Importance-Aware Adaptive Dataset Distillation

Guang Li[a], Ren Togo[b], Takahiro Ogawa[b], Miki Haseyama[b]

[a]*Education and Research Center for Mathematical and Data Science, Hokkaido University, N-12, W-7, Kita-Ku, Sapporo, 060-0812, Japan*

[b]*Faculty of Information Science and Technology, Hokkaido University, N-14, W-9, Kita-Ku, Sapporo, 060-0814, Japan*

Abstract

Herein, we propose a novel dataset distillation method for constructing small informative datasets that preserve the information of the large original datasets. The development of deep learning models is enabled by the availability of large-scale datasets. Despite unprecedented success, large-scale datasets considerably increase the storage and transmission costs, resulting in a cumbersome model training process. Moreover, using raw data for training raises privacy and copyright concerns. To address these issues, a new task named dataset distillation has been introduced, aiming to synthesize a compact dataset that retains the essential information from the large original dataset. State-of-the-art (SOTA) dataset distillation methods have been proposed by matching gradients or network parameters obtained during training on real and synthetic datasets. The contribution of different network parameters to the distillation process varies, and uniformly treating them leads to degraded distillation performance. Based on this observation, we propose an importance-aware adaptive dataset distillation (IADD) method that can improve distillation performance by automatically assigning importance weights to different network parameters during distillation, thereby synthesizing more robust distilled datasets. IADD demonstrates superior performance over other SOTA dataset distillation methods based on parameter matching on multiple benchmark datasets and outperforms them in terms of cross-architecture generalization. In addition, the analysis of self-adaptive weights demonstrates the effectiveness of IADD. Furthermore, the effectiveness of IADD is validated in a real-world medical application such as COVID-19 detection.

Keywords: Dataset distillation, parameter matching, importance-aware adaptive distillation.

1. Introduction

In the past few years, deep learning [1, 2] has achieved tremendous success in diverse fields, including computer vision [3], natural language processing [4], and speech recognition [5]. The famous deep learning models such as AlexNet [6], ResNet [7], BERT [8], ViT [9], CLIP [10], Stable Diffusion [11], and ChatGPT [12] depend on large-scale datasets for training. Nevertheless, the management of large-scale datasets presents a significant challenge, encompassing the intricate tasks of storage, transmission, and preprocessing [13]. For example, the challenge is underscored by the immense volume of data generated by high-resolution scans, such as MRI and CT scans, wherein a single 3D CT scan of the chest comprises thousands of individual slices, each containing millions of pixels [14]. Moreover, to achieve suitable performance, training on large-scale datasets requires extensive computation, almost of the order of thousands of GPU hours [15, 16, 17]. This problem can be alleviated by selecting representative training samples from large datasets, which is known as core-set or instance selection [18, 19]. Some advanced instance selection methods have been proposed in recent years [20, 21, 22]. Nevertheless, an upper bound exists on the compression rate achievable using data selection methods because certain original data cannot be discarded. Furthermore, privacy and copyright issues are associated with the use of raw data for training [23, 24].

To address the aforementioned issues, a novel task named dataset distillation was proposed by Wang et al. [25] in 2018. The objective of dataset distillation is to generate a small informative dataset ($\mathcal{D}_{\text{distill}}$) that allows models trained on it to perform comparably to those trained on the original dataset ($\mathcal{D}_{\text{original}}$), where the number of images in the synthetic dataset is considerably less than that in the original dataset. Therefore, unlike dataset selection, dataset distillation method can compress a large dataset into a smaller dataset comprising only a few images, thereby substantially improving the dataset compression rate. Because the generated distilled datasets do not contain raw data, the concerns of privacy and copyright can be addressed. Downstream applications have substantially benefited from dataset distillation, such as continual learning [26, 27], privacy [28, 29, 30, 31], biomedical [32, 33, 34], federated learning [35, 36, 37], graph neural networks [38, 39], neural architecture search [40, 41], and black box optimization [42, 43].

Email addresses: guang@lmd.ist.hokudai.ac.jp (Guang Li[a]), togo@lmd.ist.hokudai.ac.jp (Ren Togo[b]), ogawa@lmd.ist.hokudai.ac.jp (Takahiro Ogawa[b]), mhaseyama@lmd.ist.hokudai.ac.jp (Miki Haseyama[b])

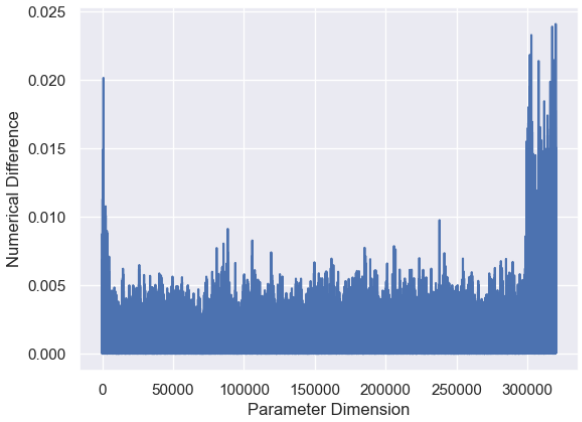


Figure 1: Visualization results of the numerical difference between parameters of teacher and student networks in the corresponding dimensions (A three-depth ConvNet on CIFAR-10).

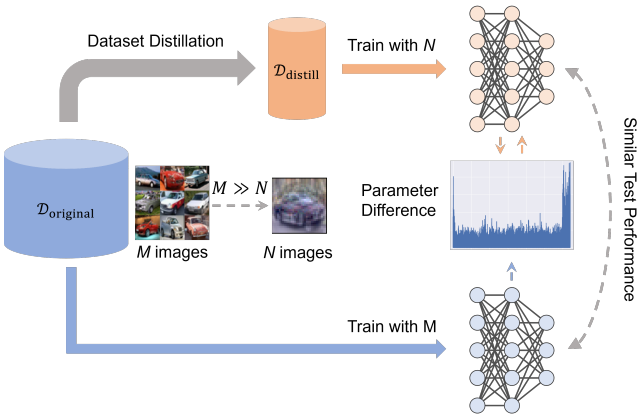


Figure 2: Concept of the proposed method. IADD aims to train the student network parameters by leveraging a distilled dataset that aligns with the teacher network parameters derived from the original large dataset. Because the parameter pairs in the teacher and student networks are different, we deal with these parameters using different importance weights.

Dataset distillation has attracted increasing attention in recent years [44, 45, 46]. The original dataset distillation task is typically considered a meta-learning problem involving bi-level optimization [25]. It optimizes the synthetic dataset by minimizing the classification loss on the original dataset in the outer loop and subsequently formulates the network parameters as functions of the learnable synthetic dataset in the inner loop. Because gradient descent involves thousands to millions of steps, this recursive computation renders it unsuitable for real-world applications. Zhao et al. [41] proposed learning synthetic datasets by matching gradients or network parameters between real and synthetic data, while training teacher (using $\mathcal{D}_{\text{original}}$) and student (using $\mathcal{D}_{\text{distill}}$) networks to avoid unrolling the recursive computation graph. Several methods have been subsequently proposed to improve dataset distillation performance via differentiable data augmentation [47], feature alignment [48, 49], contrastive signaling [50], and trajectory matching [51, 52].

Although the concept of matching gradients or network parameters is intuitive and effective, a generally overlooked problem, i.e., whether each network parameter has the same importance in the dataset distillation process remains. Different network parameters exhibit different matching difficulties during the distillation process. In particular, we distilled the CIFAR-10 dataset [53] using a three-depth ConvNet containing 320,010 parameters. For example, as depicted in Fig. 1, the numerical difference between the teacher and student networks of a few network parameters is low, whereas others are high. This implies that different network parameters differently contribute to the distillation process and equally treating these parameters affects the distillation performance. Based on this observation, different network parameters can be attributed to different importance weights, and these weights can be subsequently optimized during the distillation process. Therefore, in this study, we explore this characteristic to further improve dataset distillation performance based on gradient (parameter) matching.

Herein, we propose a novel dataset distillation method for solving the aforementioned above. The concept of the proposed method is depicted in Fig. 2. Importance-aware adaptive dataset distillation (IADD) can improve distillation performance by automatically assigning importance weights to different network parameters and synthesizing more robust distilled datasets. This ensures that crucial network parameters receive a higher weight, resulting in an improved contribution to the overall performance, whereas the effect of unimportant parameters is minimized. Furthermore, the optimization process is performed iteratively to refine the importance weights of network parameters until a desired level of performance is achieved. Instead of equally treating all network parameters similar to conventional distillation methods, IADD considers the relative importance of each parameter and accordingly optimizes the importance weights. This leads to more precise distillation and improved performance. We performed extensive experiments in different settings to demonstrate the superiority of IADD.

In summary, our study makes the following significant contributions:

- We propose a novel dataset distillation method, IADD, for synthesizing small informative datasets that preserve the information of the large original datasets.
- We design a framework that can automatically assign importance weights to different network parameters during distillation and synthesize more robust distilled datasets.
- We demonstrate that IADD achieves superior performance over other state-of-the-art (SOTA) dataset distillation methods based on parameter matching on multiple benchmark datasets and is more effective in cross-architecture generalization.
- We apply IADD to a real-world medical application such as COVID-19 detection.

This study extends our prior research, dataset distillation using parameter pruning (DDPP), which primarily focused on parameter pruning [54]. In this study, we augment our earlier work in the following aspects: First, we introduced a novel dataset distillation approach capable of dynamically assigning adaptive weights to various parameters instead of directly eliminating them. Second, we performed extensive experiments across four distinct datasets and varied settings to demonstrate the superior performance of IADD. Finally, we visualized the parameter difference and self-adaptive weights and performed a detailed analysis to gain a more comprehensive understanding of the reasons underlying the improved performance of IADD.

This paper is structured as follows. Section 2 provides an overview of related works. Section 3 details the proposed method. Section 4 presents the experimental results and analysis. Section 5 presents a discussion of our findings. Finally, Section 6 provides the concluding remarks.

