arXiv:2404.17732v1 [cs.CV] 26 Apr 2024

Generative Dataset Distillation: Balancing Global Structure and Local Details

Longzhen Li, Guang Li, Ren Togo, Keisuke Maeda, Takahiro Ogawa, and Miki Haseyama
Hokkaido University
{longzhen, guang, togo, maeda, ogawa, mhaseyama}@lmd.ist.hokudai.ac.jp

Abstract

In this paper, we propose a new dataset distillation method that considers balancing global structure and local details when distilling the information from a large dataset into a generative model. Dataset distillation has been proposed to reduce the size of the required dataset when training models. The conventional dataset distillation methods face the problem of long redeployment time and poor cross-architecture performance. Moreover, previous methods focused too much on the high-level semantic attributes between the synthetic dataset and the original dataset while ignoring the local features such as texture and shape. Based on the above understanding, we propose a new method for distilling the original image dataset into a generative model. Our method involves using a conditional generative adversarial network to generate the distilled dataset. Subsequently, we ensure balancing global structure and local details in the distillation process, continuously optimizing the generator for more information-dense dataset generation.

1. Introduction

The expansion of dataset sizes has notably propelled recent advancements in deep learning, especially within the field of computer vision [27]. However, the reliance on large datasets poses a challenge, as it often leads to considerable training expenses [1]. This issue can be addressed by two main approaches: data selection and dataset distillation. Data selection involves selecting a subset of representative data from the original large dataset [40]. Although this approach can reduce the training cost, it risks losing critical information. Dataset distillation, on the other hand, offers a more effective solution [36]. Rather than simply selecting existing data, it involves synthesizing a new and much smaller dataset that contains the important information of the original dataset. This approach can significantly reduce dataset size without substantially compromising performance. Moreover, dataset distillation offers a further advantage in terms of data privacy [19, 21].

Dataset distillation, an emerging area of interest within the research community, has evolved significantly in its algorithms and applications [14, 38]. Initially, dataset distillation creates a smaller dataset to mimic the training performance of the original dataset using meta-learning [36]. Subsequent advancements introduced gradient matching methods, focusing on aligning the gradients of models trained on both the original and distilled datasets [41]. It was further expanded with the introduction of distribution matching methods, which aim to adjust the smaller dataset's distribution to closely resemble that of the original dataset [43]. Recently, some dataset distillation methods based on matching the training trajectories have been proposed [3, 8]. The training trajectory refers to the change in the model weights during the training process. The more similar the training trajectories of the teacher and student models are, the closer the performance of the student model will be to that of the teacher model.

With the development of dataset distillation, the applications of dataset distillation have spaned various fields, including continual learning [37], privacy preserving [15, 16], and federated learning [29]. However, conventional dataset distillation methods often incur high redeployment costs because they rely on a fixed distillation ratio or the often-used image per class (IPC). Another challenge that the conventional dataset distillation method faces is the relatively poor cross-architecture performance. Distilled results on the small architecture will be hard to apply to a more complex architecture, which will lead to poor model generalization performance.

To solve the above issues, a new dataset distillation method is introduced, namely distilling the dataset into a generative model (DiM) [35]. Different from conventional methods, DiM distills the information of the whole dataset into a conditional generative adversarial network (GAN) model rather than images. This shift to model-based storage significantly enhances DiM's redeployment efficiency, as it eliminates the need for retraining when IPC or distillation ratios change, thus overcoming the limitation of conventional distillation methods. In the distillation process, DiM uses logit matching as the alignment strategy.

Logit matching focuses on the image category, emphasizing global information and high-level semantic attributes. Therefore, logit matching aligns the distilled image with the original image in terms of their category, rather than exact visual details [31]. However, it overlooks finer details such as shapes and textures, which limits the distillation accuracy and performance on cross-architecture generalization.

To address the issue of losing finer details, we propose a novel method that considers both global structure and local details. The motivation of our method is the integration of high-level semantic attributes with attention to local features that can improve the distillation process and hence generate more robust distilled datasets. The local features are extracted from the intermediate layers of the neural network, ensuring a more detailed representation of the data. Our method combines attention to both the broad, global aspects and the detailed, local features of images. Specifically, the proposed method introduces a novel loss function that simultaneously accounts for the final layer logits' discrepancies and the variance in local features contained in the intermediate network layers, ensuring a better distillation process. Therefore, our method offers a more comprehensive framework for dataset distillation, leading to more effective and accurate model training and better robustness. The effectiveness of our method has been verified through experiments on three benchmark datasets. Notably, our method's consideration of both global and local image aspects results in datasets that exhibit enhanced cross-architecture generalization capabilities, proving effective across various neural network types.

The contributions of this paper can be summarized as follows.

