Exploiting Inter-sample and Inter-feature Relations in Dataset Distillation

Wenxiao Deng[1], Wenbin Li[1]*, Tianyu Ding[2], Lei Wang[3],
Hongguang Zhang[4], Kuihua Huang[5], Jing Huo[1], Yang Gao[1]

[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]Microsoft Corporation, USA   [3]University of Wollongong, Australia
[4]Systems Engineering Institute, AMS, China   [5]School of Systems Engineering, University of Defense Technology, China

Abstract

Dataset distillation has emerged as a promising approach in deep learning, enabling efficient training with small synthetic datasets derived from larger real ones. Particularly, distribution matching-based distillation methods attract attention thanks to its effectiveness and low computational cost. However, these methods face two primary limitations: the dispersed feature distribution within the same class in synthetic datasets, reducing class discrimination, and an exclusive focus on mean feature consistency, lacking precision and comprehensiveness. To address these challenges, we introduce two novel constraints: a class centralization constraint and a covariance matching constraint. The class centralization constraint aims to enhance class discrimination by more closely clustering samples within classes. The covariance matching constraint seeks to achieve more accurate feature distribution matching between real and synthetic datasets through local feature covariance matrices, particularly beneficial when sample sizes are much smaller than the number of features. Experiments demonstrate notable improvements with these constraints, yielding performance boosts of up to 6.6% on CIFAR10, 2.9% on SVHN, 2.5% on CIFAR100, and 2.5% on TinyImageNet, compared to the state-of-the-art relevant methods. In addition, our method maintains robust performance in cross-architecture settings, with a maximum performance drop of 1.7% on four architectures. Code is available at https://github.com/VincenDen/IID.

1. Introduction

Deep learning has evolved rapidly in recent years, achieving remarkable results across various fields [10, 15, 20, 23, 31, 35, 36]. Dataset distillation [33], a process of distilling knowledge from a large real dataset to a much smaller synthetic dataset, has emerged as a key technique for efficient deep learning training. It has also been widely adopted in
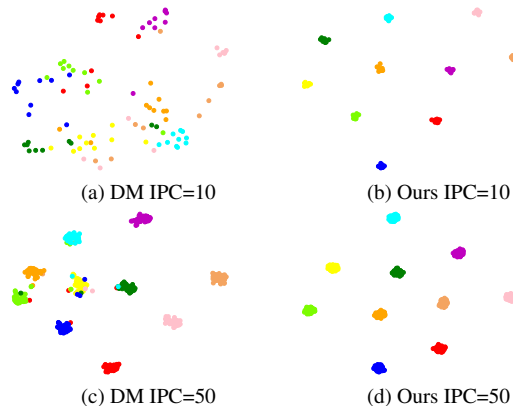
*Corresponding author

Figure 1. T-SNE visualisation of features from synthetic dataset obtained by DM [39] and our method, using a pre-trained Resnet18 on CIFAR10. Different colors represent different classes. IPC denotes the number of images per class.

areas like neural architecture search [6, 11, 37, 39, 40], continual learning [11, 24, 26], privacy protection [3, 9, 42].

Dataset distillation, also known as dataset condensation, initially relied on coreset selection methods [12, 27, 34], which involve selecting representative samples from a dataset. However, these methods have limitations in performance and scalability, especially for large datasets. Pioneering work by Wang *et al.* [33] introduces a meta-learning based approach for dataset distillation, which uses backpropagation to optimize images directly. Presently, dataset distillation methods are primarily categorized into three types: gradient matching [37, 40], trajectory matching [5], and distribution matching [39, 41]. The first two types, despite their effectiveness, are computationally expensive due to their reliance on second-order gradient optimization. In contrast, distribution matching methods [39] like those introduced by Zhao *et al.* [41], CAFE [32], and DataDAM [25] address this challenge by matching feature distributions in the embedding space, thereby reducing computational costs. More details about distribution matching method are provided in Section 3.1.

In this paper, we focus on distribution matching (DM) methods due to their impressive performance and computational efficiency. The idea of the distribution matching method is to make the synthetic dataset have the same feature distribution as the real dataset with the same class of samples, so how to carve and match this distribution is the key to improve the performance. We identify two key limitations of the methods and design two constraints to improve them. First, the feature distribution of samples within the same class in synthetic datasets could still be excessively scattered, leading to poor class discrimination in the embedding space. This issue becomes more pronounced with a smaller number of images per class (IPC), as shown in Figs. 1a and 1c. To counter this, we propose a *class centralization constraint*, designed to promote clustering of class-specific samples, demonstrated in Figs. 1b and 1d. Second, existing methods inadequately match feature distributions, focusing only on the mean features between real and synthetic datasets. We argue that a comprehensive feature distribution description should include not only means but also covariance matrices, the latter characterises inter-feature relationships. However, in synthetic datasets the number of samples is typically smaller than feature dimensions, accurately estimating covariance matrices is challenging. Our solution is a *covariance matching constraint* that can achieve more precise feature distribution matching between real and synthetic datasets through local feature covariance matrices, even with small sample sizes.

Importantly, both constraints can be seamlessly integrated with existing methods. We have applied them to DM [39] and IDM [39], two representative distribution matching-based methods, and carried out experiments on SVHN, CIFAR10, CIFAR100, and Tiny-Imagenet datasets. Our experiments demonstrate a performance gain of up to 6.6% on CIFAR10. In cross-architecture experiments, synthetic dataset obtained by the proposed method is tested on network architectures different from the distillation phase, our method exhibits a minimal performance reduction, with a maximum of only 1.7% across four architectures.