2. Related Works

2.1. Dataset Distillation Based on Performance Matching

In this section, we introduce dataset distillation methods based on performance matching. Performance matching-based methods are designed to refine a distilled dataset to enable neural networks trained on it to achieve minimal loss when applied to the original dataset. Consequently, this ensures matched performance between models trained using both the distilled and original datasets. The first dataset distillation method was proposed by Wang et al. [25]. In their method, model weights are defined as functions of distilled images and optimized using gradient-based hyperparameter optimization, which is widely used in meta-learning [55]. Subsequently, certain methods extend the original method using flexible labels [56] or soft-label [57] and add momentum [58] to improve the distillation performance.

Backpropagation is used in meta-learning methods to calculate the gradient of validation loss on synthetic datasets, which necessitates bi-level optimization [59, 60]. Consequently, the optimization of outer loops necessitates extensive computing costs, and GPU memory requirements increase as the number of inner loops increases [61]. A limited number of inner loops results in insufficient inner optimization and performance bottlenecks, and upscaling this learning schema to large models is also inconvenient. However, kernel inducing point (KIP), can solve this problem by performing convex optimization, resulting in a closed-form solution, avoiding the requirement for extensive inner loop training [62]. KIP involves initializing a labeled support set and iterating over a process where a random kernel and batches from the support and target dataset are sampled. It computes the kernel ridge-regression loss and updates the support set based on the loss. KIP includes variations like randomly augmenting sampled target batches and introducing a corruption fraction to the data, resulting in highly corrupted yet effective datasets. Several methods extended from KIP, which considerably improve distillation performance and efficiency, have been proposed such as infinitely wide convolutional networks [63], neural feature regression [64], and random feature approximation [65].

2.2. Dataset Distillation Based on Parameter Matching

In this section, we illustrate dataset distillation methods based on parameter matching. The first parameter matching method was proposed by Zhao et al. [41], which is also known as dataset condensation. While performance matching focuses on optimizing the performance of networks trained on synthetic datasets, parameter matching optimizes the performance of networks trained on the original and synthetic datasets by ensuring the consistency of the trained network parameters. Several studies have extended parameter matching using differentiable data augmentation [47], contrastive signaling [50], long-range trajectory matching [51], and parameter pruning [54].

As noted in our earlier study DDPP [54], some parameters are challenging to match in dataset distillation, resulting in a deteriorated distillation performance. To address this issue, we used parameter pruning to eliminate such challenging parameters during the distillation process. Nevertheless, removing these parameters is imprudent and can negatively influence the number of network model parameters. Therefore, herein, we propose a novel method that can assign adaptive weights to different parameters instead of directly removing parameters to solve this problem.

2.3. Dataset Distillation Based on Distribution Matching

We present several dataset distillation methods based on distribution matching here. Distribution matching produces synthetic data with a distribution close to those of the original data in a family of embedding spaces using the maximum mean discrepancy (MMD [66]). The first distribution matching method proposed by Zhao et al. [49] used the output embeddings of neural networks without the last linear layer. For each class of synthetic and original datasets, the mean vector (center) is preferred to be close. Subsequently, another method has been proposed to force the statistics of features extracted by each network layer to be consistent [48]. Although these methods reduce the synthesis cost and are generally applicable to larger datasets, the distillation performance is not improved. For more details regarding the literature on dataset distillation, please refer to the recent survey papers [44, 45, 46] or Awesome-Dataset-Distillation [67].

2.4. Dataset Distillation for Medical Task

Dataset distillation, which can reduce the complexity and volume of medical data improves efficiency in medical tasks. It has been applied to several medical scenarios, such as gastritis detection [32], COVID-19 detection [34], skin lesion classification [68], and medical data sharing [33]. However, in the dynamic field of medical applications, diverse innovative approaches have emerged. For example, hybrid solutions combine conventional machine learning, deep learning, and domain-specific knowledge, creating a synergy of methodologies [69, 70]. These methods improve interpretability, while

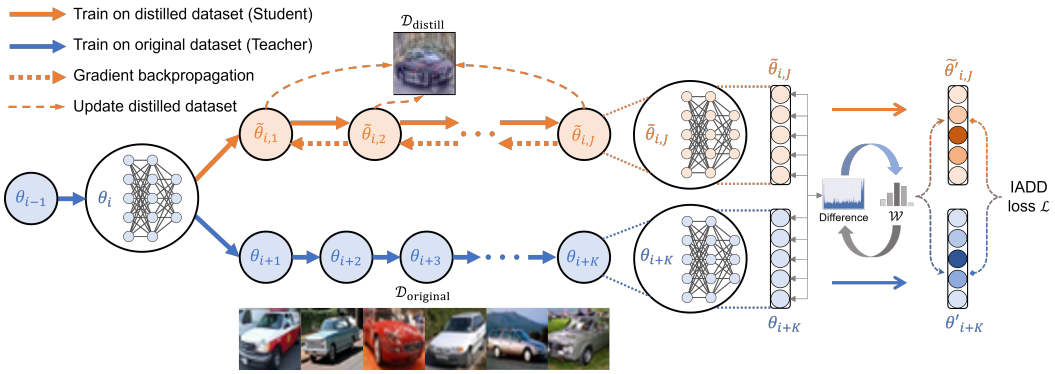


Figure 3: Overview of the proposed method. The main objective is to match the student network parameters $\tilde{\theta}'_{i,J}$ with the teacher network parameters θ'_{i+K} using the IADD loss $\mathcal{L}$. $\tilde{\theta}_{i,J}$ and θ_{i+K} denote intermediate model parameters without the processing of self-adaptive weights $\mathcal{W}$. i represents the random start timestamp of the teacher and student parameters. J and K represent gradient descent updates of teacher and student parameters, respectively.

utilizing the predictive potential of deep learning, which is particularly valuable for medical diagnosis and risk assessment [71]. Inspired by neuroscience principles, neuroheuristics have made notable strides in medical applications [72, 73]. These methods emulate neural processes, thereby improving the comprehension and analysis of intricate medical data. Neural networks designed to mirror neural functionality have improved diagnostic accuracy and decision support systems [74]. Multimodal fusion methods play a crucial role in integrating diverse data sources, including medical imaging, electronic health records, and genomic data [75, 76]. This integration provides valuable insights crucial for personalized medicine and disease prediction in an era marked by the prevalence of diverse medical data types [77, 78]. Therefore, recognizing the importance of dataset distillation methods in the medical domain and contemplating their potential synergy and collaboration with other methodologies is imperative.

3. Methodology

Figure 3 provides an overview of the proposed method. IADD aims to train the student network parameters by leveraging a distilled dataset $\mathcal{D}_{\text{distill}}$ that aligns with the teacher network parameters derived from the original large dataset $\mathcal{D}_{\text{original}}$. The proposed method involves three main stages: (1) training the teacher and student networks, (2) matching teacher and student parameters, and (3) generating an optimized distilled dataset.

3.1. Teacher and Student Network Training

In the proposed method, N teacher networks are first pretrained on the original dataset $\mathcal{D}_{\text{original}}$ for the distillation process, and their snapshot parameters are saved at each epoch. Note that pretraining of many teacher networks is only performed once for one dataset and can be reused. To improve the robustness and diversity of knowledge transfer during the distillation process, we use multiple teacher networks. Each teacher network represents a distinct snapshot of the model during training, capturing unique aspects of the data and the learning process. These snapshots collectively provide a more comprehensive understanding of the intricacies of the dataset and model training process. By utilizing multiple teacher networks, we can reduce the risk of overfitting to a single snapshot or encountering biases in a particular training stage. This ensemble of teacher networks contributes to the stability and reliability of the dataset distillation process, ultimately enhancing the generalization ability and effectiveness of IADD. Initialization of the distilled is performed using random samples from the original dataset as $\mathcal{D}_{\text{distill}} \sim \mathcal{D}_{\text{original}}$. The number of images in the distilled dataset can be obtained by IPC, where IPC denotes the number of distilled images per class and is typically used in the dataset distillation task. The teacher parameters are defined as a sequence of parameters $\{\theta_i\}_0^I$, where I represents the number of training steps. To initiate student parameters, a teacher parameter is selected at a random start timestamp i. The student parameters are initialized as $\tilde{\theta}_i = \theta_i$. To avoid using less informative parts of the teacher parameters, we introduce an upper bound I^+ on the random start timestamp i. Updating the student and teacher parameters involves executing J and K gradient descent updates, respectively, where $J \ll K$. For each student update j, a minibatch $b_{i,j}$ is sampled from the distilled dataset $\mathcal{D}_{\text{distill}}$ as follows:

$$b_{i,j} \sim \mathcal{D}_{\text{distill}}. \tag{1}$$

Subsequently, the student parameters $\tilde{\theta}$ undergo j updates using the cross-entropy loss ℓ to optimize performance as follows:

$$\tilde{\theta}_{i,j+1} = \tilde{\theta}_{i,j} - \alpha \nabla_{\tilde{\theta}_{i,j}} \ell(\mathcal{A}(b_{i,j})), \tag{2}$$

where α represents a trainable learning rate that can be updated during the training process. Moreover, the differentiable data augmentation module $\mathcal{A}$, as proposed in [47], is used to improve the distillation performance. In particular, the loss ℓ is computed as the average of the cross-entropy between the predictions of the student network and the corresponding one-hot labels. And $\nabla_{\tilde{\theta}_{i,j}} \ell$ denotes the gradients of ℓ based on $\tilde{\theta}_{i,j}$. The updates of student parameters are performed via stochastic gradient descent (SGD).

3.2. Teacher and Student Parameters Matching

Once J updates are performed, the student parameters $\tilde{\theta}_{i,J}$ can be obtained, which are then trained using the distilled dataset

Algorithm 1 Importance-Aware Adaptive Dataset Distillation

Input: $\{\theta_i\}_0^I$: teacher parameters trained on $\mathcal{D}_{\text{original}}$; $\mathcal{W}_0$: initial value for $\mathcal{W}$; α_0: initial value for α; $\mathcal{A}$: differentiable augmentation function; ϵ: threshold for pruning; T: number of distillation iterations; J: number of updates for the student network; K: number of updates for the teacher network; I^+: upper bound of the sample range of i.