- We propose a new dataset distillation method that considers both global structure and local details, which can generate more robust distilled datasets.
- By distilling the information into a generative model instead of images, the proposed method significantly improved the redeployment efficiency, which prevented the high cost of re-optimization.
- We verified the effectiveness of the proposed method on three benchmark datasets, including MNIST, Fashion MNIST, and CIFAR-10. The proposed method is also verified as having a better performance in cross-architecture generalization.

2. Related Works

In this section, we present an overview of various dataset distillation methods. These methods are categorized into three classes: performance matching, gradient matching, and distribution matching. The choice of method depends on factors such as dataset size, deployment time, and computational cost.

2.1. Dataset Distillation Using Performance Matching

First, we introduce dataset distillation methods using performance matching. The goal is to optimize distilled datasets so neural networks trained on them mirror the loss profiles of networks trained on original datasets. This parity in performance ensures that models leverage the distilled datasets as effectively as the original ones. Within the methods, it is separated into subclasses like meta-learning methods, exemplified by gradient-based hyperparameter optimization, and kernel ridge regression methods. The inception of dataset distillation is first introduced by Wang et al. [36], and they employed meta-learning paradigms to optimize model weights as functions of distilled images. Enhancements to this method have since emerged, introducing variations with flexible labels [2], soft-label approaches [30], and the incorporation of parametrization [11] to improve distillation performance. Meta-learning-based approaches employ backpropagation to compute the gradient of the validation loss on synthetic datasets, a process that requires bi-level optimization and can be computationally demanding, especially as the number of inner loops grows, leading to increased GPU memory usage [22]. Limited inner loops can result in suboptimal optimization and performance issues, and scaling such methods to larger models presents further challenges [33]. However, kernel ridge regression methods like kernel inducing point (KIP) offer an alternative by enabling convex optimization, which yields a closed-form solution that obviates the need for exhaustive inner loop training [24]. Recent advancements in this domain have introduced methods that significantly enhance distillation efficiency and performance. These include leveraging infinitely wide convolutional networks [25] and employing neural feature regression [44], each contributing to the evolution of dataset distillation methods.

2.2. Dataset Distillation Using Gradient Matching

Next, we introduce dataset distillation methods using gradient matching. Zhao et al. first proposed a gradient matching-based method termed dataset condensation [41]. Different from performance matching, which tunes the efficacy of models using synthetic datasets, gradient matching refines the performance of networks trained on both the original and synthetic datasets by aligning their training gradients. Recent advancements have augmented gradient matching with strategies like differentiable data augmentation [42], enhancing training adaptability and efficacy. Furthermore, contrastive signaling [13] has been integrated to improve feature discrimination, and long-range trajectory matching [3] has been employed to align gradient trajectories over extended training cycles. Additionally, approaches such as parameter pruning and self-adaptive pa-

rameter matching [18, 20] have been investigated to optimize distillation complexity, potentially boosting the efficiency and outcome of the distillation process.

2.3. Dataset Distillation Using Distribution Matching

Finally, we illustrate some dataset distillation methods using distribution matching. Distribution matching aims to synthesize data whose distribution closely aligns with that of the original data within a specified embedding space. Zhao et al. first introduced the method of distribution matching, which utilizes the neural network's output embeddings, excluding the final linear layer [43]. The objective is to minimize the distance between the mean vectors (centers) of synthetic and original data for each class. Building on this, another method has been developed to align the attention across different layers of the network [26]. Although these distribution matching methods reduce synthesis costs and scale well with larger datasets, they require re-optimization with any change in the distillation ratio. This necessity for re-optimization can impact their efficiency in certain applications. For a deeper exploration of dataset distillation, the reader is directed to the latest survey papers [14, 38] or the Awesome Dataset Distillation project [17], which provide a thorough overview of the field.

3. Methodology

The proposed method aims to train a generator to synthesize information-condensed images. The proposed method includes three stages: conditional GAN training, dataset distillation via balancing global structure and local details, and the deployment stage.

3.1. Conditional GAN Training

Conventional GAN networks are used to generate visually realistic images. GAN networks generally include a generator G and a discriminator D [7]. The working principle of GAN can be summarized as follows:

$$L_{\text{GAN}} = \min_G \max_D V(D,G), \tag{1}$$

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))], \tag{2}$$

where $\boldsymbol{x}$ represents real data and $\boldsymbol{z}$ represents random noise. During the GAN training process, the generator G and discriminator D compete with each other. The generator G attempts to generate more realistic images and the discriminator D tries to distinguish between the generated and real data.