Our main contributions of this work are as follows:

• We propose a simple yet effective *class centralization constraint*, significantly enhancing class discrimination in synthetic dataset, and improving the performance of dataset distillation.
• We develop a novel *covariance matching constraint*, which facilitates more accurate feature distribution matching between real and synthetic datasets even with limited sample sizes, thus effectively augmenting the general mean distribution matching approach.
• We evaluate the proposed two constraints across multiple benchmark datasets, achieving substantial performance improvements over recent baseline methods and surpassing current state-of-the-art techniques.

2. Related Works

Coreset selection. Coreset selection, an early approach to dataset distillation, involves selecting representative samples from a dataset and is applied in various contexts [1, 4, 22]. Simple methods include random selection, while more sophisticated techniques like Herding [34] focuses on class centers, and K-Center [12, 27] selects multiple centroids. The forgetting [30] identifies samples that are easily forgotten during training to select representative ones. However, coreset selection struggles with scalability on large datasets and often exhibits suboptimal performance.

Dataset distillation. Dataset distillation or condensation aims to condense a large dataset into much smaller yet informative one. It finds applications in neural architecture search [6, 11, 37, 39, 40], continual learning [11, 24, 26], and privacy protection [3, 9, 42], *etc*. The concept was introduced by Wang *et al*. [33] using a meta-learning approach. Nguyen *et al*. [21] further optimize this method using NTK-based ridge regression for better cross-architecture generalization. Zhao *et al*. [37] further improve the distillation performance using an insertable differentiable siamese augmentation. After that, Bohdal *et al*. [13] demonstrate that labels could also be distilled. Gradient matching methods proposed by Zhao *et al*. [40] and trajectory matching methods by Cazenavette *et al*. [5] aim to match gradients and model parameters of synthetic and real datasets, respectively. Recent advances [8, 18, 38] focus on preserving features and corresponding decoders, then restore the features to the original size of the image. This will enhance performance while also introducing additional preprocessing steps for synthetic datasets.

To reduce the high computational cost of the previous methods, Zhao *et al*. [39] propose distribution matching (DM), optimizing for maximum mean discrepancy (MMD) between synthetic and real datasets. CAFE [32] aligns features by matching them across layers in convolutional neural networks, and IDM [41] improves DM with partitioning augmentation and class-aware distribution regularization. DataDAM [25] proposes an attention matching framework to dataset distillation by focusing on feature attention.

Although demonstrating promising performance, existing distribution matching methods methods face two primary limitations: the dispersed feature distribution within the same class in synthetic datasets, reducing class discrimination, and an exclusive focus on mean feature consistency, lacking precision and comprehensiveness. This work addresses above limitations by proposing two plug-and-play constraints, which significantly enhance dataset distillation performance by focusing on inter-sample and inter-feature relations, respectively.

3. Method

In this section, we outline the fundamentals of dataset distillation and basic framework of distribution matching-based methods. We then delve into specifics of our proposed *class centralization constraint* and *covariance matching constraint*. Lastly, we present the entire objective function.

3.1. Preliminaries

Given a large image dataset $\mathcal{T} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_{|\mathcal{T}|}, y_{|\mathcal{T}|})\}$, containing $|\mathcal{T}|$ images and labels, dataset distillation aims to condense $\mathcal{T}$ to a significantly smaller synthetic dataset $\mathcal{S} = \{(s_1, y_1^s), (s_2, y_2^s), \ldots, (s_{|\mathcal{S}|}, y_{|\mathcal{S}|}^s)\}$ comprising $|\mathcal{S}|$ images and class labels. The goal is to ensure that a model $\psi_{\theta^{\mathcal{S}}}$, trained from scratch with $\mathcal{S}$, performs comparably to a model $\psi_{\theta^{\mathcal{T}}}$ trained with $\mathcal{T}$. This is achieved by optimizing the following objective function:

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} \mathbb{E}_{\boldsymbol{x} \sim P_{\mathcal{T}}} \|\ell(\psi_{\theta^{\mathcal{T}}}(\boldsymbol{x}), y) - \ell(\psi_{\theta^{\mathcal{S}}}(\boldsymbol{x}), y)\|, \quad (1)$$

where ℓ represents the cross-entropy loss since all existing dataset distillation methods focus on classification tasks.

DM [39] is a representative distribution matching-based dataset distillation method, aiming to minimize the maximum mean discrepancy (MMD) between the synthetic dataset and real dataset:

$$\mathbb{E}_{\boldsymbol{\theta} \sim P_{\boldsymbol{\theta}}} \left\| \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \psi_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - \frac{1}{|\mathcal{S}|} \sum_{j=1}^{|\mathcal{S}|} \psi_{\boldsymbol{\theta}}(\boldsymbol{s}_j) \right\|^2, \quad (2)$$

where $P_{\boldsymbol{\theta}}$ denotes the distribution of randomly initialized network parameters.

3.2. Class centralization constraint (inter-sample)

Synthetic datasets obtained by methods like DM [39] often exhibit insufficient class discrimination. This issue, particularly evident when the number of samples per class (IPC) is small, is illustrated by the scattered feature distribution and unclear class boundaries in synthetic datasets (Figs. 1a and 1c). To address this, we propose a class centralization constraint, which aims to cluster features $\phi(s)$ extracted from the synthetic dataset within the same class, rather than allowing them to disperse. We formulate the loss to enforce this constraint as:

$$\mathcal{L}_{CC} = \sum_{c}^{C} \left(\sum_{j=1}^{K} \max(0, \exp(\alpha \|\psi(s_j^c) - \psi(\bar{s}^c)\|^2) - \beta) \right), \quad (3)$$

where

$$\psi(\bar{s}^c) = \frac{1}{K} \sum_{j=1}^{K} \psi(s_j^c), \quad (4)$$

C is the number of categories, K is the number of samples per class, α is a scaling factor, and β is the centralization threshold. A smaller β encourages tighter clustering of samples within each class.
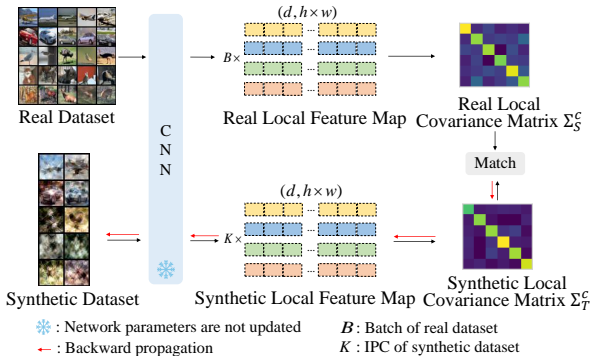

Figure 2. Illustration of the proposed covariance matching constraint. This constraint involves calculating local covariance matrices for corresponding classes in both real and synthetic datasets, followed by matching these matrices.