Output: optimized trainable learning rate α^*, optimized self-adaptive weights $\mathcal{W}^*$, and optimized distilled dataset $\mathcal{D}^*_{\text{distill}}$.

Initialize distilled dataset: $\mathcal{D}_{\text{distill}} \sim \mathcal{D}_{\text{original}}$
Initialize trainable learning rate: $\alpha = \alpha_0$
for each distillation iteration $t = 0$ to $T - 1$ **do**
  Select random start timestamp $i < I^+$
  Initialize student network using teacher parameter: $\tilde{\theta}_i = \theta_i$
  for each student update $j = 0$ to $J - 1$ **do**
    Sample a minibatch of the distilled dataset: $b_{i,j} \sim \mathcal{D}_{\text{distill}}$
    Update student network using the cross-entropy loss ℓ:
    $\tilde{\theta}_{i,j+1} = \tilde{\theta}_{i,j} - \alpha \nabla_{\tilde{\theta}_{i,j}} \ell(\mathcal{A}(b_{i,j}))$
  end for
  Obtain importance-aware parameters $\tilde{\theta}'_{i,J}$ and θ'_{i+K} with Eqs. (3)–(8)
  Calculate the IADD loss:
  $\mathcal{L} = \|\tilde{\theta}'_{i,J} - \theta'_{i+K}\|_2^2 \ / \ \|\theta_i - \theta_{i+K}\|_2^2$
  Update α, $\mathcal{W}$, and $\mathcal{D}_{\text{distill}}$ with with Eqs. (10)–(13)
end for

$\mathcal{D}_{\text{distill}}$ with the student network initialized. Similarly, we obtain the teacher parameters θ_{i+K} trained on the original dataset $\mathcal{D}_{\text{original}}$ with K updates, which are saved in the pretrained snapshots. Subsequently, the initiated parameters θ_i, student parameters $\tilde{\theta}_{i,J}$, and teacher parameters θ_{i+K} are converted into 1D vectors by concatenating the flattened parameters of each layer as follows:

$$\theta_i = [\theta_i^1, \theta_i^2, \cdots, \theta_i^P], \tag{3}$$

$$\tilde{\theta}_{i,J} = [\tilde{\theta}_{i,J}^1, \tilde{\theta}_{i,J}^2, \cdots, \tilde{\theta}_{i,J}^P], \tag{4}$$

$$\theta_{i+K} = [\theta_{i+K}^1, \theta_{i+K}^2, \cdots, \theta_{i+K}^P]. \tag{5}$$

Here, P denotes the total number of parameters. The transformation creates a unified parameter vector by aggregating parameters from different layers. The flattened parameters streamline the subsequent operations such as parameter matching and weight allocation, facilitating the model alignment and dataset distillation processes. We define the self-adaptive weights and importance-aware parameters as follows:

$$\mathcal{W} = [w^1, w^2, \cdots, w^P], \tag{6}$$

$$\tilde{\theta}'_{i,J} = \mathcal{W}\tilde{\theta}_{i,J} = [w^1\tilde{\theta}_{i,J}^1, w^2\tilde{\theta}_{i,J}^2, \cdots, w^P\tilde{\theta}_{i,J}^P], \tag{7}$$

$$\theta'_{i+K} = \mathcal{W}\theta_{i+K} = [w^1\theta_{i+K}^1, w^2\theta_{i+K}^2, \cdots, w^P\theta_{i+K}^P], \tag{8}$$

The self-adaptive weights $\mathcal{W}$ are initialized with $\mathcal{W}_0$, a unit vector with corresponding dimensions. The IADD loss $\mathcal{L}$ calculates the squared L_2 error between the importance-aware student parameters $\tilde{\theta}'_{i,J}$ and teacher parameters θ'_{i+K} as follows:

$$\mathcal{L} = \frac{\|\tilde{\theta}'_{i,J} - \theta'_{i+K}\|_2^2}{\|\theta_i - \theta_{i+K}\|_2^2}. \tag{9}$$

where the distance between the teacher parameters at starting and ending updates $\theta_i - \theta_{i+K}$, ensuring that the late training stage of teacher networks can still provide sufficient supervision, even if it has converged, thereby allowing for a more effective comparison between the student and teacher parameters. Automatically assigning importance weights to different network parameters during distillation can improve the contribution of essential parameters and penalize unimportant parameters, improving the distillation performance and generating more robust distilled datasets.

3.3. Optimized Distilled Dataset Generation

Finally, we minimize the IADD loss $\mathcal{L}$ with updates on sampled distilled dataset $\mathcal{A}(b_{i,j})$ via SGD. Subsequently, we optimize the trainable learning rate α, self-adaptive weights $\mathcal{W}$, and distilled dataset $\mathcal{D}_{\text{distill}}$ through all J updates of the student network as follows:

$$\alpha_{i,j+1} = \alpha_{i,j} - \mu\nabla_{\alpha_{i,j}}\mathcal{L}(\mathcal{A}(b_{i,j})), \tag{10}$$

$$\mathcal{W}_{i,j+1} = \mathcal{W}_{i,j} - \eta\nabla_{\mathcal{W}_{i,j}}\mathcal{L}(\mathcal{A}(b_{i,j})), \tag{11}$$

$$\mathcal{D}_{\text{distill}_{i,j+1}} = \mathcal{D}_{\text{distill}_{i,j}} - \zeta\nabla_{\mathcal{D}_{\text{distill}_{i,j}}}\mathcal{L}(\mathcal{A}(b_{i,j})), \tag{12}$$

where $\nabla_{\alpha_{i,j}}\mathcal{L}$, $\nabla_{\mathcal{W}_{i,j}}\mathcal{L}$ and $\nabla_{\mathcal{D}_{\text{distill}_{i,j}}}\mathcal{L}$ represent the gradients of $\mathcal{L}$ based on $\alpha_{i,j}$, $\mathcal{W}_{i,j}$ and $\mathcal{D}_{\text{distill}_{i,j}}$, respectively. μ, η, and ζ represent the corresponding learning rates. The self-adaptive weights $\mathcal{W}$ are updated to optimize the IADD process, which can improve the performance of the distilled network. The generated distilled dataset is then used to train the student network in the next iteration, and this process continues until convergence. Finally, we can obtain the optimized learning rate α^*, self-adaptive weights $\mathcal{W}^*$, and distilled dataset $\mathcal{D}^*_{\text{distill}}$ as follows:

$$\alpha^*, \mathcal{W}^*, \mathcal{D}^*_{\text{distill}} = \arg\min \mathcal{L}(\alpha, \mathcal{W}, \mathcal{D}_{\text{distill}}; \tilde{\theta}). \tag{13}$$

The distillation process is summarized in Algorithm 1.

The optimized learning rate α^* can serve as an adaptive regulator for the number of updates performed on the student and teacher networks (J and K, respectively), simplifying the distillation process optimization. The optimized self-adaptive weights $\mathcal{W}^*$ maximize the contribution of parameter matching of different importance to dataset distillation, resulting in higher precise distillation and improved performance.

Once we obtain the optimized distilled dataset $\mathcal{D}^*_{\text{distill}}$, we can use it to train various networks. Because of the significantly reduced volume of distilled data compared with the original data, the distilled dataset offers notable benefits. For example, it

leads to substantial savings in data storage space, which is particularly advantageous when dealing with large-scale datasets. Furthermore, the streamlined dataset size increases the model training speed, thereby reducing computational resource requirements and allowing for more rapid model iteration. In addition, the distilled dataset shows potential for diverse downstream applications. In particular, in contexts where data privacy is of utmost importance, such as in the medical imaging field and other sensitive domains, the distilled dataset can play a pivotal role.

4. Experiments

In this section, we conduct extensive experiments in different settings to demonstrate the superiority of IADD. We have provided the details of our experimental settings in Section 4.1. Sections 4.2, 4.3, 4.4, and 4.5 report the benchmark comparison, cross-architecture generalization, analysis of self-adaptive weights, and real-world medical application, respectively.

4.1. Experimental Settings

Herein, three benchmark datasets, namely, CIFAR-10 [53], CIFAR-100 [53], and Tiny ImageNet [80], were used for the dataset distillation task. The CIFAR-10 and CIFAR-100 datasets comprise images of size 32×32, whereas the Tiny ImageNet dataset comprises images of size 64×64. The CIFAR-10 dataset comprises 60,000 color images with 10 classes. Within each class, there are 6,000 images, which are further divided into 50,000 training and 10,000 test images. The CIFAR-100 dataset mirrors the CIFAR-10 dataset in structure, but offers a more extensive variety with 100 classes. It contains 60,000 color images with each class comprising 600 images. The training set includes 500 images per class, and the test set includes 100 images per class. The Tiny ImageNet dataset comprises 200 training classes with each class comprising 500 color images. In addition to the training set, a test set is present, containing 10,000 images with 200 classes.

In our experiments, we evaluated the proposed method on a large COVID-19 CXR dataset [81] to demonstrate its effectiveness in real-world medical applications, namely, COVID-19 detection[1]. The dataset comprises four classes: COVID-19 with 3,616 images, Lung Opacity with 6,012 images, Normal with 10,192 images, and Viral Pneumonia with 1,345 images. The split ratio of the training and test sets is 8:2 for each class, resulting in 16,933 images for training and 4,232 images for testing. We resized the CXR images from the original 224×224 to 112×112 for the dataset distillation process. The resolution of images in the datasets was selected to ensure that the experiments were performed on datasets with varying complexities. The training set of each dataset is used in the distillation process to generate the distilled data, and the test phase is performed on the test set of the original datasets by the student network.

[1] https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database

Extensive experiments were performed to evaluate the effectiveness of the proposed method across various settings, including benchmark comparison, cross-architecture generalization, self-adaptive weight analysis, and real-world medical application. For benchmark comparison methods, we used three instance selection methods: random selection (Random), example forgetting (Forgetting) [82], and herding method (Herding) [83]. In addition, we used 10 SOTA dataset distillation methods: dataset condensation [41], differentiable siamese augmentation (DSA) [47], distribution matching (DM) [49], aligning features (CAFE) [48], CAFE + DSA [48], effective gradient matching (EGM) [79], dataset condensation with contrastive signals (DCC) [50], matching training trajectories (MTT) [51], dataset distillation using parameter pruning (DDPP) [54], and kernel inducing point (KIP) [63].

