradient matching algorithm can be formulated as eq.1:

$$\min_{s} \mathbb{E}_{\theta_0 \sim P_{\theta_0}} \left[\sum_{t=0}^{T-1} D\left(\nabla_\theta L^S(\theta_t), \nabla_\theta L^\tau(\theta_t)\right) \right] \tag{1}$$

where P_{θ_0} is the distribution of randomly initialized model parameters, τ is the number of iterations, $\mathcal{L}^S$ and $\mathcal{L}^\tau$ represent the loss function for synthetic dataset and real train dataset. The gradients for the loss over the training samples $\mathcal{L}^\tau$ and the gradients for the loss over the condensed samples $\mathcal{L}^S$ are separated by a function termed D. Meanwhile, the D function (Gradient matching loss) in eq.2. Where $\mathcal{A}_i$, $\mathcal{B}_i$, are flattened vectors of gradients corresponding to each output node i. Specifically, the objective is to match the gradients for the synthetic and real training losses by updating the condensed samples.

$$d(\mathbf{A}, \mathbf{B}) = \sum_{i=1}^{\text{out}} \left(1 - \frac{\mathbf{A}_i \cdot \mathbf{B}_i}{\|\mathbf{A}_i\| \|\mathbf{B}_i\|}\right) \tag{2}$$

The detailed algorithm will be shown in the figure 2. In each outer loop k, we would randomly initialize a new set of model parameters , then in the inner loop, we compute the cross-entropy loss over both the real and synthetic samples for each class, then we update the parameters based on the distance of these two results. The next step is to train the model on the updated synthetic data. For calculating the distance between two cross-entropy losses, the paper [1] uses a sum of layer wise losses as eq(2), where A and B are flattened vectors gradients for the compared node in both samples.



```
Algorithm 1: Dataset condensation with gradient matching
Input: Training set $\mathcal{T}$
Required: Randomly initialized set of synthetic samples $\mathcal{S}$ for $C$ classes, probability distribution over randomly initialized weights $P_{\theta_0}$, deep neural network $\phi_\theta$, number of outer-loop steps $K$, number of inner-loop steps $T$, number of steps for updating weights $\varsigma_\theta$ and synthetic samples $\varsigma_S$ in each inner-loop step respectively, learning rates for updating weights $\eta_\theta$ and synthetic samples $\eta_S$.
for $k = 0, \cdots, K-1$ do
    Initialize $\boldsymbol{\theta}_0 \sim P_{\boldsymbol{\theta}_0}$
    for $t = 0, \cdots, T-1$ do
        for $c = 0, \cdots, C-1$ do
            Sample a minibatch pair $B_c^{\mathcal{T}} \sim \mathcal{T}$ and $B_c^{\mathcal{S}} \sim \mathcal{S}$    ▷ $B_c^{\mathcal{T}}$ and $B_c^{\mathcal{S}}$ are of the same class $c$.
            Compute $\mathcal{L}_c^{\mathcal{T}} = \frac{1}{|B_c^{\mathcal{T}}|}\sum_{(\boldsymbol{x},y)\in B_c^{\mathcal{T}}} \ell(\phi_{\boldsymbol{\theta}_t}(\boldsymbol{x}), y)$ and $\mathcal{L}_c^{\mathcal{S}} = \frac{1}{|B_c^{\mathcal{S}}|}\sum_{(\boldsymbol{s},y)\in B_c^{\mathcal{S}}} \ell(\phi_{\boldsymbol{\theta}_t}(\boldsymbol{s}), y)$
            Update $\mathcal{S}_c \leftarrow \texttt{opt-alg}_{\mathcal{S}}(D(\nabla_{\boldsymbol{\theta}}\mathcal{L}_c^{\mathcal{S}}(\boldsymbol{\theta}_t), \nabla_{\boldsymbol{\theta}}\mathcal{L}_c^{\mathcal{T}}(\boldsymbol{\theta}_t)), \varsigma_{\mathcal{S}}, \eta_{\mathcal{S}})$
        Update $\boldsymbol{\theta}_{t+1} \leftarrow \texttt{opt-alg}_{\boldsymbol{\theta}}(\mathcal{L}^{\mathcal{S}}(\boldsymbol{\theta}_t), \varsigma_{\boldsymbol{\theta}}, \eta_{\boldsymbol{\theta}})$    ▷ Use the whole $\mathcal{S}$
Output: $\mathcal{S}$
```

Fig. 2. Algorithm about dataset condensation with gradient matching [1]

e. Two potential applications for the dataset condensation with gradient matching methodology mentioned in the paper [1], continue learning and neural architecture search. Firstly, start work to apply this method on continual-learning scenario, new tasks are learned incrementally, is build model on E2E method. Since the model created on E2E method, the model uses a limited budget rehearsal memory to keep representative samples from the old tasks and knowledge distillation (KD) to regularize the network's output to consideration its previous predictions. Meanwhile, the sample selection mechanism will be replaces to a set of condensed images. In the evaluating part, the ”SVHN”, ”MNIST”,and ”USPS” dataset will be used to evaluate this model on the task-incremental learning problem. The findings demonstrate that the condensed images exhibit a higher level of data efficiency compared to those obtained through herding.
Secondly, this method has been applied in Neural Architecture Search base on ”CIFAR10” dataset [1]. The goal of this application is to define the best network, and verify the condensed images can be used in different model. In order to achieve the goal, there are 720 ConvNet network created with different parameter (W, N, A, P, D). The networks have been trained for 100 epoch on 3 small proxy datasets (ipc = 10). As the result, the method in the paper [1] achieves the highest testing performance. Meanwhile, the synthetic dataset training significantly decreases the searching time and storage space compared to original dataset training. In conclusion, the synthetic dataset can be used to train with different network in faster and less storage space way.
Additionally, there are two applications for the dataset