Since this is a plug-and-play constraint, we keep the original constraint Eq. (2) from the baseline, but it's worth noting that the previous methods used ConvNet ψ in Eq. (2) with randomly initialized parameters (DM [39]) or pretrained parameters (IDM [41]), our approach employs Resnet18 for network ψ in Eq. (3), which is distinct from network ψ in Eq. (2). Despite ResNet's more complex architecture, which typically poses challenges for small datasets, our experiments reveal that Resnet18 more effectively differentiates between class features. Additionally, using different neural networks enhances the cross-architecture generalization of dataset distillation. As illustrated in Figs. 1b and 1d, our class centralization constraint results in a more concentrated feature distribution within each class in the synthetic dataset, achieving clear differentiation between classes. The effectiveness of this constraint is further validated in our experimental results.

3.3. Covariance matching constraint (inter-feature)

The essence of distribution matching-based dataset distillation lies in aligning the feature distributions of real and synthetic datasets. While existing methods primarily focus on matching feature means, an effective representation of feature distribution shall also consider the covariance matrix, which captures inter-feature relationships. However, in distilled synthetic datasets, the number of samples in each class is often much smaller than the feature dimensions. This is often known as "small sample problem" and it could lead to less precise covariance matrix estimations.

To address this, we propose the covariance matching constraint for more accurate feature distribution matching, even when the sample size is smaller than the dimensions of features extracted by network ψ. As shown in Fig. 2, for the feature of a single sample, rather than flattening, we reshape it into a tensor of shape (d, hw), resulting in

d hw-dimensional local feature descriptors, expressed as $X_i \in \mathbb{R}^{d \times hw}$ for real dataset, $S_i \in \mathbb{R}^{d \times hw}$ for synthetic dataset. In this way, we greatly reduce the feature dimensions and avoid high-dimensional vector space computation, which facilitates a more accurate evaluation of the covariance matrices. We then calculate the local feature covariance matrices $\Sigma_\tau \in \mathbb{R}^{d \times d}$ and $\Sigma_s \in \mathbb{R}^{d \times d}$ for the real and synthetic datasets, respectively, and compute a matching loss between these two matrices:

$$\mathcal{L}_{CM} = \sum_{c=1}^{C} \|\Sigma_s^c - \Sigma_\tau^c\|^2 , \tag{5}$$

where

$$\Sigma_s^c = \frac{1}{K} \sum_{i=1}^{K} (S_i^c - \bar{S}^c)(S_i^c - \bar{S}^c)^\top , \tag{6}$$

$$\Sigma_\tau^c = \frac{1}{B} \sum_{i=1}^{B} (X_i^c - \bar{X}^c)(X_i^c - \bar{X}^c)^\top . \tag{7}$$

Here, K is the number of samples per class in the synthetic dataset $\mathcal{S}$, B is the batch size per class of the real dataset $\mathcal{T}$, $\bar{S}^c \in \mathbb{R}^{d \times hw}$ and $\bar{X}^c \in \mathbb{R}^{d \times hw}$ are the feature means of class c for the synthetic and real datasets, respectively.

3.4. The objective function

Our proposed constraints are designed as plug-and-play, making them adaptable to various distribution matching-based methods. For instance, when employing DM [39] as the baseline method, the overall objective function can be expressed as:

$$\mathcal{L} = \mathcal{L}_{DM} + \lambda_{CC}\mathcal{L}_{CC} + \lambda_{CM}\mathcal{L}_{CM} . \tag{8}$$

Similarly, when using IDM [41] as the baseline, the objective function will be defined as:

$$\mathcal{L} = \mathcal{L}_{IDM} + \lambda_{CC}\mathcal{L}_{CC} + \lambda_{CM}\mathcal{L}_{CM} . \tag{9}$$

In above formulas, λ_{CC} and λ_{CM} are the weighting parameters, and $\mathcal{L}_{DM}$ and $\mathcal{L}_{IDM}$ represent the loss functions in DM [39] and IDM [41], respectively.

4. Experiments

4.1. Experimental setup

Datasets. We evaluate our method on several benchmark datasets for dataset distillation: SVHN [28], CIFAR10 and CIFAR100 [16], as well as the larger TinyImageNet [17]. SVHN contains over 600,000 digit images of house numbers around the world. CIFAR10 and CIFAR100 feature 10 and 100 classes respectively, each with 600 images per class, and an image resolution of 32×32. For larger datasets, we performed on TinyImageNet [17], which contains 200 classes with 600 images in each class, which have an image resolution of 64×64.

Network architectures. For $\mathcal{L}_{DM}$, $\mathcal{L}_{IDM}$ in Eq. (8) and Eq. (9), we utilize the ConvNet with the same architecture as in DM [39] and IDM [41] to extract features. This architecture comprises 3 blocks, each with a 3×3 convolutional layer, instance normalization, ReLU activation, and a 3×3 average pooling with a stride of 2. For our class centralization constraint $\mathcal{L}_{CC}$, we employ a pretrained ResNet18 [14], trained for 30 epochs on each corresponding dataset. The architecture of ResNet18 includes 4 layers with 2 blocks per layer, and convolutional layer, instance normalization, and ReLU activation. ConvNet, AlexNet, VGG11, and ResNet18 are used in the cross-architecture generalization experiments, and we follow the architectural setup of DM [39].