This study presents experimental results, which involve the training of five networks from scratch on the distilled dataset, with the average accuracy and standard deviation reported as performance metrics. For the other SOTA methods, we directly used the reported accuracy in the original papers for fair comparison. We adopted a sample 128-width ConvNet [84] as the network structure for dataset distillation. The architecture of ConvNet comprises multiple convolution blocks, each containing a 3×3 convolution layer with 128 filters, instance normalization, ReLU activation, and 2×2 average pooling with stride two. A single linear layer generates logits after the convolution blocks. The number of such blocks is determined by the dataset resolution and specified for each benchmark dataset. To ensure that the distillation process does not collapse when the self-adaptive weights $\mathcal{W}$ approach zero, we set the learning rate of the SGD optimizer to a low value, such as 1e-4, which can control the update speed of the self-adaptive weights $\mathcal{W}$.

To evaluate the effectiveness of the proposed method in real-world medical applications, we performed experiments using training from scratch (From Scratch) and transfer learning as two baseline methods. In addition, we used six SOTA self-supervised learning methods: simple siamese self-supervised learning (SimSiam) [15], bootstrap your own latent (BYOL) [16], cross-view self-supervised learning (Cross) [85], self-knowledge distillation based self-supervised learning (SKD) [86], masked autoencoder (MAE) [87], and region-guided masked image modeling (RGMIM) [88]. These self-supervised learning methods generally use ResNet50 [7] or ViT-Base [9] as their network structures. The performance of each method was evaluated in terms of four-class accuracy on a large COVID-19 CXR dataset.

4.2. Benchmark Comparison

This section presents a comparative analysis of the proposed dataset distillation method against other SOTA methods on three benchmark datasets: CIFAR-10, CIFAR-100, and Tiny ImageNet. The proposed method uses zero-phase component analysis (ZCA) whitening with default parameters for distillation performance improvement [51]. We used a three-depth ConvNet (ConvNetD3) for CIFAR-10 and CIFAR-100 and a four-depth ConvNet (ConvNetD4) for Tiny ImageNet.

Table 1: Test accuracy compared with SOTA dataset distillation methods on several benchmark datasets. IPC represents the number of distilled images per class. The first three comparison methods are instance selection methods, and the others are SOTA dataset distillation methods. All the presented results are the average value obtained over five trials.

Dataset	CIFAR-10			CIFAR-100			Tiny ImageNet	
IPC	1	10	50	1	10	50	1	10
Random	14.4±2.0	26.0±1.2	43.4±1.0	4.2±0.3	14.6±0.5	30.0±0.4	1.4±0.1	5.0±0.2
Forgetting	13.5±1.2	23.3±1.0	23.3±1.1	4.5±0.2	15.1±0.3	30.5±0.3	1.6±0.1	5.1±0.2
Herding	21.5±1.2	31.6±0.7	40.4±0.6	8.4±0.3	17.3±0.3	33.7±0.5	2.8±0.2	6.3±0.2
DC [41]	28.3±0.5	44.9±0.5	53.9±0.5	12.8±0.3	25.2±0.3	29.7±0.3	5.3±0.2	11.1±0.3
DSA [47]	28.8±0.7	52.1±0.5	60.6±0.5	13.9±0.3	32.3±0.3	42.8±0.4	6.6±0.2	16.3±0.2
DM [49]	26.0±0.8	48.9±0.6	63.0±0.4	11.4±0.3	29.7±0.3	43.6±0.4	3.9±0.2	13.5±0.3
CAFE [48]	30.3±1.1	46.3±0.6	55.5±0.6	12.9±0.3	27.8±0.3	37.9±0.3	-	-
CAFE + DSA [48]	31.6±0.8	50.9±0.5	62.3±0.4	14.0±0.3	31.5±0.2	42.9±0.2	-	-
EGM [79]	30.0±0.6	50.2±0.6	60.0±0.4	12.7±0.4	31.1±0.3	-	-	-
DCC [50]	34.0±0.7	54.5±0.5	64.2±0.4	14.6±0.3	33.5±0.3	40.0±0.3	-	-
MTT [51]	46.3±0.8	65.3±0.7	71.6±0.2	24.3±0.3	40.1±0.4	47.7±0.2	8.8±0.3	23.2±0.2
DDPP [54]	46.4±0.6	65.5±0.3	71.9±0.2	24.6±0.1	**43.1±0.3**	48.4±0.3	-	-
IADD	**46.5±1.1**	**66.7±0.8**	**72.6±0.3**	**25.2±0.1**	42.7±0.5	**49.0±0.3**	**9.6±0.4**	**24.1±0.3**
Original Dataset	84.8±0.1			56.2±0.3			37.6±0.4	

Table 2: Test accuracy compared with KIP [63].

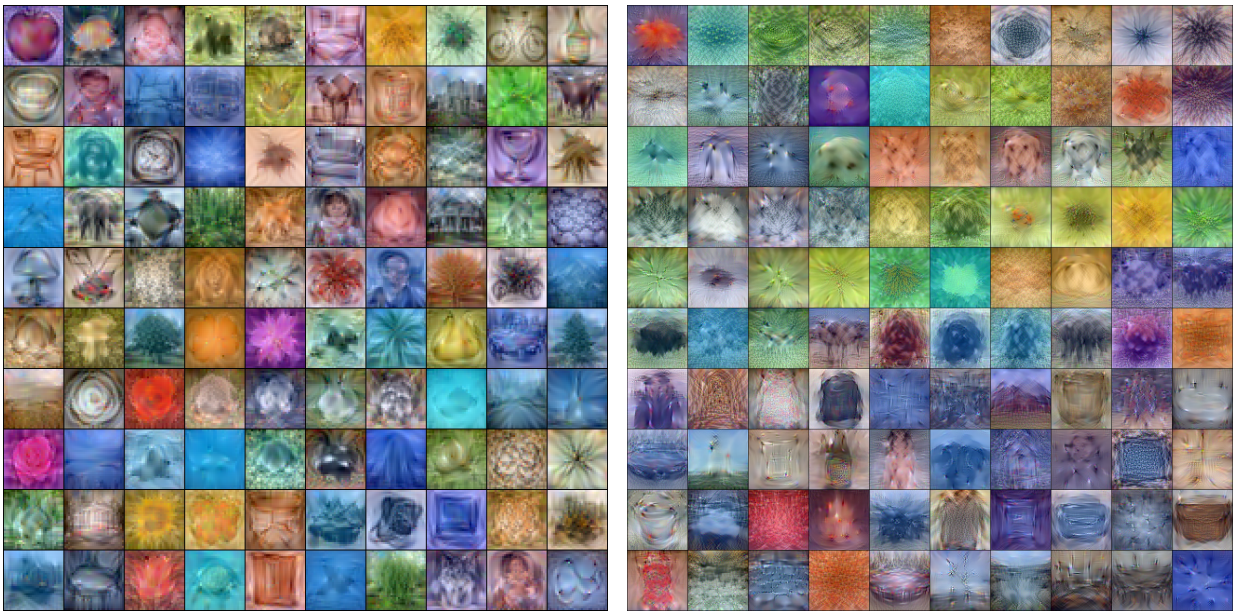
	IPC	KIP-1024	KIP-128	IADD-128
CIFAR-10	1	49.9	38.3	**46.5**
	10	62.7	57.6	**66.7**
	50	68.6	65.8	**72.6**
CIFAR-100	1	15.7	18.2	**25.2**
	10	28.3	32.8	**42.7**
	50	-	-	**49.0**

We pretrained 200 teacher networks for CIFAR-10 and CIFAR-100 and 100 teacher networks for Tiny ImageNet, each for 50 epochs. IPC denotes the number of distilled images per class. The number of distillation iterations T was set to 5,000, and the IPC was varied across datasets, with values of 1, 10, and 50 for CIFAR-10 and CIFAR-100 and 1 and 10 for Tiny ImageNet. The above settings are the same as those in the previous dataset distillation methods. For a fair comparison with KIP [63], we used their original 1024-width ConvNet (KIP-1024) and 128-width ConvNet (KIP-128). In addition, we used custom ZCA in the distillation and evaluation processes.

The experimental results in Table 1 reveal the superiority of IADD over the existing dataset selection and dataset distillation methods across most IPC settings. For CIFAR-10 with IPC = 10, IADD achieved a 1.2% increase in accuracy compared with our previous DDPP method. For CIFAR-100 with IPC = 10, IADD achieved a 2.6% increase in accuracy compared with the SOTA method MTT. Notably, when all importance weights are uniform, IADD degrades to MTT, indicating the effectiveness of the self-adaptive weights. DDPP may not be suitable for large-scale datasets because it tends to collapse during the distillation process because of the pruned parameters. In contrast, the proposed method based on self-adaptive weights is expected to perform efficiently on large-scale datasets, because it can effectively adapt to data characteristics and adjust the importance of different parameters during the distillation process.

Figure 4: Distilled CIFAR-10 dataset with IPC = 1 and 10.

Table 2 shows that our method outperforms KIP when using the same 128-width ConvNet. Even when KIP used a 1024-width ConvNet, IADD still achieved higher accuracy, except for CIFAR-10 with IPC = 1. We did not conduct experiments for KIP on CIFAR-100 with IPC = 50 because of the computational resource limitations of KIP; therefore, we only report our results in this study. The KIP method was specifically designed to function on large-width neural networks in the dataset



(a) CIFAR-100 (b) Tiny ImageNet

Figure 5: Distilled CIFAR-100 and Tiny ImageNet datasets (selected images) with IPC = 1.

distillation task. However, IADD can achieve suitable distillation performance with a normal-width network structure, which reduces both computing costs and time.

