TGDD: Trajectory Guided Dataset Distillation with Balanced Distribution

Fengli Ran[1,2], **Xiao Pu**[1], **Bo Liu**[1*], **Xiuli Bi**[1*], **Bin Xiao**[1,3],

[1]Chongqing University of Posts and Telecommunications
[2]Chongqing Polytechnic University of Electronic Technology
[3]Jinan Inspur Data Technology Co., Ltd.
fengliran@hotmail.com, {puxiao, boliu, bixl, xiaobin}@cqupt.edu.cn

Abstract

Dataset distillation compresses large datasets into compact synthetic ones to reduce storage and computational costs. Among various approaches, distribution matching (DM)-based methods have attracted attention for their high efficiency. However, they often overlook the evolution of feature representations during training, which limits the expressiveness of synthetic data and weakens downstream performance. To address this issue, we propose Trajectory Guided Dataset Distillation (TGDD), which reformulates distribution matching as a dynamic alignment process along the model's training trajectory. At each training stage, TGDD captures evolving semantics by aligning the feature distribution between the synthetic and original dataset. Meanwhile, it introduces a distribution constraint regularization to reduce class overlap. This design helps synthetic data preserve both semantic diversity and representativeness, improving performance in downstream tasks. Without additional optimization overhead, TGDD achieves a favorable balance between performance and efficiency. Experiments on ten datasets demonstrate that TGDD achieves state-of-the-art performance, notably a 5.0% accuracy gain on high-resolution benchmarks.

Code — https://github.com/FlyFinley/TGDD

Introduction

In the deep learning era, the exponential expansion of dataset sizes has brought substantial computational and storage burdens. For instance, data volumes have ballooned from ImageNet's tens of millions (Krizhevsky, Sutskever, and Hinton 2017) to LAION-5B's billions of image–text pairs (Schuhmann et al. 2022). Reducing large datasets to smaller ones that can preserve performance under resource constraints has become a key challenge in advancing AI.

To tackle this challenge, a strategy is coreset selection (Feldman 2020; Chai et al. 2023), which picks a small representative subset from the original dataset. However, this approach discards most samples, overlooks their training potential, and diminishes the information available for downstream tasks. Another strategy is dataset distillation (Cazenavette et al. 2023; Yang et al. 2024), which directly synthesizes a compact set with higher information density to approximate the performance of the full dataset.

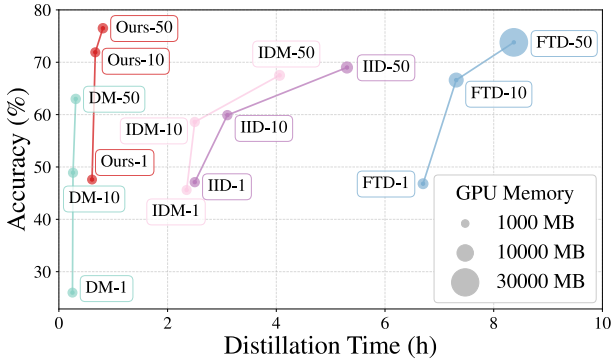
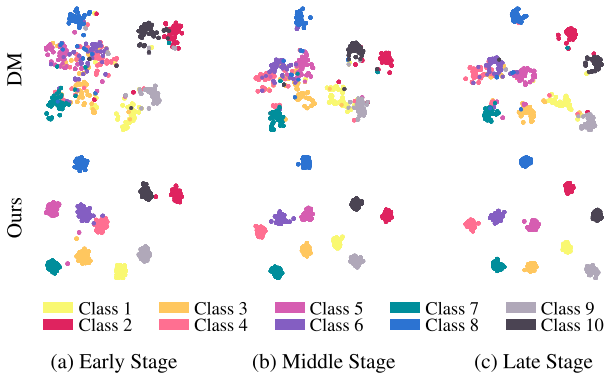


Figure 1: Accuracy, distillation time, and GPU memory comparison on CIFAR-10 under different IPCs. Pretraining time is included (5 trajectories for ours, 100 for FTD). Our method balances performance and cost effectively.