Different variants of the GAN network were later developed, and in the proposed method, we chose conditional GAN as the generative model to generate the distilled data. Compared with conventional GAN networks, conditional GAN introduces specific information into the input to generate images [23]. In the proposed method, we use labels as specific information. The training process of conditional GAN can be summarized as follows:

$$L_{\text{CGAN}} = \min_G \max_D V(D,G), \tag{3}$$

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z}|\boldsymbol{y})))], \tag{4}$$

where $\boldsymbol{y}$ represents the introduced labels.

The first step of the proposed method is to train a generator that can generate visually realistic images. The process for the trained generator in conditional GAN to generate discriminative images can be summarized as follows:

$$\mathcal{S} = G\left([\boldsymbol{z} \oplus \boldsymbol{y}]\,; \mathcal{W}\right), \tag{5}$$

where $\boldsymbol{z}$ is the random noise and $\boldsymbol{y}$ is the corresponding label. $\oplus$ denotes the concatenation operation. $\mathcal{S}$ represents the synthetic dataset and $\mathcal{W}$ is the parameter of generator G. The synthetic dataset $\mathcal{S}$ is initially generated by the generator G and subsequently optimized through the dataset distillation process, which distills the information from the original dataset $\mathcal{T}$. The optimizing method introduced in the next section will update G to achieve better performance to generate more efficient images that contain more valuable information.

The most important difference between conventional dataset distillation methods and the proposed method is that the former ultimately saves the distilled images, which means distilling the information into images, whereas the goal of the proposed method is to save the trained generator, which means distilling the information into a generative model. The trained generator can be used to generate any number of distilled images, which saves a lot of redeployment costs.

3.2. Dataset Distillation via Balancing Global Structure and Local Details

The obtained conditional GAN generator can enable the synthesis of visually convincing images. However, the synthetic dataset $\mathcal{S}$ often lacks the compressed information in the original dataset $\mathcal{T}$, hindering the performance of downstream tasks. Our method tackles this limitation by focusing on the optimization of the generator, aiming to enhance its capability to produce distilled data that goes beyond simple

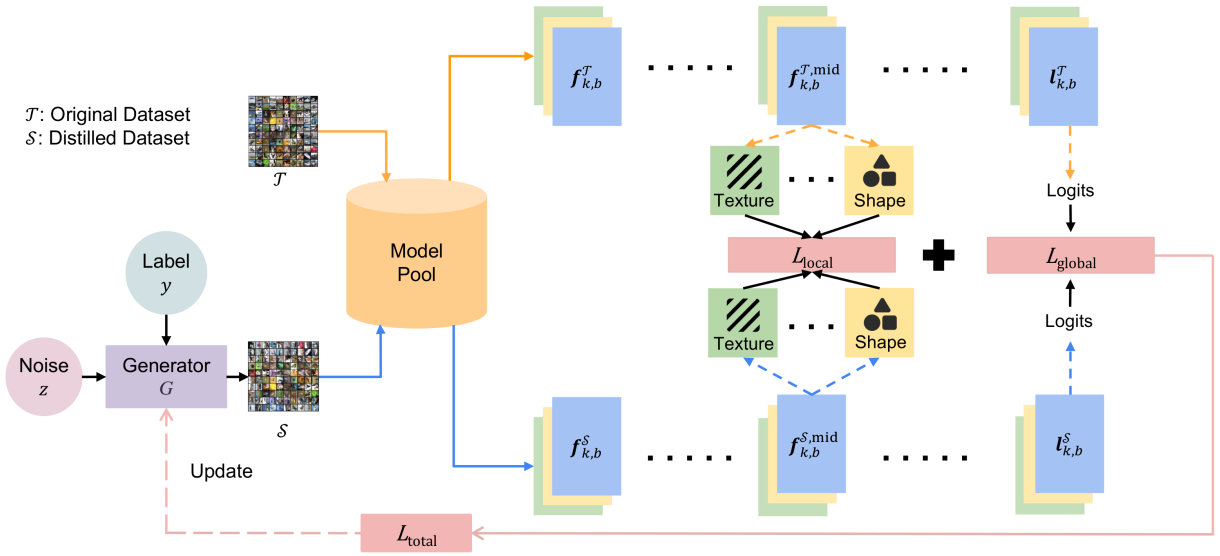


Figure 1. Overview of the distillation process. The goal is to train a generator that synthesizes images rich in information (referred to as distilled images), taking into account both global structure and local details.

visual authenticity, capturing a more potent and discerning representation of the important information.