The visualization results of the distilled datasets are shown in Figs. 4 and 5, along with CIFAR-10 with IPC = 1 and 10, CIFAR-100 dataset with IPC = 1, and Tiny ImageNet (selected images) with IPC = 1. As illustrated in Fig. 4, the generated images appear more diverse when IPC = 10, given that the distillation process can compress the discriminative features of a class into multiple images. For instance, the generated images contain differently colored cars in the second row and various types of horses in the eighth row. Furthermore, when IPC = 1, the generated images are more abstract and densely packed with information, as the distillation process requires all class features to be compressed into a single image. In addition, when the resolution increases (from CIFAR-100 to Tiny ImageNet), as depicted in Fig. 5, the texture of the generated image is clearer and can show certain details, such as the beak of a penguin in the second image from the left in the third row and the ears of a cat in the third image from the left in the fourth row. The distilled images of the other SOTA methods can be easily found in their original papers.

We also conducted runtime complexity analysis of MTT, DDPP, and IADD. We measured the runtime results over 10 iterations, with each iteration consisting of 50 matching steps. These computations were performed on a single NVIDIA RTX A6000 GPU, with a batch size of 100. The results reveal that the distillation runtimes of MTT and DDPP average 11.5±0.5 and 12.5±0.5 s, respectively. In contrast, IADD exhibits a slightly higher distillation runtime, averaging 13.0±0.5 s. These findings indicate that the optimization of self-adaptive weights in our approach does not considerably increase the distillation runtime.

4.3. Analysis of Self-Adaptive Weights

In this section, we present two experiments to analyze the self-adaptive weights of IADD from different perspectives. The self-adaptive weights $\mathcal{W}$ are initialized with $\mathcal{W}_0$, a unit vector with corresponding dimensions. Subsequently, we proceeded with the distillation process and iteratively optimized the self-adaptive weights $\mathcal{W}$ via SGD until convergence was achieved.

In the first experiment, we demonstrated the analysis of the self-adaptive weights of CIFAR-10 with different IPC values. Figures 6 (a), (c), and (e) show the visualization results of the numerical difference between the parameters of the teacher and student networks in the corresponding dimensions with IPC = 1, IPC = 10, and IPC = 50, respectively. Figures 6 (b), (d), and (f) show the visualization results of the optimized self-adaptive weights in the corresponding dimensions with IPC = 1, IPC = 10, and IPC = 50, respectively. As illustrated in Fig. 6, across all settings, the adaptive weights of parameters with considerable differences gradually decrease during the training process. Conversely, the weights of parameters with moderate differences remain high. The experimental findings indicate that the proposed method for optimizing self-adaptive weights successfully issues the challenges of capturing variations in the complexity of parameter matching across various dimensions. This is because the self-adaptive weights can capture the subtle differences in the parameters of the teacher and student networks, and accordingly adjust the relative importance of various parameters to better align the datasets. In particular, they enable the dataset distillation process to make precise and context-dependent adjustments, thereby improving the distillation performance.

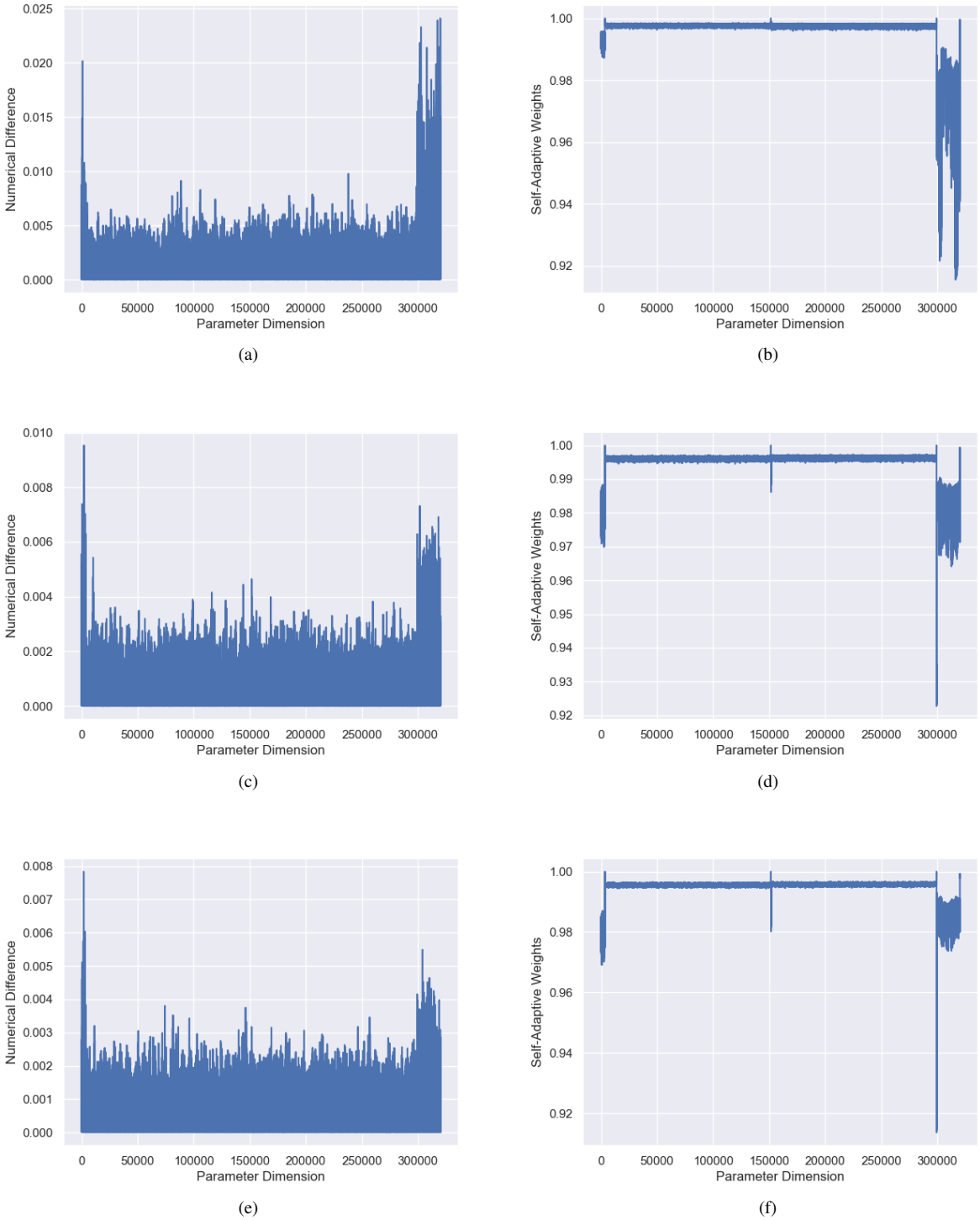


Figure 6: Analysis of self-adaptive weights of CIFAR-10 with various IPCs. (a), (c), and (e) present the visualization results of the numerical difference between the teacher and student network parameters in the corresponding dimensions with IPC = 1, IPC = 10, and IPC = 50, respectively. (b), (d), and (f) depict the visualization results of the optimized self-adaptive weights in the corresponding dimensions with IPC = 1, IPC = 10, and IPC = 50, respectively.

The teacher and student network parameters may exhibit non-uniformity across the corresponding dimensions, posing a challenge in matching these parameters. This non-uniformity is particularly prominent in the early and late layers, corresponding to input and output processing, respectively. These layers serve distinct functions, such as feature extraction and classification, and are thus subject to different sources of variation. Consequently, achieving an optimal balance of the param-

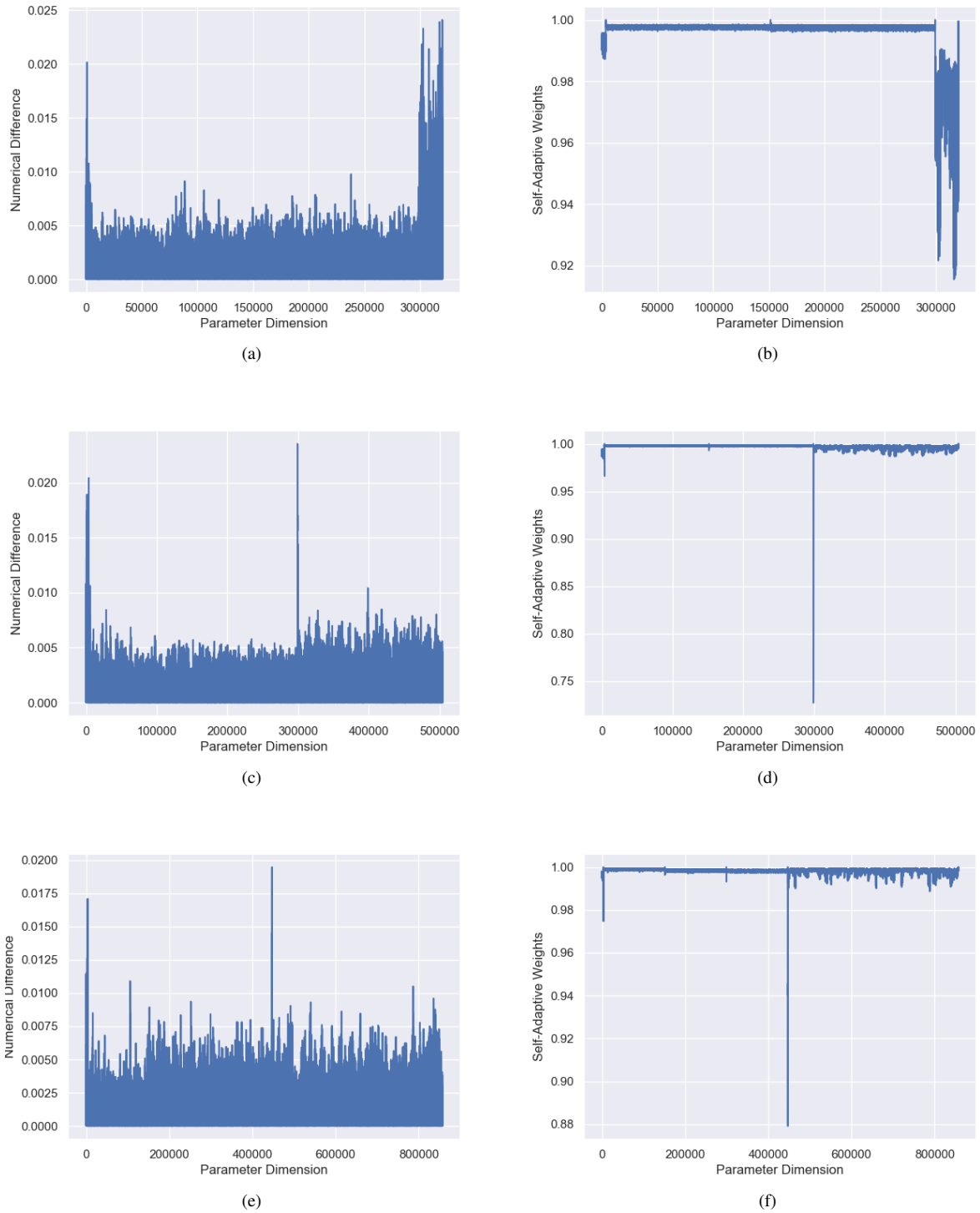


Figure 7: Analysis of self-adaptive weights of different datasets with IPC = 1. (a), (c), and (e) show the visualization results of the numerical difference between teacher and student network parameters in the corresponding dimensions on CIFAR-10, CIFAR-100, and Tiny ImageNet, respectively. (b), (d), and (f) show the visualization results of the optimized self-adaptive weights in the corresponding dimensions on CIFAR-10, CIFAR-100, and Tiny ImageNet, respectively.