condensation with gradient matching methodology come to mind. Firstly, in the realm of medical image analysis, where the enormity of medical datasets poses a challenge, especially given the privacy constraints in sharing such data between hospitals. By employing dataset condensation, one can transform a voluminous medical image dataset into a more compact, synthetic version that retains key features crucial for precise model training. Secondly, this methodology has promising use in natural language processing (NLP), particularly for language translation models. Typically, developing cutting-edge translation models demands a large, bilingual or multilingual text corpus, which can be a resource-intensive endeavor. With dataset condensation, it's feasible to generate a smaller, synthetic dataset that effectively embodies the vital linguistic elements, structures, and vocabularies found in the extensive original corpus.

2. Dataset Distillation Learning

MNIST Implement

(a) In relation to the MNIST dataset, we engage with images measuring 28 by 28 pixels. The test set comprises 10,000 authentic validation images. Utilizing the Python package fvcore, we have ascertained that the computational complexity, measured in floating-point operations (FLOPs), is approximately 492,462,080,000. Subsequent to employing the original dataset for training the chosen model, we achieved a training accuracy of 99.98% and a testing accuracy of 99.47%.

(b) We adhere to the parameter specifications outlined in the accompanying table and employ the GitHub code from the referenced paper [**?**]. This process involves initializing our synthetic images using the actual dataset, followed by commencing the iterative process. For each iteration in the outer loop, we compute the loss for both the synthetic and real datasets. This loss is then inputted into the match loss function, which serves as the basis for updating our gradients. In the inner loop, we execute the model training, taking into account every parameter update. The default configuration for this segment is as follows:

- ipc = 10
- optimizor: SGD
- Learning Rate for image and net: 0.1 and 0.01
- Iteration: 100
- minibatch size: 256

(c) The image (Figure 3 and Figure 4) shown the condensed images per class for MNIST dataset:
The condensed images are recognizable. When we compare the initial and final states, we can still identify the synthetic images that were trained using an initialization from the real dataset. However, there are some differences in each number, including the presence of shadows close to the original number. These shadows seem to reflect hidden features within the data.

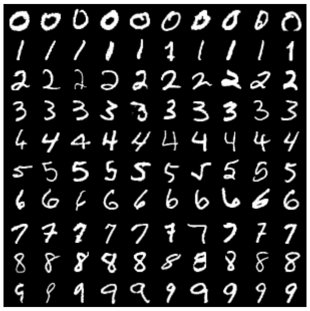


Fig. 3. Real Dataset

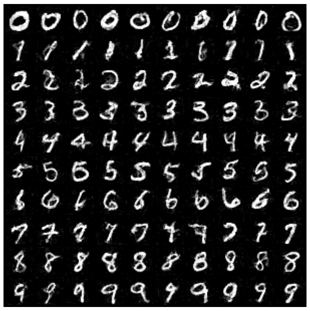


Fig. 4. Synthetic Dataset

(d) When we initialize the synthetic dataset from random noise, we have the table:

Dataset	Test Accuracy
Original	0.9952
Real Synthetic	0.961
Noise Synthetic	0.9488

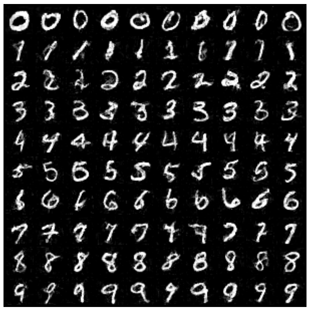


Fig. 5. Initialize from real

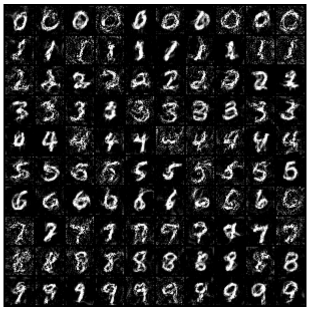


Fig. 6. initialize from noise

In quantitative terms, when comparing outcomes with various initialization, the results stemming from Gaussian noise initialization seem notably less recognizable. Although the original images are still discernible, they exhibit an increased level of noise, with numerous parts of the images com we haposed of scattered points. This is due to the fact that training from white noise necessitates more iterations to capture the same underlying information as training from the actual dataset. If we were to employ the same number of iterations, the images would inevitably contain less meaningful information. In qualitative terms, it is evident that the accuracy achieved with real data initialization surpasses that obtained through initialization with Gaussian noises.