As shown in Fig. 1, we use a random initial model from the model pool to match the global structure and local details between the original dataset $\mathcal{T}$ and the synthetic dataset $\mathcal{S}$. Then it continuously optimizes the generator by minimizing the loss between the synthetic dataset and the original dataset, thereby generating data that is more efficient for downstream classification tasks. Unlike conventional dataset distillation methods, we use a model pool that contains multiple convolutional neural networks and randomly select one from them to perform the matching between the original dataset $\mathcal{T}$ and the synthetic dataset $\mathcal{S}$. With the model pool, we can improve the robustness and generalization performance of dataset distillation. By using different models to perform matching steps, the features of the original dataset $\mathcal{T}$ are more fully utilized. The use of a model pool makes the proposed method have better cross-architecture stability, avoiding overfitting to specific architectures and making our method more robust.

In our method, the matching of synthetic dataset and original dataset can be divided into two parts, the matching of global structure and the matching of local details. The matching of global structure aims to analyze whether the synthetic dataset is consistent with the original dataset in terms of high-level semantic information, such as categories. The global loss is obtained by comparing the high-level semantic information of the original dataset and the synthetic dataset. When comparing the global information, we use logical matching to compare whether the synthetic dataset has achieved similar logits to the original dataset. This step uses the output from the last layer of a randomly selected model. The global loss can be defined as follows:

$$L_{\text{global}} = \sum_{k=1}^{K}\sum_{b=1}^{B}(\mathbf{l}^{\mathcal{S}}_{k,b} - \mathbf{l}^{\mathcal{T}}_{k,b})^2, \tag{6}$$

where B and K denote the batch size and the number of categories, respectively. $\mathbf{l}^{\mathcal{S}}$ and $\mathbf{l}^{\mathcal{T}}$ represent the output logits of the synthetic dataset and original dataset, respectively.

However, focusing only on global information will cause the loss of valuable detailed information in the data. Therefore, we further perform matching on local features such as texture and shape. In this way, the synthetic dataset $\mathcal{S}$ contains more valuable detailed information. When calculating the local loss, we propose using feature matching and selecting information from intermediate layers to compare the matching degree between the original data and the synthetic dataset in terms of texture, shape, and other detail aspects. The local loss can be calculated as follows:

$$L_{\text{local}} = \sum_{k=1}^{K}\sum_{b=1}^{B}(\mathbf{f}^{\mathcal{S},\text{mid}}_{k,b} - \mathbf{f}^{\mathcal{T},\text{mid}}_{k,b})^2, \tag{7}$$

where mid denotes an intermediate layer of the randomly selected network. $\mathbf{f}^{\mathcal{S}}$ and $\mathbf{f}^{\mathcal{T}}$ represent the output features of the synthetic dataset and original dataset, respectively.

The total loss function of the proposed method is a combination of global loss L_{global}, local loss L_{local}, and conditional GAN loss L_{CGAN}. We also defined ω_g and ω_l to represent the weights of global loss and local loss. The calculation of total loss L_{total} can be summarized as follows:

$$L_{\text{total}} = \omega_g L_{\text{global}} + \omega_l L_{\text{local}} + L_{\text{CGAN}}. \tag{8}$$

During the dataset distillation stage, the optimization process focuses on minimizing the total loss function. Through this minimization process, the generator progressively improves its ability to generate data similar to the desired target distribution. This minimization process can be summarized as follows:

$$\mathcal{W}^* = \arg\min_{\mathcal{W}} L_{\text{total}}, \tag{9}$$

where $\mathcal{W}^*$ is the optimized parameters of the generator G. The proposed method ensures balancing global structure and local details of the original dataset and synthetic dataset during the dataset distillation process as much as possible by matching them and making the synthetic dataset contain as much detailed information as possible, thereby generating distilled datasets for downstream tasks.

3.3. Deployment Stage

After the above optimization, the generator cannot only generate visually realistic images but also generate distilled images. These distilled images contain more key information that is helpful for downstream tasks such as recognition and classification. Therefore, during the deployment phase, we provide various random noises $\mathbf{z}$ and corresponding labels $\mathbf{y}$ to the generator and use the generator to dynamically generate various distilled dataset $\mathcal{S}^*$ as follows:

$$\mathcal{S}^* = G\left([\mathbf{z} \oplus \mathbf{y}]; \mathcal{W}^*\right). \tag{10}$$

This distilled dataset can be used to serve as the alternative for the original dataset to effectively reduce the volume of the dataset. Moreover, since we saved the trained generator, the information of the whole dataset was distilled into the generative model during this process, rather than static images. Therefore, when we apply the new proposed method to other architectures or change the distillation ratio, there is no need to retrain the model. This improves the efficiency of redeployment a lot. Algorithm 1 summarizes the proposed method. The generative dataset distillation method trains a generator G of conditional GAN first. Then the global-local coherence of the original and synthetic was matched. Finally, the generator G is updated to generate a more efficient distilled dataset.