eters in these layers is critical for realizing high distillation performance. The proposed method for optimizing self-adaptive weights successfully overcomes these challenges by enabling precise and context-dependent adjustment of network parameters, resulting in improved distillation performance.

In the second experiment, we present the analysis of the self-adaptive weights of different datasets with IPC = 1. Figures 7 (a), (c), and (e) show the visualization results of the numerical

difference between teacher and student network parameters in the corresponding dimensions on CIFAR-10, CIFAR-100, and Tiny ImageNet, respectively. Figures 7 (b), (d), and (f) show the visualization results of the optimized self-adaptive weights in the corresponding dimensions on CIFAR-10, CIFAR-100, and Tiny ImageNet, respectively. As illustrated in Fig. 7, across all settings, we observed that the parameters exhibiting considerable disparities tend to experience a gradual reduction in their weight values, whereas those with moderate differences maintain higher weights, which is consistent with the previous experiment.

In addition, we found that the optimization process of the self-adaptive weights experienced variations when using different datasets and training with networks of different depths. Different datasets have unique characteristics in terms of data distribution, feature representation, and noise levels. Consequently, these variations necessitate adaptive adjustments during the optimization process to effectively capture dataset-specific patterns. In addition, the depths of neural networks affect the optimization process. Networks with varying depths exhibit distinct levels of abstraction and feature hierarchies. Shallow networks tend to focus on capturing simpler features, whereas deep networks delve into intricate and abstract representations. These disparities influence the matching and weighing parameters in the optimization process. Deeper layers typically involve more abstract and task-specific information affecting the matching dynamics and consequently influencing the weight assignment.

4.4. Cross-Architecture Generalization

This section presents an evaluation of the cross-architecture generalization capability of the proposed method, where distilled images generated by ConvNetD3 on CIFAR-10 are used for testing on other architectures. The IPC was set to 10, and the same pretrained teacher networks used in Section 4.2 were used for rapid distillation and experimentation. The three main networks, namely AlexNet [6], VGG11 [89], and ResNet18 [7], were used to evaluate cross-architecture generalization with high accuracy.

Table 3 shows that the proposed method outperforms the SOTA methods MTT and DDPP for all architectures, indicating that our method can generate more robust distilled images. Notably, our method achieved an accuracy increase of 3.1% compared with DDPP for ResNet18. This improvement can be attributed to the automatic assignment of importance weights to various parameters during distillation, improving the contribution of important parameters and penalizing unimportant parameters. Consequently, our method effectively captures the important features of the data, resulting in improved performance of cross-architecture generalization.

4.5. Real-World Medical Application

In this section, the effectiveness of the proposed method is evaluated on a COVID-19 CXR dataset for real-world applications. A five-depth ConvNet (ConvNetD5) was used for distillation, which was necessary because of the considerable increase in image resolution compared with CIFAR-10 and CIFAR-100. Herein, 100 teacher networks were pretrained for the distillation process, with each teacher network trained for 50 epochs. The COVID-19 detection accuracy was tested with the IPC set to 20. For comparison, the COVID-19 detection accuracy of SOTA self-supervised learning methods was evaluated when 42 randomly selected images per class were chosen, accounting for 1% of the training set. For MTT and DDPP, we used the same training settings as those used in the proposed method.

Table 3: Test accuracy of cross-architecture generalization on CIFAR-10 dataset with IPC = 10.

Method	ConvNetD3	AlexNet	VGG11	ResNet18
DSA [47]	52.1±0.4	35.9±1.3	43.2±0.5	42.8±1.0
KIP [63]	47.6±0.9	24.4±3.9	42.1±0.4	36.8±1.0
MTT [51]	64.3±0.7	34.2±2.6	50.3±0.8	46.4±0.6
DDPP [54]	65.4±0.4	35.8±1.3	52.9±0.9	51.8±1.1
IADD	**66.2±0.6**	**36.5±1.1**	**53.4±1.3**	**54.9±0.7**

Table 4: COVID-19 detection accuracy of different methods. The first four comparison methods are baseline methods, the next six comparison methods are SOTA self-supervised learning methods, and the others are SOTA dataset distillation methods.

Method	Images	Structure	Accuracy
From Scratch	168	ResNet50 [7]	28.4%
From Scratch	168	ViT-Base [9]	41.3%
Transfer	168	ResNet50	53.9%
Transfer	168	ViT-Base	68.9%
SimSiam [15]	168	ResNet50	62.3%
BYOL [16]	168	ResNet50	68.3%
Cross [85]	168	ResNet50	74.7%
SKD [86]	168	ResNet50	74.2%
MAE [87]	168	ViT-Base	75.4%
RGMIM [88]	168	ViT-Base	77.1%
MTT [51]	80	ConvNetD5	82.7%
DDPP [54]	80	ConvNetD5	84.1%
IADD	80	ConvNetD5	**85.2%**
Original Dataset	16,933	ConvNetD5	88.9%

Table 4 shows the superior performance of the proposed method over MTT and DDPP in terms of COVID-19 detection accuracy, particularly when using a small number of distilled CXR images. The proposed method achieves high test accuracy even when the IPC value is low (20; corresponding to 80 distilled CXR images). Moreover, the proposed method considerably outperforms the SOTA self-supervised learning method, using a simpler network and fewer training images, indicating its effectiveness in real-world COVID-19 detection applications. In addition, the table presents the upper bound accuracy of 88.9% when training on the original dataset. Even with high-level compression, no considerable accuracy degradation is observed, which further demonstrates the effectiveness of the proposed method.

Examples of real and distilled CXR images are shown in Fig. 8. The distilled images exhibit visual differences com-

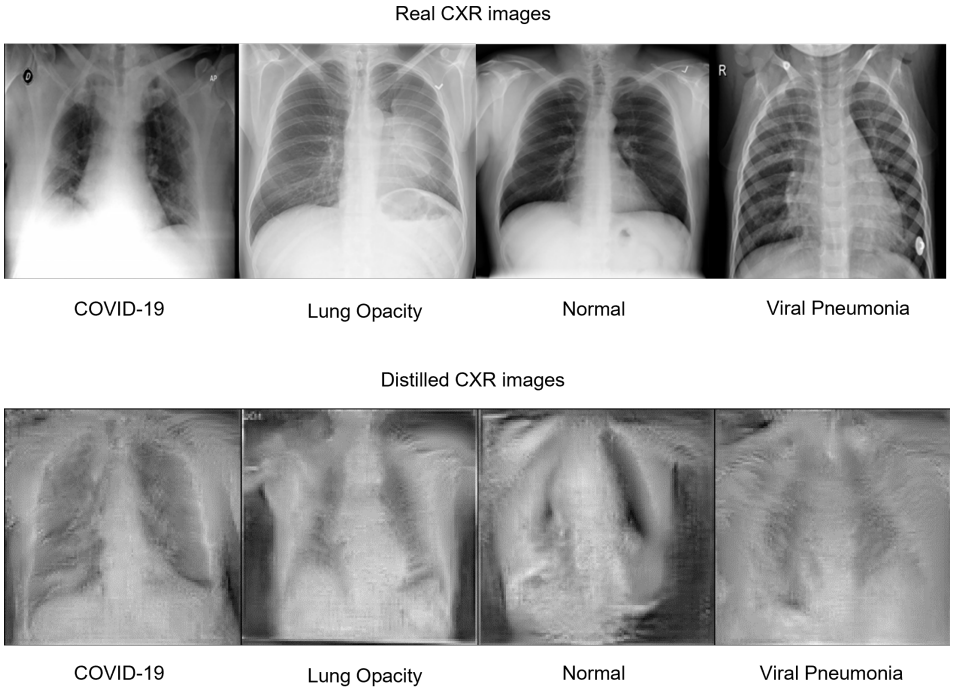


Figure 8: Examples of real and distilled CXR images of four classes (COVID-19, Lung Opacity, Normal, and Viral Pneumonia).

pared with the original CXR images. The sharing of medical datasets between hospitals is typically hindered by privacy-protection issues and the high cost of transmitting and storing numerous high-resolution medical images [90]. Dataset distillation can address this issue by synthesizing a small anonymous dataset such that models trained on it can achieve performance comparable to that when using the original large dataset. This potential demonstrates the capability of dataset distillation to solve existing medical sharing problems [91]. In addition, the combination of medical dataset distillation and federated learning [92, 93] may contribute to clinical usage.