(e) For testing the condensed images on the real dataset, we have the following result:

Dataset	Iteration	Time	Test Accuracy
Synthetic Dataset with noise initialization	20	4s	0.9546
Synthetic Dataset with real data initialization	20	6s	0.9608
Real Dataset	20	420s	0.9994

In terms of test accuracy, both models were assessed 20 times, and the model trained on the real dataset outperformed the other one, albeit by a slight margin. The model trained on the synthetic dataset exhibited an accuracy that was only 4% lower. However, it's worth noting that training the model on real data required 7 minutes for the MNIST dataset, whereas training on the MNIST synthetic dataset was completed in less than 10 seconds. The model, which underwent training utilizing compressed imagery and commenced with Gaussian noise initialization, demonstrates notable efficiency, evidenced by its rapid training time and commendable performance.

MHIST Implement

(a) We start with trained the "ConvNet" network with the original dataset and obtained the test accuracy and floating-point operations (FLOPs) for the test set. At the start of this part of work, we load the MHIST data and pre-processed the images, like reshape, normalization. In order to reduce the consume of RAM, the image will be reshaped to 32*32. The "ConvNet" net will be trained with learning rate = 0.01 in the first of 100 epoch and the learning rate in the rest 100 epoch will be reduce to 0.001 (0.01*0.1). Meanwhile, the SGD will be used as an optimizer.The result shows below:

- **Flops**: 50771968.0
- Train Loss: 0.0007   Training accuracy: 1.00
- Test Loss: 0.8812   Testing accuracy: 0.7881

(b) In order to train and evaluate a neural network model, specifically focusing on data condensation and synthetic data generation, function DataD_GradientM has been created. In the function, we first organize the dataset and initialize the synthetic dataset basic real dataset. Also, we use Gradient Matching algorithm to create a synthetic dataset and save in a list. The default setting in this part will be following:

- ipc = 50: there are 2 classes ('HP', "SSA"), for each class with 50 pictures.
- optimizor: SGD
- Learning Rate for image and net: 0.1 and 0.01
- Iteration: 200
- minibatch size: 128

(c) The image below shows the original image and synthetic image.There are two rows and they represent the class "SSA" and "HP". By comparing the synthetic dataset and real dataset, there are some shadows in the synthetic dataset. The original image is more clear to see and identify, but the synthetic image contains more information for network training and learning. In additionally, in order to overcome the limitation of RAM, we resize the image to 32*32. As the result, the final test accuracy is 0.7206 (72.06%) and training loss is 0.00614. The synthetic dataset will show in figure 3

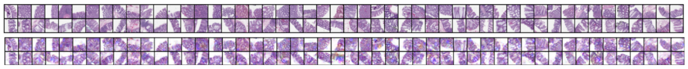


Fig. 7. Initially Real Synthetic Dataset (Top) and Updated Synthetic Dataset (Bottom)

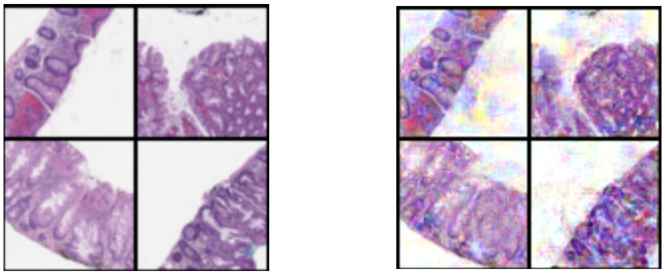


Fig. 8. Real Synthetic Dataset in Bigger size. Initial Real Synthetic(left); Updated Real Synthetic (right)

(d) Also, we initialize the synthetic dataset basic to the random Gaussian Noise. The figure below represents the synthetic dataset of the Gaussian Noise before iteration and after iteration. Once the noise synthetic dataset is created, the image is full of noise and contains less information. As the number of iterations increases, the synthetic dataset gradually learns the information in the real data, and the picture becomes clearer. Picture 2 shows the results of the synthetic dataset after 200 iterations. Although the picture has become clearer, there is still a lot of noise. As the result, the test accuracy is 0.6592 (65.92%), which is lower than the synthetic dataset initialize with real data.Qualitatively, images condensed using Gaussian noise initialization appear more obscure compared to their predecessors, suggesting a potential reduction in information content.

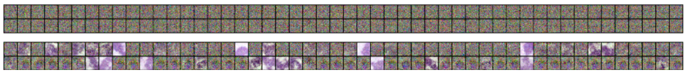


Fig. 9. Initially Noise Synthetic Dataset (Top) and Updated Synthetic Dataset (Bottom)

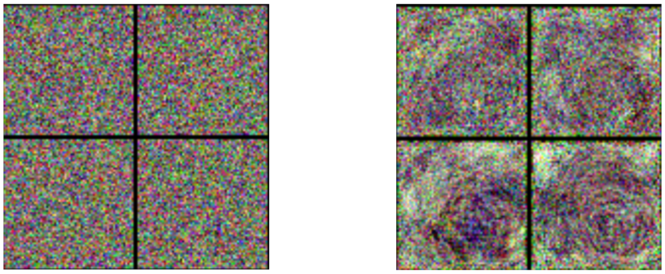


Fig. 10. Noisy Synthetic Dataset in Bigger size. Initial Noise Synthetic(left); Updated Noise Synthetic (right)

(e) By comparing the test accuracy of 3 training dataset, original dataset, real synthetic dataset and noise synthetic dataset, the noise synthetic contributes the lowest test accuracy with the same iteration number, 200. It is because there are a lot of noise points in noise synthetic images, it will affect the training of the model. The data shows in Table III.