Algorithm 1 Generative dataset distillation considering global-local coherence

Require: G: generator of conditional GAN; $\mathcal{W}$: parameter of generator G; z: random noises; y: random label; ε: learning rate

Ensure: $\mathcal{W}^*$: the optimized parameters; $\mathcal{S}^*$: the distilled dataset

Train a conditional GAN
for each epoch c = 1 to C **do**
  for each interaction i = 1 to I **do**
    Calculate the conditional GAN loss L_{CGAN}
    Update the parameter of G:
    $\mathcal{W} = \mathcal{W} - \varepsilon \frac{\partial L_{\text{CGAN}}}{\partial \mathcal{W}}$
  end for
end for
Matching the global-local coherence of the original dataset and synthetic dataset:
for each epoch e = 1 to E **do**
  for each interaction i = 1 to I **do**
    Calculate the total loss:
    $L_{\text{total}} = \omega_g L_{\text{global}} + \omega_l L_{\text{local}} + L_{\text{CGAN}}$
    Update the parameter of G:
    $\mathcal{W} = \mathcal{W} - \varepsilon \frac{\partial L_{\text{total}}}{\partial \mathcal{W}}$
  end for
end for
Obtain the optimized parameter of G:
$\mathcal{W}^* = \arg\min_{\mathcal{W}} L_{\text{total}}$
Generate the distilled dataset $\mathcal{S}^*$:
$\mathcal{S}^* = G\left([\boldsymbol{z} \oplus \boldsymbol{y}]; \mathcal{W}^*\right)$

4. Experiments

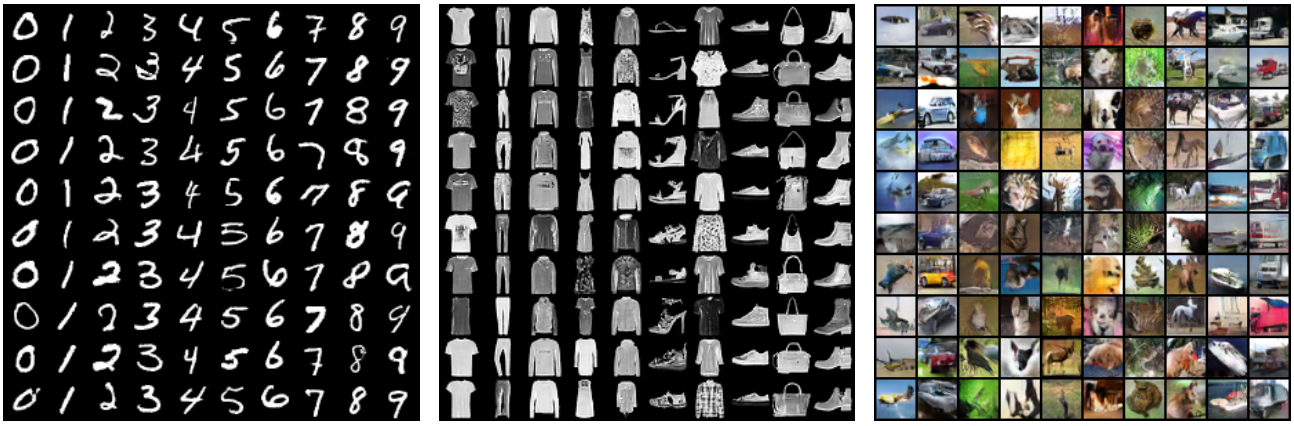
4.1. Experimental Settings

We used three benchmark datasets (MNIST, Fashion MNIST, and CIFAR-10) in the experiments for comparison with other methods. They all have 10 classes and the resolution of images in the three datasets is all 32×32. For comparative methods, we used seven SOTA dataset distillation methods, including dataset condensation (DC) [41], differentiable siamese augmentation (DSA) [42], distribution matching (DM) [43], aligning features (CAFE) [34], kernel inducing point (KIP) [25], matching training trajectories (MTT) [3], and neural feature regression with pooling (FrePo) [44]. We also compared with baseline method CGAN [23] and the generative-based dataset distillation method DiM [35].

To improve the generalization performance and avoid over-reliance on a single network, when optimizing the generator, we applied the model pool to get the randomly initialized model. The model pool has several models such as ConvNet3 [6], ResNet10, and ResNet18 [9]. Randomly selected models are used to match the global and local features of the synthetic dataset and the original dataset. When matching local features between two datasets, we focus on specific intermediate layers that demonstrate a superior ability to extract local features, such as the second layer within ResNet [39]. We conducted three experiments to verify the effectiveness of the proposed method, including benchmark comparison, cross-architecture generalization, and hyperparameter ablation study. All the experimen-

Table 1. Comparation with data selection methods and SOTA dataset distillation methods on three benchmark datasets.