5. Discussion

We have found that certain parameters can be difficult to match during the dataset distillation process, resulting in a negative impact on distillation performance, as observed in our previous work DDPP [54]. To solve this problem, DDPP uses parameter pruning to remove these parameters during the distillation process. However, directly removing difficult-to-match parameters is ineffective and can affect the number of network model parameters. Hence, DDPP is unsuitable for large-scale datasets because it tends to collapse during the distillation process, which is caused by the pruned parameters. In contrast to directly removing difficult-to-match parameters, IADD can improve distillation performance by automatically assigning importance weights to parameters during distillation. Furthermore, because IADD automatically optimizes the weights according to the importance of the parameters, we do not encounter the hyperparameter tuning problem in DDPP for defining difficult-to-match parameters. The promising results provide further evidence that supports the potential of the proposed method to contribute to the development of more efficient and effective dataset distillation algorithms.

For the other SOTA dataset distillation methods, DM [49] focuses on aligning dataset distributions to improve parameter alignment. Its efficacy depends on the specific distribution matching strategy employed. CAFE [48] emphasizes feature-level alignment through feature matching. Its performance is influenced by factors such as feature complexity and network architecture. MTT [51] concentrates on aligning the training process for dynamic knowledge transfer. This method necessitates meticulous tuning and careful consideration of difficult-to-match parameters. KIP [63] was specifically designed to operate on large-width neural networks in the dataset distillation task. In contrast, our method, IADD, achieves suitable distillation performance with a normal-width network structure. This distinction reduces the computing costs and considerably lowers the processing time.

Although our proposed method has shown promising results in parameter matching-based dataset distillation, it has several limitations. First, the proposed method is specifically designed for parameter matching-based algorithms, and extending it to other types of dataset distillation such as meta-learning or distribution matching remains to be researched further. Second, the proposed method may not apply to large-scale models such as vision transformers because of their high number of parameters, which can impact the optimization of self-adaptive weights. Moreover, although the proposed method has demonstrated its effectiveness in COVID-19 detection, its performance on other downstream tasks, such as continuous learning or neural network architecture search, is yet to be validated.

We also highlight some possible future research directions. Existing dataset distillation methods mainly rely on supervised learning, and the potential of unsupervised or self-supervised learning-based methods for dataset distillation needs to be investigated [94]. According to the results from the COVID-19 dataset, we do not observe an explicit effect of the imbalance on the distillation process, probably because the imbalance ratio is not very high. The dataset imbalance or long-tail problem is an interesting topic for dataset distillation, and it will be one of our future research directions [95].

6. Conclusion

In this study, we propose a novel dataset distillation method called IADD that improves distillation performance by assigning importance weights to different network parameters during distillation and generating more robust distilled datasets. Specifically, the proposed method improves the contribution of important parameters and penalizes unimportant parameters during the distillation process. Our experimental results show that IADD outperforms other SOTA distillation methods based on parameter matching on three benchmark datasets and exhibits improved cross-architecture generalization performance. Furthermore, we validate the effectiveness of our method on a real-world COVID-19 CXR dataset.

Our method has shown promise in parameter matching-based dataset distillation; however, it has some limitations. It is specifically tailored for parameter matching-based methods and may not easily extend to other dataset distillation methods such as meta-learning or distribution matching. Furthermore, its applicability to large-scale models such as vision transformers remains uncertain because of potential challenges in optimizing self-adaptive weights for such models.

Given that existing dataset distillation methods predominantly rely on supervised learning paradigms, there is a compelling need to delve into the uncharted territory of unsupervised or self-supervised learning-based methods for dataset distillation. As part of our ongoing research, we intend to delve deeper into the intricacies of dataset imbalance and its implications for dataset distillation, particularly in scenarios characterized by significant class imbalances.

Ethical Approval

No ethics approval is required.

Declaration of Competing Interest

None declared.

Acknowledgments

This study was partly supported by JSPS KAKENHI Grant Numbers JP21H03456 and JP23K11141, and AMED Grant Number JP23zf01270004h0003. This study was conducted at the Data Science Computing System of Education and Research Center for Mathematical and Data Science, Hokkaido University.

References

[1] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Networks 61 (2015) 85–117 (2015).
[2] M. Yutaka, L. Yann, S. Maneesh, P. Doina, S. David, S. Masashi, U. Eiji, M. Jun, Deep learning, reinforcement learning, and world models, Neural Networks 152 (2022) 267–275 (2022).
[3] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Communications of the ACM 60 (6) (2017) 84–90 (2017).
[4] T. Young, D. Hazarika, S. Poria, E. Cambria, Recent trends in deep learning based natural language processing, IEEE Computational Intelligence Magazine 13 (3) (2018) 55–75 (2018).
[5] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al., Deep speech 2: End-to-end speech recognition in english and mandarin, in: Proceedings of the International Conference on Machine Learning (ICML), 2016, pp. 173–182 (2016).
[6] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2012, pp. 1097–1105 (2012).
[7] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778 (2016).
[8] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), 2019 (2019).
[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, in: Proceedings of the International Conference on Learning Representations (ICLR), 2021 (2021).
[10] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: Proceedings of the International Conference on Machine Learning (ICML), 2021, pp. 8748–8763 (2021).
[11] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 10684–10695 (2022).
[12] O. AI, Gpt-4 technical report, arXiv preprint arXiv:2303.08774 (2023).
[13] H.-N. Dai, R. C.-W. Wong, H. Wang, Z. Zheng, A. V. Vasilakos, Big data analytics for large-scale wireless networks: Challenges and opportunities, ACM Computing Surveys 52 (5) (2019) 1–36 (2019).
[14] P. Ernst, S. Chatterjee, G. Rose, O. Speck, A. Nürnberger, Sinogram upsampling using primal-dual unet for undersampled ct and radial mri reconstruction, Neural Networks (2023).
[15] X. Chen, K. He, Exploring simple siamese representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 15750–15758 (2021).
[16] J.-B. Grill, F. Strub, F. Altché, Tallec, et al., Bootstrap your own latent-a new approach to self-supervised learning, in: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2020, pp. 21271–21284 (2020).
[17] G. Li, R. Togo, T. Ogawa, M. Haseyama, Tribyol: Triplet byol for self-supervised representation learning, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 3458–3462 (2022).
[18] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, Y. Bengio, An empirical investigation of catastrophic forgetting in gradient-based neural networks, arXiv preprint arXiv:1312.6211 (2013).
[19] O. Sener, S. Savarese, Active learning for convolutional neural networks: A core-set approach, arXiv preprint arXiv:1708.00489 (2017).
[20] B. Mirzasoleiman, J. Bilmes, J. Leskovec, Coresets for data-efficient training of machine learning models, in: Proceedings of the International Conference on Machine Learning (ICML), 2020, pp. 6950–6960 (2020).
[21] K. Killamsetty, D. Sivasubramanian, G. Ramakrishnan, R. Iyer, Glister: Generalization based data subset selection for efficient and robust learn-