Evaluation. We adhere to the evaluation protocol of previous works [18, 33, 37, 39, 40]. On various datasets, we synthesize distilled datasets with IPC = $1, 10, 50$, then train a randomly initialized model from scratch using these datasets. The training configurations follow the prior works [18, 33, 37, 39, 40]. We compute the Top-1 accuracy on the test set of the original real dataset, repeating each experiment five times to calculate the mean value.

Implementation details. Our method is implemented based on DM [41] and IDM [41], adhering to their baseline hyperparameter settings. We follow the DSA [37] augmentation method used in previous works. Synthetic datasets are learned using SGD with a learning rate of 1, and a batch size of 5000 for IDM experiments with IPC=1 on CIFAR10/100, and 256 in other scenarios. We set λ_{CC} as 0.05, and λ_{CM} as 0.01 for IPC = 1, and 10, while for IPC = 50, λ_{CC} is set as 0.003 and λ_{CM} as 0.01. All experiments are conducted using the PyTorch framework on two RTX3090 GPUs, each with 24 GB of memory.

4.2. Comparison with state-of-the-art methods

Competitive methods. We compare our method with both classic coreset selection methods, including Random, Herding [34], K-Center [27], and Forgetting [30], and state-of-the-art dataset distillation techniques including DD [33], LD [2], DC [40], DSA [37], KIP [21], CAFE [32], DCC [19], DM [39], IDM [41], DataDAM [25]. For DM [39] and IDM [41], we report results reproduced using their official codes, while results for other methods are quoted from their respective original papers.

Performance comparison. Tabs. 1 and 2 show the results of our method in comparison with previous methods across various datasets. The experimental results demonstrate significant improvements over the baselines. Specifically, our method surpasses DM by 2.9% and 2.7% on SVHN at IPC=10 and 50, respectively; 6.6% and 2.9% on CIFAR10 at IPC=10 and 50; and 2.5% and 2.0% on TinyImageNet at IPC=10 and 50. When compared with IDM, our method shows an improvement of 1.9%, 2.6%

Table 1. Comparative analysis of dataset distillation methods. Ratio (%): the proportion of condensed images relative to the number of entire training set. Whole Dataset: the accuracy of training on the entire original dataset. The **best** results in each column are highlighted for clarity.

Method	Venue	SVHN			CIFAR10			CIFAR100		
IPC		1	10	50	1	10	50	1	10	50
Ratio (%)		0.0014	0.14	0.7	0.02	0.2	1	0.2	2	10
Random	Classic	14.6±1.6	35.1±4.1	70.9±0.9	14.4±2.0	26.0±1.2	43.4±1.0	4.2±0.3	14.6±0.5	30.0±0.4
Herding [34]	Classic	20.9±1.3	50.5±3.3	72.6±0.8	21.5±1.2	31.6±0.7	40.4±0.6	8.4±0.3	17.3±0.3	33.7±0.5
K-Center [27]	Classic	21.0±1.5	14.0±1.3	20.1±1.4	21.5±1.3	14.7±0.9	27.0±1.4	8.4±0.3	17.3±0.2	30.5±0.3
Forgetting [30]	Classic	12.1±5.6	16.8±1.2	27.2±1.5	13.5±1.2	23.3±1.0	23.3±1.1	4.5±0.3	15.1±0.2	30.5±0.4
DD [33]	Arxiv'18	-	-	-	-	36.8±1.2	-	-	-	-
LD [2]	NeurIPS'20	-	-	-	25.7±0.7	38.3±0.4	42.5±0.4	11.5±0.4	-	-
DC [40]	ICLR'21	31.2±1.4	76.1±0.6	82.3±0.3	28.3±0.5	44.9±0.5	53.9±0.5	12.8±0.3	25.2±0.3	-
DSA [37]	ICML'21	27.5±1.4	79.2±0.5	84.4±0.4	28.8±0.7	52.1±0.5	60.6±0.5	13.9±0.3	32.3±0.3	42.8±0.4
KIP [21]	NeurIPS'21	62.4±0.3	81.1±0.5	84.3±0.1	29.8±0.2	46.1±0.3	53.2±0.4	12.0±0.2	29.0±0.2	-
CAFE [32]	CVPR'22	42.9±3.3	77.9±0.6	82.3±0.3	30.3±1.1	46.3±0.6	55.5±0.6	12.9±0.3	27.8±0.3	37.9±0.3
DCC [19]	ICML'22	34.3±1.6	76.2±0.8	83.3±0.2	34.0±0.7	54.4±0.5	64.2±0.4	14.6±0.3	33.5±0.3	39.3±0.4
DataDAM [25]	ICCV'23	-	-	-	32.0±1.2	54.2±0.8	67.0±0.4	14.5±0.5	34.8±0.5	49.4±0.3
DM [39]	WACV'23	21.6±0.8	72.8±0.3	82.6±0.5	26.4±0.8	48.5±0.6	62.2±0.5	11.4±0.3	29.7±0.3	43.0±0.4
DM+Ours	-	-	75.7±0.3	**85.3±0.2**	-	55.1±0.1	65.1±0.2	-	32.2±0.5	43.6±0.3
IDM [41]	CVPR'23	65.3±0.3	81.0±0.1	84.1±0.1	45.2±0.5	57.3±0.3	67.2±0.1	23.1±0.2	44.7±0.1	49.9±0.2
IDM+Ours	-	**66.3±0.1**	**82.1±0.3**	85.1±0.5	**47.1±0.1**	**59.9±0.2**	**69.0±0.3**	**24.6±0.1**	**45.7±0.4**	**51.3±0.4**
Whole Dataset		95.4±0.2			84.8±0.1			56.2±0.3		

and 1.8% on CIFAR10 at IPC=1, 10, and 50, respectively, and 1.5%, 1.0%, and 1.4% on CIFAR100 at IPC=10, 50. Our IDM-based method also outperforms all other state-of-the-art methods, achieving accuracies of 59.9% and 69.0% on CIFAR10, and 45.7% and 51.3% on CIFAR100, both at IPC=10 and 50. On the larger TinyImageNet dataset, our method also achieves state-of-the-art performance with an accuracy of 23.3% at IPC=10, surpassing the second best method by 1.3%. For the simpler SVHN dataset with IPC=10, using DM as the baseline proved most effective, suggesting that simpler approaches may be more beneficial for less complex datasets.