Dataset	Test Accuracy
Original	0.7881
Real Synthetic	0.7206
Noise Synthetic	0.6592

TABLE I
MHIST ACCURACY COMPARISON

The data presented in Table IV offers an insightful comparison between the training outcomes of the ”ConvNet” network using both synthetic and real datasets. When evaluating the time efficiency, it is evident that training with a synthetic dataset is more time-effective compared to using a real dataset. Specifically, for the real dataset, the ”ConvNet” network requires a total of 62 seconds to complete 200 iterations of model training with an image size of 32 x 32 pixels. In contrast, the same network, when trained on a synthetic dataset, accomplishes the training process in just 2.96 seconds for the same number of iterations and image size. This notable difference in training duration can be attributed to the smaller volume of images in the synthetic dataset as opposed to the real dataset, naturally resulting in extended training times for the latter.

Moreover, a comparative analysis of the test accuracy reveals that the model trained with the real dataset achieves a higher accuracy of 0.7881, surpassing the 0.7206 accuracy level of the model trained with the synthetic dataset. This discrepancy underscores an important consideration: although utilizing a synthetic dataset for training markedly reduces the time requirement, there remains a discernible disparity in the performance effectiveness when compared to models trained with real datasets. Consequently, while synthetic datasets offer a time-efficient training solution, they do not entirely match the efficacy achieved through real dataset training, indicating an area for potential improvement in synthetic data-based model training methodologies.

Dataset	Iteration	Time	Test Accuracy
Synthetic Dataset	200	2.960s	0.7226
Real Dataset	200	900s	0.7881

TABLE II
MHIST PERFORMANCE COMPARISON OF DATASETS

3. Cross-architecture Generalization

MNIST Implement

The synthetic dataset, despite its apparent noisiness from a human visual perspective, captures essential latent features that are integral to the dataset. These features play a critical role in classification tasks, offering valuable insights that are not immediately discernible. The utilization of gradient matching techniques in conjunction with synthetic data facilitates the incorporation of these highly informative elements. This methodological approach significantly mitigates the losses outlined in the research paper, thereby augmenting the overall efficacy of the models.

In terms of model selection for our experimental framework, we have opted for AlexNet and VGG11, representing a cross-architecture approach. This selection was informed by their robustness and adaptability across different datasets and training conditions. Each model underwent a training period of approximately 6 seconds, a duration that is remarkably efficient considering the complexity of the tasks involved. Notably, the test accuracy achieved post-training was commendably high, underscoring the effectiveness of our training methodology.

Among the models evaluated, ConvNet emerged as the most outstanding in terms of performance, although even the least effective model in our study achieved a test accuracy of around 0.733. This is a testament to the potency of images processed through gradient matching training, which proved to be advantageous across a variety of models. In conclusion, our findings robustly support the assertion that leveraging images condensed via gradient matching training techniques offers substantial benefits in enhancing model performance across diverse architectural frameworks.

Model	Time	Test Accuracy
AlexNet	6s	0.73
ConvNet	6s	0.9603
VGG11	6s	0.9510

MHIST Implement

The research involves the utilization of a synthetic dataset for the training and testing of three distinct models: AlexNet, ConvNet, and VGG11. The ’Time’ column in our data representation is designed to quantify the duration each model requires to execute a single inference cycle or to process a given batch of data. In this context, ConvNet demonstrates the highest efficiency, completing the task in a mere 2 seconds. This is followed by Alex-Net, which requires 5 seconds, and VGG11, which is relatively slower, taking 6 seconds for the same process.

When we draw a comparison between the time efficiencies of the synthetic and real datasets, it becomes evident that the synthetic dataset significantly reduces the time required for both training and testing phases. This efficiency gain, however, comes with a trade-off in terms of test accuracy. The models trained with the synthetic dataset exhibit lower test accuracy when compared to their counterparts trained with a real dataset. This disparity in performance metrics is a crucial aspect to consider in the evaluation of synthetic dataset utility.

Despite these variances in accuracy, it is noteworthy that the gradient matching method demonstrates a versatile adaptability across different architectural frameworks. This method proves particularly beneficial in conserving training time, even in scenarios where the models have not been previously exposed to a real dataset before testing. This versatility and time efficiency highlight the potential of gradient matching as a viable approach in machine learning, especially in situations where rapid model training is prioritized and where access to extensive real datasets may be limited. Overall, the application of the gradient matching method across diverse architectures not only streamlines the training process but also opens up avenues for further exploration in the field of synthetic data utilization in machine learning.

Model	Time	Test Accuracy
AlexNet	5s	0.6377
ConvNet	2s	0.7206
VGG11	6s	0.6960

4. Application

Building upon the foundational research delineated in [4], the FedD3 framework marks a significant advancement in federated learning by integrating the concept of dataset distillation. This innovative approach facilitates efficient federated learning processes by enabling the transmission of locally distilled datasets to a central server in a singular, consolidated manner. Such a methodology ensures that highly informative training data is relayed across constrained bandwidths, all while maintaining strict adherence to privacy standards.

Contrasting with conventional federated learning methods that predominantly focus on the exchange of model updates, FedD3 introduces a unique paradigm where connected clients independently distill their local datasets. Following this, the framework proceeds to aggregate these autonomously distilled datasets—typically comprising a select number of unrecognizable images—from various networks for the purpose of model training. This strategic approach of handling a compact cluster of condensed images through distributed systems is noteworthy. It not only maintains a parity in accuracy levels when compared to traditional methods but also significantly reduces the processing time required for such operations.