Dataset	MNIST			Fashion MNIST			CIFAR-10		
IPC	1	10	50	1	10	50	1	10	50
Random [41]	64.9±3.5	95.1±0.9	97.9±0.2	51.4±3.8	73.8±0.7	82.5±0.7	14.4±2.0	26.0±1.2	43.4±1.0
Herding [4]	89.2±1.6	93.7±0.3	94.8±0.2	67.0±1.9	71.1±0.7	71.9±0.8	21.5±1.3	31.6±0.7	40.4±0.6
K-Center [5]	89.3±1.5	84.4±1.7	97.4±0.3	66.9±1.8	54.7±1.5	68.3±0.8	21.5±1.3	14.7±0.9	27.0±1.4
Forgetting [32]	35.5±5.6	68.1±3.3	88.2±1.2	42.0±5.5	53.9±2.0	55.0±1.1	13.5±1.2	23.3±1.0	23.3±1.1
DC [41]	91.7±0.5	97.4±0.2	98.8±0.2	70.5±0.6	82.3±0.4	83.6±0.4	28.3±0.5	44.9±0.5	53.9±0.5
DSA [42]	88.7±0.6	97.8±0.1	99.2±0.1	70.6±0.6	84.6±0.3	88.7±0.2	28.8±0.7	52.1±0.5	60.6±0.5
DM [43]	89.9±0.8	97.6±0.1	98.6±0.1	71.5±0.5	83.6±0.2	88.2±0.1	26.5±0.4	48.9±0.6	63.0±0.4
EGM [10]	91.9±0.4	97.9±0.2	98.6±0.1	71.4±0.4	85.4±0.3	87.9±0.2	30.0±0.6	50.2±0.6	60.0±0.4
CAFE [34]	93.1±0.3	97.5±0.1	98.9±0.2	73.7±0.7	83.0±0.4	88.2±0.3	31.6±0.8	50.9±0.5	62.3±0.4
KIP [25]	90.1±0.1	97.5±0.0	98.3±0.1	70.6±0.6	84.6±0.3	88.7±0.2	49.9±0.2	62.7±0.3	68.6±0.2
MTT [3]	91.4±0.9	97.3±0.1	98.5±0.1	75.1±0.9	87.2±0.3	88.3±0.1	46.3±0.8	65.3±0.7	71.6±0.2
FRePo [44]	93.8±0.6	98.4±0.1	99.2±0.1	75.6±0.5	86.2±0.3	89.6±0.1	46.8±0.7	65.5±0.6	71.7±0.2
CGAN [23]	96.1±0.7	97.8±0.3	98.4±0.3	81.5±0.5	84.0±0.2	86.3±0.3	46.4±1.2	62.7±0.9	68.1±0.8
DiM [35]	96.5±0.6	98.6±0.2	**99.2±0.2**	84.5±0.4	88.2±0.2	89.8±0.1	51.3±1.0	66.2±0.5	72.6±0.4
Ours	**97.3±0.3**	**98.8±0.2**	99.1±0.1	**85.5±0.2**	**88.6±0.1**	**90.1±0.1**	**52.3±0.6**	**66.7±0.3**	**73.1±0.2**
Original Dataset		99.6±0.0			93.5±0.1			84.8±0.1	



(a) MNIST (b) Fashion MNIST (c) CIFAR-10

Figure 2. Distilled MNIST, Fashion MNIST, and CIFAR-10 datasets with IPC = 10.

tal results are average accuracy and standard deviation of five networks trained from scratch on the distilled dataset and tested on the original dataset.

4.2. Benchmark Comparison

In this subsection, we verify the effectiveness of the proposed method by comparing it with other methods on three benchmark datasets, i.e., MNIST, Fashion MINIST, and CIFAR-10. We designed three sets of experiments for each dataset. Each set applies IPC = 1, 10, and 50, respectively. ConvNet3 was used as the test model. From Table 1, we can see that our method achieves better performance under majority settings and shows better stability. In most experiments, the accuracy has been improved by about 0.5%, especially when IPC = 1, the proposed method improved the accuracy by about 1% and improved the stability. Figure 2 shows the visualization results on distilled MNIST, fashionMNIST, and CIFAR-10 datasets obtained using the proposed method. Based on the main experimental and visualization results, we can see that the proposed method can improve the performance of generative dataset distillation while maintaining the visual authenticity of the distilled dataset.

4.3. Cross-architecture Generalization

In this subsection, we verify the effectiveness of our method in cross-architecture generalization. Cross-architecture means using distilled images generated by some architectures and testing on other architectures. In our experiments, ConvNet3 and ResNet18 were selected as matching models when optimizing the generator G. To validate the generalization performance on other architectures, we used

Table 2. Cross-architecture generalization ability comparation on CIFAR-10 dataset with IPC = 10.