Notably, our method, when used with DM as the baseline, does not provide IPC=1 results due to its requirement for a sample size greater than 1. However, IDM, with its ability to preserve more samples at IPC=1 through partitioning and expansion augmentation, is compatible with our approach.

4.3. Cross-architecture generalization

Cross-architecture generalization is a crucial metric for evaluating dataset distillation, particularly because it is hard to predict the neural network architectures that will be used in a real application. Significant performance drop upon changing model architectures is undesirable. We evaluate the synthetic dataset generated by our method on CIFAR10 across four different architectures, comparing it with previ-

Table 2. Comparative analysis of dataset distillation methods on higher resolution datasets. The **best** results in each column are highlighted for clarity.

Method	Venue	TinyImageNet		
IPC		1	10	50
Ratio (%)		0.2	2	10
Random	Classic	1.4±0.1	5.0±0.2	15.0±0.4
Herding [34]	Classic	2.8±0.2	6.3±0.2	16.7±0.3
K-Center [27]	Classic	1.6±0.2	5.1±0.1	15.0±0.3
Forgetting [30]	Classic	1.6±0.2	5.1±0.3	15.0±0.1
DataDAM [25]	ICCV'23	8.3±0.4	18.7±0.3	**28.7±0.3**
DM [39]	WACV'23	3.9±0.2	12.9±0.4	24.1±0.3
DM+Ours	-	-	15.4±0.2	26.1±0.1
IDM [41]	CVPR'23	9.8±0.2	21.9±0.2	26.2±0.3
IDM+Ours	-	**10.0±0.1**	**23.3±0.1**	27.5±0.3
Whole Dataset		37.6±0.6		

ous methods. Tabs. 3 and 4 present the cross-architecture performance for IPC=10 and 50. Our experiments included widely-used models such as ConvNet, AlexNet [7], VGG11 [29], and ResNet18 [14], with architectural details following previous work. Each experiment is repeated five times to determine the mean value.

For IPC=50 as shown in Tab. 4, our method's per-

Table 3. Cross-architecture testing on CIFAR10 with IPC=10.

	ConvNet	AlexNet	VGG11	ResNet18
DM [39]	48.9±0.6	38.8±0.5	42.1±0.4	41.2±1.1
DSA [37]	52.1±0.7	35.9±2.6	43.2±0.5	42.8±1.0
KIP [21]	47.6±0.9	24.4±3.9	42.1±0.4	36.8±1.0
IDM [41]	53.0±0.3	44.6±0.8	47.8±1.1	44.6±0.4
IDM+Ours	**59.9.±0.2**	**56.8±0.02**	**58.0±0.5**	**56.9±0.5**

Table 4. Cross-architecture testing on CIFAR10 with IPC=50.

	ConvNet	AlexNet	VGG11	ResNet18
DC [40]	53.9±0.5	28.8±0.7	38.8±1.1	20.9±1.0
CAFE [32]	62.3±0.4	43.2±0.4	48.8±0.5	43.3±0.7
DSA [37]	60.6±0.5	53.7±0.6	51.4±1.0	47.8±0.9
DM [39]	63.0±0.4	60.1±0.5	57.4±0.8	52.9±0.4
KIP [21]	56.9±0.4	53.2±1.6	53.2±0.5	47.6±0.8
DataDAM [25]	67.0±0.4	63.9±0.9	64.8±0.5	60.2±0.7
IDM+Ours	**69.0±0.2**	**67.3±0.2**	**67.3±0.3**	**68.3±0.2**

formance on ResNet18 shows an 8% improvement over DataDAM [25]. More notably, across ConvNet, AlexNet, VGG11, and ResNet18, we record accuracies of 69.0%, 67.3%, 67.3%, and 68.3%, respectively, maintaining a performance drop of less than 1.7% across these architectures. This is particularly significant considering that previous methods often experienced substantial performance drops on ResNet18. For IPC=10 as shown in Tab. 3, our method performs significantly better than others, with a performance loss not exceeding 3.1% on AlexNet, VGG11, and ResNet18, while KIP [21] loss 10.8% from ConvNet to Resnet18. The better cross-architecture generalization performance achieved by our method allows us to utilize more diverse model architectures in the dataset distillation phase perhaps facilitating the synthesis of datasets that are more generalizable across different model architectures.

Note that some existing methods only appear in one of the two tables because they only reported the result for either IPC = 10 or IPC = 50.

4.4. Ablation study

Analysis of cluster constraint threshold. The β in Eq. (3) serves as the centralization threshold, where a larger β indicates sample features are farther from the class feature center, while a smaller β brings them closer. We conducted experiments with varying β values, ranging from 0.0 to 2.0. To isolate the effects of β, we keep α in Eq. (3) constant and do not include our proposed covariance matching constraint. The results on CIFAR10 with IPC=10 are visualized in Fig. 3, and the performance corresponding to each β value is detailed in Tab. 5.