ing, in: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2021, pp. 8110–8118 (2021).
[22] K. Killamsetty, S. Durga, G. Ramakrishnan, A. De, R. Iyer, Grad-match: Gradient matching based data subset selection for efficient deep model training, in: Proceedings of the International Conference on Machine Learning (ICML), 2021, pp. 5464–5474 (2021).
[23] T. Panch, P. Szolovits, R. Atun, Artificial intelligence, machine learning and health systems, Journal of Global Health 8 (2) (2018).
[24] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, et al., The future of digital health with federated learning, NPJ Digital Medicine 3 (1) (2020) 119 (2020).
[25] T. Wang, J.-Y. Zhu, A. Torralba, A. A. Efros, Dataset distillation, arXiv preprint arXiv:1811.10959 (2018).
[26] F. Wiewel, B. Yang, Condensed composite memory continual learning, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), 2021, pp. 1–8 (2021).
[27] M. Sangermano, A. Carta, A. Cossu, D. Bacciu, Sample condensation in online continual learning, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), 2022, pp. 1–8 (2022).
[28] T. Dong, B. Zhao, L. Liu, Privacy for free: How does dataset condensation help privacy?, in: Proceedings of the International Conference on Machine Learning (ICML), 2022, pp. 5378–5396 (2022).
[29] D. Chen, R. Kerkouche, M. Fritz, Private set generation with discriminative information, in: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2022 (2022).
[30] Y. Wu, X. Li, F. Kerschbaum, H. Huang, H. Zhang, Towards robust dataset learning, arXiv preprint arXiv:2211.10752 (2022).
[31] Y. Liu, Z. Li, M. Backes, Y. Shen, Y. Zhang, Backdoor attacks against dataset distillation, in: Proceedings of the Network and Distributed System Security Symposium (NDSS), 2023 (2023).
[32] G. Li, R. Togo, T. Ogawa, M. Haseyama, Soft-label anonymous gastric x-ray image distillation, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2020, pp. 305–309 (2020).
[33] G. Li, R. Togo, T. Ogawa, M. Haseyama, Compressed gastric image generation based on soft-label dataset distillation for medical data sharing, Computer Methods and Programs in Biomedicine (2022).
[34] G. Li, R. Togo, T. Ogawa, M. Haseyama, Dataset distillation for medical dataset sharing, in: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), Workshop, 2023, pp. 1–6 (2023).
[35] R. Song, D. Liu, D. Z. Chen, A. Festag, C. Trinitis, M. Schulz, A. Knoll, Federated learning via decentralized dataset distillation in resource-constrained edge environments, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), 2023, pp. 1–10 (2023).
[36] Y. Xiong, R. Wang, M. Cheng, F. Yu, C.-J. Hsieh, FedDM: Iterative distribution matching for communication-efficient federated learning, in: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Workshop, 2022 (2022).
[37] P. Liu, X. Yu, J. T. Zhou, Meta knowledge condensation for federated learning, in: Proceedings of the International Conference on Learning Representations (ICLR), 2023 (2023).
[38] W. Jin, L. Zhao, S. Zhang, Y. Liu, J. Tang, N. Shah, Graph condensation for graph neural networks, in: Proceedings of the International Conference on Learning Representations (ICLR), 2022 (2022).
[39] W. Jin, X. Tang, H. Jiang, Z. Li, D. Zhang, J. Tang, B. Ying, Condensing graphs via one-step gradient matching, in: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2022 (2022).
[40] F. P. Such, A. Rawal, J. Lehman, K. Stanley, J. Clune, Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data, in: Proceedings of the International Conference on Machine Learning (ICML), 2020, pp. 9206–9216 (2020).
[41] B. Zhao, H. Bilen, Dataset condensation with gradient matching, in: Proceedings of the International Conference on Learning Representations (ICLR), 2021 (2021).
[42] C. Chen, Y. Zhang, J. Fu, X. Liu, M. Coates, Bidirectional learning for offline infinite-width model-based optimization, in: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2022 (2022).
[43] C. Chen, Y. Zhang, X. Liu, M. Coates, Bidirectional learning for offline model-based biological sequence design, in: Proceedings of the International Conference on Machine Learning (ICML), 2023, pp. 5351–5366 (2023).
[44] N. Sachdeva, J. McAuley, Data distillation: A survey, Transactions on Machine Learning Research (2023).
[45] S. Lei, D. Tao, A comprehensive survey to dataset distillation, IEEE Transactions on Pattern Analysis and Machine Intelligence (2023).
[46] R. Yu, S. Liu, X. Wang, A comprehensive survey to dataset distillation, IEEE Transactions on Pattern Analysis and Machine Intelligence (2023).
[47] B. Zhao, H. Bilen, Dataset condensation with differentiable siamese augmentation, in: Proceedings of the International Conference on Machine Learning (ICML), 2021, pp. 12674–12685 (2021).
[48] K. Wang, B. Zhao, X. Peng, Z. Zhu, S. Yang, S. Wang, G. Huang, H. Bilen, X. Wang, Y. You, CAFE: Learning to condense dataset by aligning features, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 12196–12205 (2022).
[49] B. Zhao, H. Bilen, Dataset condensation with distribution matching, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2023 (2023).
[50] S. Lee, S. Chun, S. Jung, S. Yun, S. Yoon, Dataset condensation with contrastive signals, in: Proceedings of the International Conference on Machine Learning (ICML), 2022, pp. 12352–12364 (2022).
[51] G. Cazenavette, T. Wang, A. Torralba, A. A. Efros, J.-Y. Zhu, Dataset distillation by matching training trajectories, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 4750–4759 (2022).
[52] J. Du, Y. Jiang, V. T. F. Tan, J. T. Zhou, H. Li, Minimizing the accumulated trajectory error to improve dataset distillation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023 (2023).
[53] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).
[54] G. Li, R. Togo, T. Ogawa, M. Haseyama, Dataset distillation using parameter pruning, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences (2023).
[55] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International conference on machine learning, PMLR, 2017, pp. 1126–1135 (2017).
[56] O. Bohdal, Y. Yang, T. Hospedales, Flexible dataset distillation: Learn labels instead of images, in: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Workshop, 2020 (2020).
[57] I. Sucholutsky, M. Schonlau, Soft-label dataset distillation and text dataset distillation, in: International Joint Conference on Neural Networks (IJCNN), 2021, pp. 1–8 (2021).
[58] Z. Deng, O. Russakovsky, Remember the past: Distilling datasets into addressable memories for neural networks, in: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2022 (2022).
[59] D. Maclaurin, D. Duvenaud, R. Adams, Gradient-based hyperparameter optimization through reversible learning, in: Proceedings of the International Conference on Machine Learning (ICML), 2015, pp. 2113–2122 (2015).
[60] J. Lorraine, P. Vicol, D. Duvenaud, Optimizing millions of hyperparameters by implicit differentiation, in: Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), 2020, pp. 1540–1552 (2020).
[61] P. Vicol, J. P. Lorraine, F. Pedregosa, D. Duvenaud, R. B. Grosse, On implicit bias in overparameterized bilevel optimization, in: Proceedings of the International Conference on Machine Learning (ICML), 2022, pp. 22234–22259 (2022).
[62] T. Nguyen, Z. Chen, J. Lee, Dataset meta-learning from kernel ridge-regression, in: Proceedings of the International Conference on Learning Representations (ICLR), 2021 (2021).
[63] T. Nguyen, R. Novak, L. Xiao, J. Lee, Dataset distillation with infinitely wide convolutional networks, in: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2021, pp. 5186–5198 (2021).
[64] Y. Zhou, E. Nezhadarya, J. Ba, Dataset distillation using neural feature regression, in: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2022 (2022).
[65] N. Loo, R. Hasani, A. Amini, D. Rus, Efficient dataset distillation using random feature approximation, in: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2022 (2022).
[66] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, A. Smola, A

kernel two-sample test, The Journal of Machine Learning Research 13 (1) (2012) 723–773 (2012).
[67] G. Li, B. Zhao, T. Wang, Awesome dataset distillation, https://github.com/Guang000/Awesome-Dataset-Distillation (2022).
[68] Y. Tian, J. Wang, J. Yueming, L. Wang, Communication-efficient federated skin lesion classification with generalizable dataset distillation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Workshop, 2023, pp. 1–10 (2023).
[69] G. Wang, J. C. Ye, B. De Man, Deep learning for tomographic image reconstruction, Nature Machine Intelligence 2 (12) (2020) 737–748 (2020).
[70] J. Willard, X. Jia, S. Xu, M. Steinbach, V. Kumar, Integrating scientific knowledge with machine learning for engineering and environmental systems, ACM Computing Surveys 55 (4) (2022).
[71] V. Gavrishchaka, O. Senyukova, M. Koepke, Synergy of physics-based reasoning and machine learning in biomedical applications: towards unlimited deep learning with limited data, Advances in Physics: X 4 (1) (2019) 1582361 (2019).
[72] Q. Ke, J. Zhang, W. Wei, D. Połap, M. Woźniak, L. Kośmider, R. Damaševĭcius, A neuro-heuristic approach for recognition of lung diseases from x-ray images, Expert Systems with Applications 126 (2019) 218–232 (2019).
[73] R. Rajagopal, R. Karthick, P. Meenalochini, T. Kalaichelvi, Deep convolutional spiking neural network optimized with arithmetic optimization algorithm for lung disease detection using chest x-ray images, Biomedical Signal Processing and Control 79 (2023) 104197 (2023).
[74] Y. Si, J. Wang, H. Xu, K. Roberts, Enhancing clinical concept extraction with contextual embeddings, Journal of the American Medical Informatics Association 26 (11) (2019) 1297–1304 (2019).
[75] F. Wang, A. Preininger, Ai in health: state of the art, challenges, and future directions, Yearbook of Medical Informatics 28 (01) (2019) 016–026 (2019).
[76] G. Yang, Q. Ye, J. Xia, Unbox the black-box for the medical explainable ai via multi-modal and multi-centre data fusion: A mini-review, two showcases and beyond, Information Fusion 77 (2022) 29–52 (2022).
[77] M. Subramanian, A. Wojtusciszyn, L. Favre, S. Boughorbel, J. Shan, K. B. Letaief, N. Pitteloud, L. Chouchane, Precision medicine in the era of artificial intelligence: implications in chronic disease management, Journal of Translational Medicine 18 (1) (2020) 1–12 (2020).
[78] K. A. Tran, O. Kondrashova, A. Bradley, E. D. Williams, J. V. Pearson, N. Waddell, Deep learning in cancer diagnosis, prognosis and treatment selection, Genome Medicine 13 (1) (2021) 1–17 (2021).
[79] Z. Jiang, J. Gu, M. Liu, D. Z. Pan, Delving into effective gradient matching for dataset condensation, arXiv preprint arXiv:2208.00311 (2022).
[80] Y. Le, X. Yang, Tiny imagenet visual recognition challenge, CS 231N 7 (7) (2015) 3 (2015).
[81] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. A. Kashem, M. T. Islam, S. Al Maadeed, S. M. Zughaier, M. S. Khan, et al., Exploring the effect of image enhancement techniques on covid-19 detection using chest x-ray images, Computers in Biology and Medicine 132 (2021) 104319 (2021).
[82] M. Toneva, A. Sordoni, R. T. d. Combes, A. Trischler, Y. Bengio, G. J. Gordon, An empirical study of example forgetting during deep neural network learning, in: Proceedings of the International Conference on Learning Representations (ICLR), 2019 (2019).
[83] Y. Chen, M. Welling, A. Smola, Super-samples from kernel herding, in: Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), 2010 (2010).
[84] S. Gidaris, N. Komodakis, Dynamic few-shot visual learning without forgetting, in: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4367–4375 (2018).
[85] G. Li, R. Togo, T. Ogawa, M. Haseyama, Covid-19 detection based on self-supervised transfer learning using chest x-ray images, International Journal of Computer Assisted Radiology and Surgery (2022) 715–722 (2022).
[86] G. Li, R. Togo, T. Ogawa, M. Haseyama, Self-knowledge distillation based self-supervised learning for covid-19 detection from chest x-ray images, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 1371–1375 (2022).
[87] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 16000–16009 (2022).
[88] G. Li, R. Togo, T. Ogawa, M. Haseyama, Rgmim: Region-guided masked image modeling for covid-19 detection, arXiv preprint arXiv:2211.00313 (2022).
[89] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Proceedings of the International Conference on Learning Representations (ICLR) (2015).
[90] Y. Ye, J. Shi, D. Zhu, L. Su, Y. Huang, J. Huang, Management of medical and health big data based on integrated learning-based health care system: A review and comparative analysis, Computer Methods and Programs in Biomedicine (2021) 106293 (2021).
[91] F. K. Dankar, R. Badji, A risk-based framework for biomedical data sharing, Journal of Biomedical Informatics 66 (2017) 231–240 (2017).
[92] R. Song, L. Zhou, L. Lyu, A. Festag, A. Knoll, Resfed: Communication efficient federated learning with deep compressed residuals, IEEE Internet of Things Journal (2023).
[93] R. Song, R. Xu, A. Festag, J. Ma, A. Knoll, Fedbevt: Federated learning bird's eye view perception transformer in road traffic systems, IEEE Transactions on Intelligent Vehicles (2023).
[94] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, J. Tang, Self-supervised learning: Generative or contrastive, IEEE Transactions on Knowledge and Data Engineering 35 (1) (2021) 857–876 (2021).
[95] Y. Zhang, B. Kang, B. Hooi, S. Yan, J. Feng, Deep long-tailed learning: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence (2023).