Method	ConvNet3	ResNet18	AlexNet	VGG11
DSA [42]	52.1±0.4	42.8±1.0	35.9±1.3	43.2±0.5
KIP [25]	47.6±0.9	36.8±1.0	24.4±3.9	42.1±0.4
MTT [3]	64.3±0.7	46.4±0.6	34.2±2.6	50.3±0.8
FRePo [44]	65.5±0.4	57.7±0.7	61.9±0.7	59.4±0.7
DiM [35]	66.2±0.5	69.2±0.3	67.3±0.9	66.8±0.5
Ours	**66.7±0.3**	**71.6±0.2**	**68.0±0.3**	**67.4±0.6**

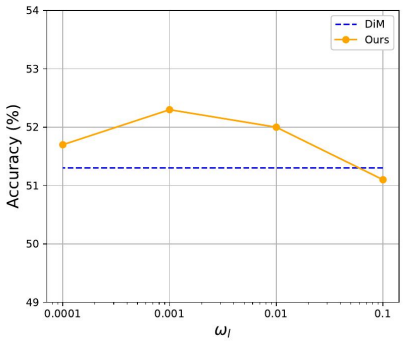


Figure 3. Ablation study of ω_l on CIFAR-10 dataset with IPC = 1.

AlexNet [12] and VGG11 [28]. These architectures were trained on the distilled dataset and tested on the original dataset. We set the IPC = 10 to keep the same setting as the previous methods to make a fair comparison. Table 2 shows that the proposed method outperforms conventional dataset distillation methods in terms of cross-architecture generalization. The distilled dataset demonstrates higher accuracy across different architectures. In comparison with the DiM method, the distillation data derived from the proposed method shows enhancements in performance on various architectures and exhibits better stability.

4.4. Hyperparameter Ablation Study

Since DiM has proved that the weight of global loss ω_g = 0.01 leads to the best performance. Hence, we used the same value of ω_g and set up an experiment on CIFAR-10 with IPC = 1 to explore the impact of the weight of local loss ω_l. As shown in Fig. 3, when the weight of the local loss L_{local} was set to 0.001, the proposed method achieved the highest average accuracy. When the local loss weight is too large, it reduces the impact of global loss L_{global} and conditional GAN loss L_{CGAN}, thereby reducing the accuracy, while a local loss weight that is too small will not allow the generator to effectively learn the local features. Although DiM has shown the best value of ω_g, the impact of the different ω_g values is still worth exploring in future works.

5. Conclusion

This paper has proposed a novel dataset distillation method. During the dataset distillation process, the proposed method takes into account both the global structure and local details, thereby ensuring that high-level semantic information and mid-level feature information are simultaneously distilled into the generative model. Experimental results show that the proposed method outperforms other SOTA dataset distillation methods on three benchmark datasets.

Acknowledgement

This research was supported in part by JSPS KAKENHI Grant Numbers JP21H03456, JP23K11211, and JP23K11141.

References

[1] Laith Alzubaidi, Jinglan Zhang, and et al. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 8:1–74, 2021. 1

[2] Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales. Flexible dataset distillation: Learn labels instead of images. In *Proc. NeurIPS Workshop*, 2020. 2

[3] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proc. CVPR*, pages 4750–4759, 2022. 1, 2, 5, 6, 7

[4] Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding. In *Proc. UAI*, 2010. 6

[5] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Proc. NeurIPS*, 2017. 6

[6] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proc. CVPR*, pages 4367–4375, 2018. 5

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. NeurIPS*, 2014. 3

[8] Ziyao Guo, Kai Wang, George Cazenavette, Hui Li, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching. In *Proc. ICLR*, 2024. 1

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778, 2016. 5

[10] Zixuan Jiang, Jiaqi Gu, Mingjie Liu, and David Z. Pan. Delving into effective gradient matching for dataset condensation. *arXiv preprint arXiv:2208.00311*, 2022. 6

[11] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. In *Proc. ICML*, pages 11102–11118, 2022. 2

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. NeurIPS*, pages 1097–1105, 2012. 7

[13] Saehyung Lee, Sanghyuk Chun, Sangwon Jung, Sangdoo Yun, and Sungroh Yoon. Dataset condensation with contrastive signals. In *Proc. ICML*, pages 12352–12364, 2022. 2

[14] Shiye Lei and Dacheng Tao. A comprehensive survey to dataset distillation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):17–32, 2023. 1, 3

[15] Guang Li, Ren Togo, Takahiro Ogawa, and Miki Haseyama. Soft-label anonymous gastric x-ray image distillation. In *Proc. ICIP*, pages 305–309, 2020. 1