Our implementation of this framework, influenced and guided by the code from [5], demonstrates these principles in action. The ensuing results provide empirical evidence of the efficacy and efficiency of the FedD3 framework in optimizing federated learning processes, highlighting its potential to revolutionize the field by balancing the dual objectives of processing efficiency and data privacy.

Method	**Dataset**	**Time**	**Accuracy**
FedD3	MNIST	3 min for 500 epochs	0.7385

III. Task 2

1. Read paper and answer the question

Paper 48: Dataset condensation with distribution matching [6]

a. The innovative methodology termed ”Dataset Condensation with Distribution Matching” represents a significant leap forward in addressing a crucial knowledge gap. It pioneers the generation of informative samples, shifting away from the conventional reliance on pre-existing data samples. This technique emerges as a highly effective strategy for surmounting the limitations posed by the information bottleneck in data processing.

   Furthermore, the authors adeptly address the challenge of computational burden, a key concern in the field. Traditional approaches necessitate updating network weights through numerous steps within each outer iteration, coupled with the complex task of unrolling recursive computation graphs. In contrast, this new approach ingeniously employs the matching of feature distributions between synthetic and original training images. Such a methodological shift offers a substantial advantage by obviating the need for the intensive and resource-heavy process of unrolling computational graphs, thereby significantly diminishing the synthesis cost.

   In comparison to previous methodologies that required resolving bi-level optimization challenges or intricate tuning of hyperparameters across both outer and inner loop optimizations, this novel approach simplifies the process. It utilizes networks with random initialization to embed both real and synthetic data, following which the maximum mean discrepancy is employed to calculate loss. This not only streamlines the computational process but also enhances the efficiency and efficacy of data synthesis and processing. By introducing these methodological advancements, ”Dataset Condensation with Distribution Matching” sets a new benchmark in the field, offering a more streamlined, cost-effective, and technically efficient solution to the challenges of data processing and optimization.

b. In the past, two predominant methodologies have been employed in this domain. The first approach required an extensive and resource-intensive unrolling of computational graphs, which often resulted in substantial computational overhead. The second approach was characterized by the necessity for multiple updates of network weights during each outer iteration, similarly incurring significant computational costs. These traditional methods, while effective to a certain degree, presented considerable challenges in terms of efficiency and resource allocation.

   However, in the current paper, the author has introduced an innovative methodology that marks a departure from these conventional approaches. This novel method hinges on the alignment of feature distributions between synthetic and original training images. Central to this process is the application of the Maximum Mean Discrepancy (MMD) as the loss function, which is adeptly employed to harmonize the disparities in the embeddings of the datasets.

   This approach stands out for its increased efficiency when compared to the aforementioned traditional methods. By optimizing the process of aligning synthetic and real training data, the new method significantly reduces the computational burden typically associated with such tasks. Furthermore, it does not compromise on performance, delivering results that are comparable, if not superior, to those achieved by the previous methods. This groundbreaking development not only enhances the efficiency of computational processes but also opens new avenues for further research and innovation in the field, potentially leading to even more sophisticated and streamlined methodologies in the future.

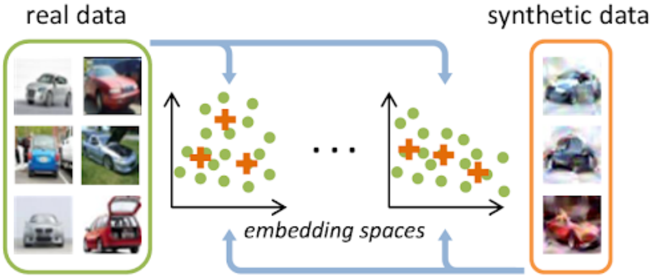


Fig. 11. The general methodology of distribution matching

c. In the research paper [6], the authors engage in the process of training synthetic data by minimizing the discrepancy that exists between two distributions across multiple embedding spaces. Consequently, they formulate the ultimate optimization functions that need to be addressed as follows:

$$\begin{aligned}\min_{S} \mathbb{E}_{\theta\sim P_\theta}\mathbb{E}_{\omega\sim\Omega}\Bigg\| & \frac{1}{|T|}\sum_{i=1}^{|T|}\psi_\theta(A(x_i,\omega)) \\ & -\frac{1}{|S|}\sum_{j=1}^{|S|}\psi_\theta(A(s_j,\omega))\Bigg\|^2\end{aligned} \tag{3}$$

In contrast to the challenge posed by the matching gradient problem, the approach described here employs a simplified optimization framework without the presence of both inner and outer loops. Instead, it relies solely on a single loop for updating the synthetic data. The training algorithm follows a straightforward procedure: pairs of real and synthetic data batches are sampled, and the mean discrepancy between the embedding data batches is computed. Subsequently, the synthetic data undergoes updates through stochastic gradient descent to minimize this discrepancy. Furthermore, for the embedding functions, a deep neural network is employed with varying random initializations to enhance diversity and exploration.

d. The advantages of the method is reducing the synthesis cost compared to existing approaches. And it can be applied to larger datasets, since the training can be run independently for each class, allowing for parallel processing, and these methods don't need to do the tuning hyperparameters. However, there are disadvantages for this method, as mentioned in the paper [6] there are still questions why randomly initialized networks provide meaningful embeddings for distribution matching. So, they don't have the mathematical proof, only the observation from previous work. And I think this method can handle large-scale datasets effectively. Since these methods don't require the hyperparameter tuning and have effectiveness across various architectures. For the computing resources, it can run independently for each class, so we can do the parallel processing to deal with large-scale datasets like ImageNet.