Our visualizations demonstrate that feature distribution per class becomes more dispersed with larger β values and more concentrated with smaller β. Performance evaluation experiments reveal that optimal performance is achieved when β is below 0.7. This suggests that smaller β values, which lead to higher class discrimination, are advantageous for dataset distillation, enhancing performance without significantly compromising generalization.

Effectiveness of each component. To validate the effectiveness of the proposed two components, we perform ablation experiments. We test the performance impact of implementing the class centralization constraint and covariance matching constraint both separately and combined. Tab. 6 reveals that each constraint individually enhances DM's performance by 3.7% and 3.1% on CIFAR10, and 1.4% and 1.5% on CIFAR100, respectively. Notably, the greatest performance improvement, 6.6% on CIFAR10 and 2.5% on CIFAR100, is achieved when both components are used concurrently.

Evaluation of weighting parameter λ. We perform a sensitivity analysis focusing on the weighting parameters in Eq. (8). The results, shown in Fig. 4, indicate that larger values of λ_{CC} and λ_{CM} correspond to a more pronounced impact on the optimization objective. Varying λ_{CC} within a range of 0.01 to 0.09 resulted in a performance difference of 1.47%, suggesting moderate sensitivity. Similarly, altering λ_{CM} between 0.005 and 0.015 lead to a 1.1% performance variation, indicating comparable sensitivity levels.

Number of iterations required for convergence. Contrary to previous methods which typically require around 20,000 training iterations, our method achieves convergence with significantly fewer iterations. As demonstrated in Fig. 5, less than 2,000 iterations are needed for IPC=10 to attain peak performance. For IPC=50, optimal performance is essentially reached at about 3,000 iterations. In addition to this, we find that accuracy grows quickly in the early stages of training, and if the training time requirement is high while the performance requirement is relatively low, we can consider using early stopping training methods.

Different compression ratios. Previous dataset distillation studies have primarily focused on improving performance at very small compression ratios, such as 0.2% (IPC=10 on CIFAR10) and 1% (IPC=50 on CIFAR10). However, an equally important consideration is determining the necessary compression ratio to retain performance comparable to the full dataset. To explore this, we conduct experiments at higher IPC values, including IPC=100, 200, 400, 600, 800, and 1000, and compared our method's performance with previous methods, as shown in Fig. 6 and Tab. 7, with most results sourced from DC-BENCH [6].

Our method significantly outperforms others like DC [40], DSA [37], and DM [39] at these varied IPC levels. Notably, at IPC=1000 (a 20% compression ratio), our

Table 5. Comparison of performance for different values of β on CIFAR10 with IPC=10.

β	0.0	0.1	0.7	0.9	1.0	1.1	1.2	1.3	1.5	2.0
Acc (%)	**52.2±0.2**	**52.2±0.1**	**52.2±0.1**	51.5±0.3	51.1±0.2	49.9±0.2	49.2±0.2	49.1±0.1	48.9±0.3	48.9±0.1

(a) $\beta = 0.0$ (b) $\beta = 0.1$ (c) $\beta = 0.5$ (d) $\beta = 0.7$ (e) $\beta = 0.9$

(f) $\beta = 1.1$ (g) $\beta = 1.2$ (h) $\beta = 1.3$ (i) $\beta = 1.5$ (j) $\beta = 2.0$
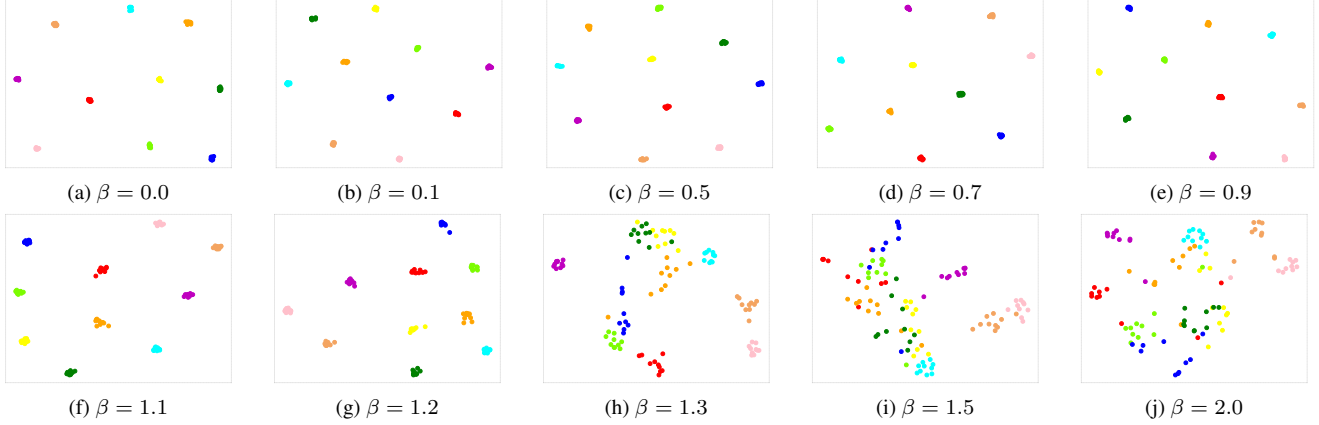
Figure 3. Visualization of different β on CIFAR10 with IPC=10.

Table 6. Ablation study on the proposed two components. Results are averaged over 5 runs.

Method	CIFAR10	CIFAR100
Baseline	48.5±0.2	29.7±0.1
+ Class centralization	52.2±0.3	31.1±0.2
+ Covariance matching	51.6±0.1	31.2±0.1
+ Both	**55.1±0.1**	**32.2±0.3**

(a) IPC=10   (b) IPC=50
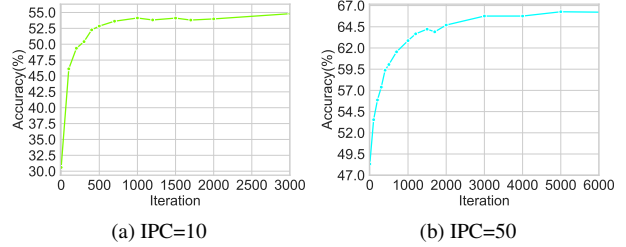
Figure 5. Accuracy progression over iteration on CIFAR10.