[16] Guang Li, Ren Togo, Takahiro Ogawa, and Miki Haseyama. Compressed gastric image generation based on soft-label dataset distillation for medical data sharing. *Computer Methods and Programs in Biomedicine*, 227:107189, 2022. 1

[17] Guang Li, Bo Zhao, and Tongzhou Wang. Awesome dataset distillation. https://github.com/Guang000/Awesome-Dataset-Distillation, 2022. 3

[18] Guang Li, Ren Togo, Takahiro Ogawa, and Miki Haseyama. Dataset distillation using parameter pruning. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 2023. 3

[19] Guang Li, Ren Togo, Takahiro Ogawa, and Miki Haseyama. Dataset distillation for medical dataset sharing. In *Proc. AAAI Workshop*, pages 1–6, 2023. 1

[20] Guang Li, Ren Togo, Takahiro Ogawa, and Miki Haseyama. Importance-aware adaptive dataset distillation. *Neural Networks*, 2024. 3

[21] Noel Loo, Ramin Hasani, Mathias Lechner, Alexander Amini, and Daniela Rus. Dataset distillation fixes dataset reconstruction attacks. In *Proc. ICLR*, 2024. 1

[22] Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *Proc. AISTATS*, pages 1540–1552, 2020. 2

[23] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 3, 5, 6

[24] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. In *Proc. ICLR*, 2021. 2

[25] Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. In *Proc. NeurIPS*, pages 5186–5198, 2021. 2, 5, 6, 7

[26] Ahmad Sajedi, Samir Khaki, Ehsan Amjadian, Lucy Z. Liu, Yuri A. Lawryshyn, and Konstantinos N. Plataniotis. DataDAM: Efficient dataset distillation with attention matching. In *Proc. ICCV*, pages 17097–17107, 2023. 3

[27] Saptarshi Sengupta, Sanchita Basak, Pallabi Saikia, Sayak Paul, Vasilios Tsalavoutis, Frederick Atiah, Vadlamani Ravi, and Alan Peters. A review of deep learning with special emphasis on architectures, applications and recent trends. *Knowledge-Based Systems*, 194:105596, 2020. 1

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015. 7

[29] Rui Song, Dai Liu, Dave Zhenyu Chen, Andreas Festag, Carsten Trinitis, Martin Schulz, and Alois Knoll. Federated learning via decentralized dataset distillation in resource-constrained edge environments. In *Proc. IJCNN*, pages 1–10, 2023. 1

[30] Ilia Sucholutsky and Matthias Schonlau. Soft-label dataset distillation and text dataset distillation. In *Proc. IJCNN*, pages 1–8, 2021. 2

[31] Jialin Tian, Xing Xu, Zheng Wang, Fumin Shen, and Xin Liu. Relationship-preserving knowledge distillation for zero-shot sketch based image retrieval. In *Proc. ACM MM*, pages 5473–5481, 2021. 2

[32] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In *Proc. ICLR*, 2019. 6

[33] Paul Vicol, Jonathan P Lorraine, Fabian Pedregosa, David Duvenaud, and Roger B Grosse. On implicit bias in overparameterized bilevel optimization. In *Proc. ICML*, pages 22234–22259, 2022. 2

[34] Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. CAFE: Learning to condense dataset by aligning features. In *Proc. CVPR*, pages 12196–12205, 2022. 5, 6

[35] Kai Wang, Jianyang Gu, Daquan Zhou, Zheng Zhu, Wei Jiang, and Yang You. DiM: Distilling dataset into generative model. *arXiv preprint arXiv:2303.04707*, 2023. 1, 5, 6, 7

[36] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. 1, 2

[37] Enneng Yang, Li Shen, Zhenyi Wang, Tongliang Liu, and Guibing Guo. An efficient dataset condensation plugin and its application to continual learning. In *Proc. NeurIPS*, 2023. 1

[38] Ruonan Yu, Songhua Liu, and Xinchao Wang. A comprehensive survey to dataset distillation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):150–170, 2023. 1, 3

[39] Zhecheng Yuan, Zhengrong Xue, Bo Yuan, Xueqian Wang, Yi Wu, Yang Gao, and Huazhe Xu. Pre-trained image encoder for generalizable visual reinforcement learning. In *Proc. NeurIPS*, 2022. 5

[40] Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, and Xia Hu. Data-centric ai: Perspectives and challenges. In *Proc. SDM*, pages 945–948, 2023. 1

[41] Bo Zhao and Hakan Bilen. Dataset condensation with gradient matching. In *Proc. ICLR*, 2021. 1, 2, 5, 6

[42] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *Proc. ICML*, pages 12674–12685, 2021. 2, 5, 6, 7

[43] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proc. WACV*, 2023. 1, 3, 5, 6

[44] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. In *Proc. NeurIPS*, 2022. 2, 5, 6, 7