Paper 13: Towards Lossless Dataset Distillation via Difficulty-Aligned Trajectory Matching [2]

a. In ”Towards Lossless Dataset Distillation via Difficulty-Aligned Trajectory Matching ” [2] mentioned that there remains a significant gap between the performance of their distilled datasets (Synthetic Dataset) and the real counterparts (Real Dataset). The gap between the performance of the synthetic dataset and original dataset can be minimize by increasing the number of picture per class. However, increasing the number of pictures per class (IPC), the dataset distillation methods become less effective, even performing worse than random selection. In [2], they tried to implement the DATM Dataset Distillation method to minimize the loss in the dataset distillation.

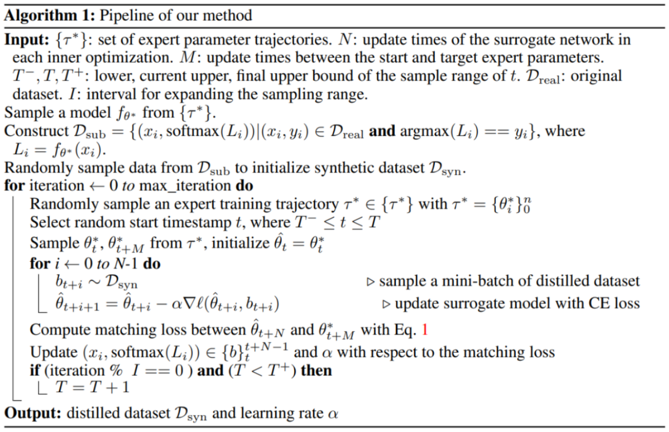
b. According to Ziyao's group [2], they tried to manage to align the difficulty of the learned partners with the size of the distilled dataset. As a result, the new method will be able to work well in both low and high IPC settings. By comparing the related method, trajectories matching (TM) bases distillation methods, vary widely depending on teacher training stage. In other words, matching early or late trajectories causes the synthetic data to learn easy or hard patterns respectively. The method in this paper [2], making the optimization stable enough for learning soft labels during the distillation by learning easy and hard patterns sequentially.

c. The pipeline of Dataset Distillation by Matching Training Trajectories will be shown in figure 3.1. In the paper [2], the author's group starts by experimenting with the influence of matching trajectories from different training stages. As a result,there are two observations: early trajectories are better with smaller synthetic dataset; matching late trajectories leads to poor performance in low IPC settings. Basically the finds in the starting step, the formulated bellowing will be used to calculate the trajectory segment.

$$T^* = \left\{ \theta_0^*, \theta_1^*, \dots, \underbrace{\theta_{T-}^*, \dots, \theta_{T+}^*}_{\text{matching range}}, \dots, \theta_n^* \right\}. \tag{4}$$

In addition, the network will be trained on the mini batch of synthetic images with a fixed number of steps, and the loss is computed based on the equation (5). Where θ_T^* and θ_{T+M}^* are the randomly sampled from a set of expert trajectories and $\{T^*\}$ is the start parameters and target parameters used for the matching. At the end, the batch of synthetic samples will be updated by the loss.

$$\mathcal{L} = \frac{\|\hat{\theta}_{t+N} - \theta_{t+M}^*\|_2^2}{\|\theta_t^* - \theta_{t+M}^*\|_2^2} \tag{5}$$



```
Algorithm 1: Pipeline of our method
Input: $\{\tau^*\}$: set of expert parameter trajectories. $N$: update times of the surrogate network in
  each inner optimization. $M$: update times between the start and target expert parameters.
  $T^-, T, T^+$: lower, current upper, final upper bound of the sample range of $t$. $\mathcal{D}_{\text{real}}$: original
  dataset. $I$: interval for expanding the sampling range.
Sample a model $f_{\theta^*}$ from $\{\tau^*\}$.
Construct $\mathcal{D}_{\text{sub}} = \{(x_i, \text{softmax}(L_i)) | (x_i, y_i) \in \mathcal{D}_{\text{real}}$ and $\text{argmax}(L_i) == y_i\}$, where
  $L_i = f_{\theta^*}(x_i)$.
Randomly sample data from $\mathcal{D}_{\text{sub}}$ to initialize synthetic dataset $\mathcal{D}_{\text{syn}}$.
for iteration ← 0 to max_iteration do
    Randomly sample an expert training trajectory $\tau^* \in \{\tau^*\}$ with $\tau^* = \{\theta_i^*\}_0^n$
    Select random start timestamp $t$, where $T^- \leq t \leq T$
    Sample $\theta_t^*, \theta_{t+M}^*$ from $\tau^*$, initialize $\hat{\theta}_t = \theta_t^*$
    for $i$ ← 0 to $N$-1 do
        $b_{t+i} \sim \mathcal{D}_{\text{syn}}$                                  ▷ sample a mini-batch of distilled dataset
        $\hat{\theta}_{t+i+1} = \hat{\theta}_{t+i} - \alpha \nabla \ell(\hat{\theta}_{t+i}, b_{t+i})$        ▷ update surrogate model with CE loss
    Compute matching loss between $\hat{\theta}_{t+N}$ and $\theta_{t+M}^*$ with Eq. 1
    Update $(x_i, \text{softmax}(L_i)) \in \{b\}_t^{t+N-1}$ and $\alpha$ with respect to the matching loss
    if (iteration % $I == 0$ ) and ($T < T^+$) then
        $T = T + 1$
Output: distilled dataset $\mathcal{D}_{\text{syn}}$ and learning rate $\alpha$
```