(a) Ablation of λ_{CC} (b) Ablation of λ_{CM}
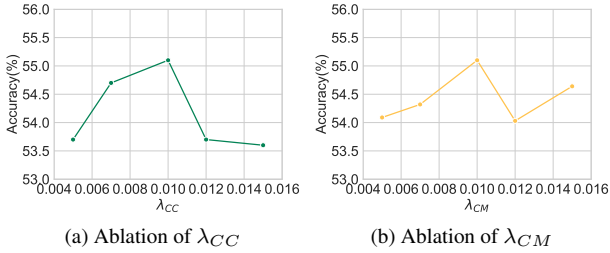
Figure 4. Ablation study of the weighting parameter on CIFAR10.

Figure 6. Performance comparison at different compression ratios on CIFAR10.

method achieves 93.3% of the full dataset's performance. However, it's observed that as the compression ratio increases, the performance gap between all methods, including classical and even random methods, narrows. This suggests that current dataset distillation approaches are more effective at smaller compression ratios. In future research, it would be valuable to explore dataset distillation methods suitable for larger compression ratios, aiming for an optimal balance between performance and data reduction.
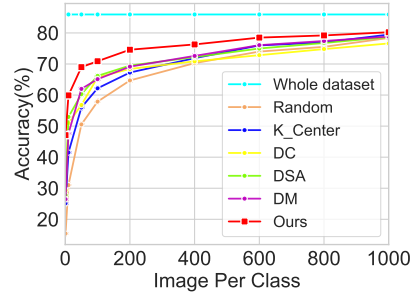
4.5. Applications

Continual learning. The goal of continual learning is to develop models capable of adapting to new tasks while minimizing the forgetting of previously learned tasks. Dataset distillation, with its ability to maintain performance with a minimal number of samples, has potential applications

Table 7. Accuracy (%) with different compression ratios on CI-FAR10.

IPC	Random	K-Center	DC	DSA	DM	Ours
1	15.40	25.16	29.34	27.76	26.45	**47.11**
10	31.00	41.49	50.99	52.96	47.64	**59.92**
50	50.55	56.00	56.81	60.28	61.99	**69.01**
100	57.89	62.18	65.70	66.18	65.12	**70.46**
200	64.70	67.25	68.41	69.49	69.15	**74.15**
400	70.28	71.88	70.86	72.22	72.61	**76.35**
600	74.00	75.98	72.84	74.99	76.07	**78.38**
800	75.52	76.94	74.80	76.76	77.41	**79.21**
1000	78.38	79.47	76.62	78.68	78.83	**80.31**

(a) Step=5   (b) Step=10
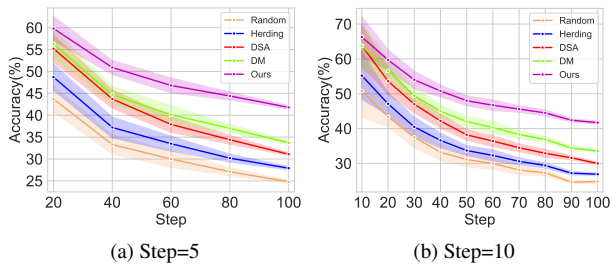
Figure 7. Continual learning on CIFAR100.

(a) SVHN   (b) CIFAR10

(c) CIFAR100 (Partial)   (d) TinyImageNet (Partial)

Figure 8. Visualization of synthetic images.

in continual learning. Following the setup from the prior work [39], we limit the buffer size to 20 images per category. Experiments are conducted in both 5-step (20 classes per task) and 10-step (10 classes per task) scenarios. For each step, three networks are randomly initialized, trained on the synthetic dataset from the buffer, and evaluated on a test set of seen classes to determine the mean performance. Five class orders are randomly generated for these experiments, and the mean and variance are calculated. The results, depicted in Fig. 7, show that our method notably outperforms DM [39], DSA [37], and Herding [34], with the performance gap widening as the number of steps increases.

4.6. Visualization

The synthetic datasets, initialized randomly from real datasets, are visualized in Fig. 8. These include synthesized datasets from SVHN, CIFAR10/100, and Tiny-ImageNet at IPC=10. For datasets with smaller resolutions like SVHN and CIFAR10/100, the majority of high-frequency information is preserved, making them more recognizable to the human eye. In contrast, for higher resolution datasets such as Tiny-ImageNet, the synthetic dataset's visualization appears much different from the real dataset, becoming less distinguishable visually. Differential visualization results appeared between low-resolution data sets and high-resolution data sets, suggesting that in the future we may need to design different dataset distillation methods for datasets of different scales.
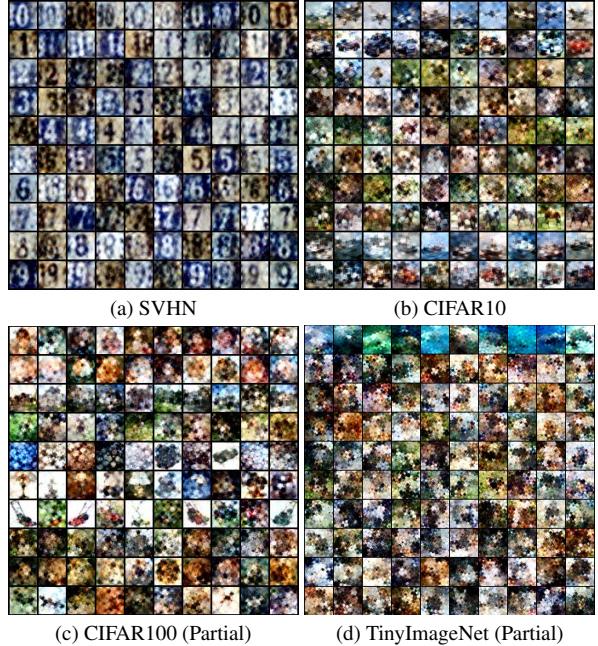
5. Conclusion