Fig. 12. Pipeline of GATM mentioned in the book Towards Lossless Dataset Distillation via Difficulty-Aligned Trajectory Matching [2]

d. Dataset Distillation (DD) methods, particularly the advanced Difficulty-Aligned Trajectory Matching (DATM) method, have unique advantages and disadvantages. There are some advantages for the DATM, flexibility and adaptability, enhanced learning capability. For the Flexibility and adaptability, DATM adjusts the complexity of the patterns it generates based on the size of the synthetic dataset [2]. This adaptability makes it effective across various IPC settings, ensuring that the distilled dataset remains representative of the original. For the enhanced learning capability, the DATM has the ability to ensure that the synthetic dataset covers a broad spectrum of the original dataset's patterns.In addition, DATM can decrease the loss between real dataset and synthetic dataset. However, there is a disadvantage of DATM, complexity in trajectory matching. The trajectory matching process can be intricate, requiring careful calibration to ensure an appropriate balance of easy and hard patterns. Meanwhile, DATM methods can analyze and inspect the cases of large-scale datasets. Large-scale datasets often contain a wide range of patterns, from simple to complex. DATM's approach to learning both easy and hard patterns makes it suitable for distilling such datasets, ensuring that the distilled version retains the diversity and complexity of the original. In paper [2], the authors compared with previous dataset distillation methods on Tiny ImageNet, as the result, DATM is better than other methods.

2. Implementation

Paper 48: Dataset condensation with distribution matching [6]

a. To apply the methodology described [6], our approach incorporates dataset condensation through distribution matching. This process encompasses a sequence of steps beginning with data acquisition and argument initialization. Contrary to previous frameworks that employed an inner training loop, our modified procedure eliminates this component. Instead, during each iteration, we systematically process each category of images by selecting representative samples from the authentic dataset and establishing a corresponding synthetic dataset. The pivotal phase of this method involves embedding; we concurrently embed both real and synthetic images. Following this, we compute the mean loss across the embeddings and utilize this metric to refine our model parameters. The evaluation phase adheres to conventional protocols. Regarding the outcomes:

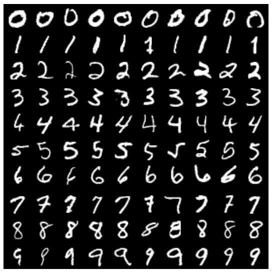


Fig. 13. original dataset

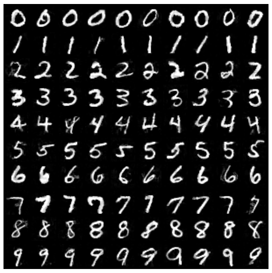


Fig. 14. condensed using DM

It is evident that distribution matching not only facilitates image condensation but also delivers commendable performance. While gradient matching offers a sufficiently brief training duration, distribution matching further reduces the time required for training. The observed test accuracies are as follows:

Method	Training Time	Test Accuracy
Gradient Matching	6s	0.961
Distribution Matching	2s	0.8679

b. To thoroughly assess the distinctions between gradient matching and distribution matching, we have experimented with various initialization and architectures. As indicated by the literature, distribution matching slightly compromises accuracy in exchange for significant computational efficiency. Synthesizing all the data at our disposal, our inference appears to be substantiated. From a qualitative standpoint, the images produced through distribution matching exhibit less noise compared to those from gradient matching, suggesting a potential reduction in condensed information. This could be attributed to the employment of mean loss matching, which may also contribute to reduced variation within the images. Overall, while distribution matching may not achieve the same level of performance as gradient matching, its advantage lies in reduced training time. Both methodologies demonstrate compatibility with diverse architectures. Regarding test accuracy, my assessment is that the methodologies in question do not surpass the gradient matching algorithm's performance. Discussing generalization and recognition capabilities entails ensuring that the methodology accurately identifies new instances and effectively generalizes to unfamiliar data, adaptable across various models. Distribution matching, while reducing test accuracy for the sake of minimizing training time, is particularly useful for training on extensive datasets. Its single-loop structure is especially beneficial for more complex image datasets, offering significant time savings. However, its potential drawback lies in capturing less information in the condensed images. This limitation may hinder a model's ability to develop a comprehensive understanding of the real dataset, impacting its effectiveness in recognizing and generalizing from the data.

Method	Initialization	Training Time	Test Accuracy	Model
GM	real	6s	0.96	ConvNet
GM	noise	6s	0.9488	ConvNet
GM	real	6s	0.9343	ConvNet
DM	real	2s	0.8679	VGG11
DM	noise	2s	0.8239	ConvNet
DM	real	3s	0.6617	VGG11

Paper 13: Towards lossless dataset distillation via difficulty-aligned trajectory matching [2]

a. According to the article "Towards Lossless Dataset Distillation via Difficulty- Aligned Trajectory Matching" [2], there is an lossless dataset distillation method will be implement, called **D**ifficulty- **A**ligned **T**rajectory **M**atch **(DATM)**. At the start of work, the buffer filers will be created, which generate expert trajectories. By comparing to task 1, the synthetic dataset will be initialized with "samples_predicted_correctly". The figure 13 shows the changing of synthetic dataset, initialized synthetic dataset (left), synthetic dataset after 200 iteration (right). The "samples_predicted_correctly" method is an introduced approach to initializing synthetic data in the paper. During the initialization process, there is a model (Temp_Net) will be used to evaluate a batch of real images. After the evaluating, only those images that the model correctly identifies are chosen to form the initial synthetic dataset. The Table VII shows the model result about DATM and GM method. By comparing the training time, the model use **G**radient **M**atching (GM) synthetic dataset takes fewer time than using DATM synthetic dataset. However, the model trained with DATM synthetic dataset have better performance than the model trained with GM mini-batch dataset.

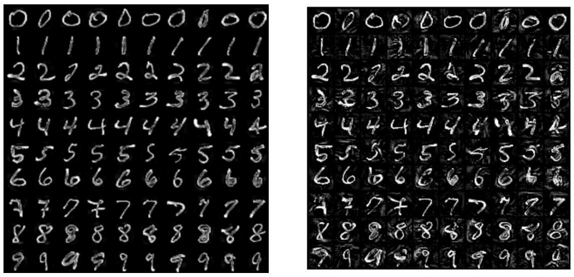


Fig. 15. The dataset will be initialize with "sample_predicated_correctly" method. The initialize synthetic dataset (left) shows the 10 class with ipc = 10. The synthetic dataset after DATM (right) shows image for 10 class with more shadows than left.

b. According to the paper [2], DATM is an consistently outperforms previous dataset distillation methods and is the method to achieve lossless distillation. In the task 2, we tried to implement DATM to distillate the MNIST dataset. For the synthetic dataset initialization part, DATM will be tested in both "real" and "samples_predicted_correctly" way. The Table VIII shows the result of DATM and GM in MNIST. In our experiment shows that the DATM can contributed an lossless way to do dataset distillation. For example, the performance of VGG11 is 0.9487, which trained with DATM synthetic dataset with real initialization way, higher than the VGG11 trained with GM synthetic dataset. By comparing the time cost, the model trained on DATM synthetic dataset a little bit slower than the GM synthetic dataset.

Method	Training Time	Test Accuracy
GM	6s	0.9601
DATM	7s	0.9692

TABLE III
DATM AND GM RESULT COMPARISON

Method	Initialization	Training Time	Test Accuracy	Model
baseline	original	420s	0.9994	ConvNet
GM	real	6s	0.9601	ConvNet
GM	real	6s	0.9459	VGG11
DATM	real	7s	0.9570	ConvNet
DATM	SPC	7s	0.9599	ConvNet
DATM	SPC	8s	0.9460	VGG11
DATM	real	9s	0.9487	VGG11

TABLE IV
DATM, BASELINE AND GM METHOD COMPARISON

In the paper [2], the DATM with 1000 ipc test accuracy surpasses that of a model trained on the original dataset with "CIFR-10". In addition, the selected method, DATM, shows superior generalizability compared to other methods. It performs best on unseen networks when IPC is small, reflecting the good generalizability of the data and labels distilled by this method. Thus, DATM provide an lossless way to do dataset distillation. For our test, the performance is not expected, there are some reasons. Firstly, there is not enough buffers created. Secondly, the iteration time is not enough. We tried to set the iteration to 200 in DATM method and the test accuracy, 0.9692, higher than the GM method.

REFERENCES

[1] B. Zhao, K. R. Mopuri, and H. Bilen, ”Dataset Condensation with Gradient Matching,” in ICLR, vol. 1, no. 2, p. 3, 2021. [Online]. Available: https://arxiv.org/abs/2006.05929

[2] Z. Guo, K. Wang, G. Cazenavette, H. Li, K. Zhang, and Y. You, ”Towards Lossless Dataset Distillation via Difficulty-Aligned Trajectory Matching,” arXiv preprint arXiv:2310.05773, 2023. [Online]. Available: https://arxiv.org/abs/2310.05773

[3] G. Cazenavette, T. Wang, A. Torralba, A. A. Efros, and J.-Y. Zhu, ”Dataset Distillation by Matching Training Trajectories,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4750-4759, 2022. [Online]. Available: https://arxiv.org/abs/2203.11932

[4] R. Song, D. Liu, D. Z. Chen, A. Festag, C. Trinitis, M. Schulz, and A. C. Knoll, ”Federated Learning via Decentralized Dataset Distillation in Resource-Constrained Edge Environments,” CoRR, vol. abs/2208.11311, 2022. [Online]. Available: http://arxiv.org/abs/2208.11311

[5] Rruisong, “FedD3,” 2023. [Online]. Available: https://github.com/rruisong/FedD3. Accessed: December. 2, 2023.

[6] B. Zhao and H. Bilen, ”Dataset Condensation with Distribution Matching,” arXiv preprint arXiv:2110.04181, 2021. [Online]. Available: https://arxiv.org/abs/2110.04181.