Previous distribution matching-based methods for dataset distillation face two primary challenges: insufficient class discrimination and incomplete distribution matching. To address these, we introduced the class centralization constraint and the covariance matching constraint, focusing on improving both inter-sample and inter-feature relations. The class centralization constraint improves class discrimination by clustering samples closer to their class centers, while the covariance matching constraint aligns inter-feature relationships between real and synthetic datasets. Our method, tested across various resolutions, has demonstrated significant superiority over previous methods and excelled in cross-architecture scenarios. Additionally, we explored larger compression ratios to determine the necessary compression ratio for maintaining performance comparable to the full dataset.

References

[1] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019. 2

[2] Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales. Flexible dataset distillation: Learn labels instead of images. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS Workshop)*, 2020. 4, 5

[3] Nicholas Carlini, Vitaly Feldman, and Milad Nasr. No free lunch in" privacy for free: How does dataset condensation help privacy". *Proceedings of the International Conference on Machine Learning (ICML)*, 2022. 1, 2

[4] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 233–248, 2018. 2

[5] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4750–4759, 2022. 1, 2

[6] Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Dcbench: Dataset condensation benchmark. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 35:810–822, 2022. 1, 2, 6

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 248–255, 2009. 5

[8] Zhiwei Deng and Olga Russakovsky. Remember the past: Distilling datasets into addressable memories for neural networks. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 35:34391–34404, 2022. 2

[9] Tian Dong, Bo Zhao, and Lingjuan Lyu. Privacy for free: How does dataset condensation help privacy? In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 5378–5396, 2022. 1, 2

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 1

[11] Jiawei Du, Yidi Jiang, Vincent YF Tan, Joey Tianyi Zhou, and Haizhou Li. Minimizing the accumulated trajectory error to improve dataset distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3749–3758, 2023. 1, 2

[12] Reza Zanjirani Farahani and Masoud Hekmatfar. *Facility location: concepts, models, algorithms and case studies*. Springer Science & Business Media, 2009. 1, 2

[13] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021. 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 770–778, 2016. 4, 5

[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 4700–4708, 2017. 1

[16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4

[17] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 4

[18] Hae Beom Lee, Dong Bok Lee, and Sung Ju Hwang. Dataset condensation with latent space knowledge factorization and sharing. *arXiv preprint arXiv:2208.10494*, 2022. 2, 4

[19] Saehyung Lee, Sanghyuk Chun, Sangwon Jung, Sangdoo Yun, and Sungroh Yoon. Dataset condensation with contrastive signals. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 12352–12364, 2022. 4, 5

[20] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 10012–10022, 2021. 1

[21] Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 34:5186–5198, 2021. 2, 4, 5, 6

[22] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 2001–2010, 2017. 2

[23] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 779–788, 2016. 1

[24] Andrea Rosasco, Antonio Carta, Andrea Cossu, Vincenzo Lomonaco, and Davide Bacciu. Distilled replay: Overcoming forgetting through synthetic samples. In *International Workshop on Continual Semi-Supervised Learning*, pages 104–117, 2021. 1, 2

[25] Ahmad Sajedi, Samir Khaki, Ehsan Amjadian, Lucy Z Liu, Yuri A Lawryshyn, and Konstantinos N Plataniotis. Datadam: Efficient dataset distillation with attention matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 17097–17107, 2023. 1, 2, 4, 5, 6

[26] Mattia Sangermano, Antonio Carta, Andrea Cossu, and Davide Bacciu. Sample condensation in online continual learning. In *International Joint Conference on Neural Networks (IJCNN)*, pages 01–08, 2022. 1, 2

[27] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *Proceedings*

of the International Conference on Learning Representations (ICLR), 2018. 1, 2, 4, 5

[28] Pierre Sermanet, Soumith Chintala, and Yann LeCun. Convolutional neural networks applied to house numbers digit classification. In *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*, pages 3288–3291, 2012. 4

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, 2014. 5

[30] 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. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 2, 4, 5

[31] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 7472–7481, 2018. 1

[32] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12196–12205, 2022. 1, 2, 4, 5, 6

[33] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. 1, 2, 4, 5

[34] Max Welling. Herding dynamical weights to learn. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1121–1128, 2009. 1, 2, 4, 5, 8

[35] Yang Yang, Da-Wei Zhou, De-Chuan Zhan, Hui Xiong, and Yuan Jiang. Adaptive deep models for incremental learning: Considering capacity scalability and sustainability. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 74–82, Anchorage, AK, 2019. 1

[36] Yang Yang, Zhen-Qiang Sun, Hengshu Zhu, Yanjie Fu, Yuanchun Zhou, Hui Xiong, and Jian Yang. Learning adaptive embedding considering incremental class. *IEEE Trans. Knowl. Data Eng.*, 35(3):2736–2749, 2023. 1

[37] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 12674–12685, 2021. 1, 2, 4, 5, 6, 8

[38] Bo Zhao and Hakan Bilen. Synthesizing informative training samples with gan. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS Workshop)*, 2022. 2

[39] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (CVPR)*, pages 6514–6523, 2023. 1, 2, 3, 4, 5, 6, 8

[40] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. 2021. 1, 2, 4, 5, 6

[41] Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7856–7865, 2023. 1, 2, 3, 4, 5, 6

[42] Tianhang Zheng and Baochun Li. Differentially private dataset condensation. 2022. 1, 2