Existing dataset distillation methods generally fall into two categories: optimization-oriented (OO)-based and distribution-matching (DM)-based approaches. OO-based techniques recreate the learning dynamics by forcing synthetic data to induce the same gradient (Zhao and Bilen 2021; Kim et al. 2022) or parameter (Cazenavette et al. 2022; Cui et al. 2023) as the original data throughout training. As shown in Figure 1, although methods like FTD (Du et al. 2023) are effective, their iterative updates between data and model incur heavy computational costs, limiting their scalability to large datasets. In contrast, DM-based methods focus on the statistical properties of the data, directly aligning synthetic and original feature distributions in the embedding space without requiring model updates, thereby dramatically accelerating the distillation process.

A critical problem for DM-based methods is effectively mapping data into the embedding space. However, most DM-based approaches adopt randomly initialized networks for feature extraction, overlooking how model representations evolve throughout training (Zeiler and Fergus 2014; Rahaman et al. 2019), which yields inadequate embeddings. As shown in Figure 2, DM's synthetic datasets exhibit significantly different class separability across models; only

*Corresponding Author



(a) Early Stage (b) Middle Stage (c) Late Stage

Figure 2: t-SNE visualization of synthetic features generated by DM and our method under IPC-50, using pretrained models on CIFAR-10 at different training stages.

highly optimized networks can distinguish categories effectively. This makes synthetic sets hard to learn and undermines downstream performance.

To address these issues, we propose Trajectory Guided Dataset Distillation (TGDD), which recasts the static distribution matching into a dynamic balancing process along the trajectory. Specifically, TGDD first builds expert trajectories by saving model snapshots at different training stages. Leveraging these trajectories, TGDD then aligns the feature distributions of the synthetic and original data at every stage, enriching the semantic diversity of the synthetic dataset. Simultaneously, TGDD applies a stage-wise distribution constraint regularization, enforcing intra-class compactness and improving representativeness of the synthetic dataset. As shown in Figure 2, TGDD balances the diversity and representativeness of the synthetic dataset throughout training, making it a good surrogate of the original dataset and improving its downstream performance. Since expert trajectories are constructed using only the original dataset, they can be pretrained and reused in different settings. TGDD achieves a favorable balance between distillation performance and computational efficiency.

In summary, our contributions are as follows:

- We reformulate distribution matching as a dynamic balance process between feature alignment and distribution constraint throughout training, unifying representativeness and diversity in the synthetic distribution.
- We introduce Trajectory-Guided Distribution Distillation (TGDD), which leverages multiple pretrained expert trajectories to perform efficient feature alignment and stage-wise distribution constraint during training.
- Extensive experiments across ten datasets demonstrate that TGDD achieves state-of-the-art performance, notably, a 5.0% accuracy gain is achieved on high-resolution benchmarks.

Related Work

Coreset Selection

Coreset selection (Chai et al. 2023; Lee et al. 2024) aims to extract a small, representative subset from a large dataset and has been widely adopted in continual learning (Rebuffi et al. 2017; Aljundi et al. 2019) and active learning (Agarwal et al. 2020; Kim and Shin 2022). Early approaches rely on uniform random sampling, while later work develops more advanced criteria. Herding (Welling 2009) iteratively minimizes the distance between the subset and the full-set feature centroids. K-Center (Sener and Savarese 2017) refines this idea by choosing samples that minimize the dataset's maximum covering radius. Forgetting (Toneva et al. 2018) retains examples most prone to being forgotten during training. Despite their efficiency, these approaches irrevocably discard significant portions of the original data and thus cannot guarantee optimal downstream performance (Lei and Tao 2023).

Dataset Distillation

Dataset distillation compresses large datasets into smaller ones while preserving original information (Yu, Liu, and Wang 2023), benefiting tasks such as neural architecture search (Such et al. 2020), federated learning (Pi et al. 2023; Wang et al. 2024b), continual learning (Yang et al. 2023), and privacy protection (Dong, Zhao, and Lyu 2022). Existing methods can be broadly categorized as follows.

Optimization-Oriented (OO)-Based Methods cast dataset distillation as bilevel optimization, updating model parameters on synthetic data while refining that data to preserve original performance. The pioneering work DD (Wang et al. 2018) introduces the concept of dataset distillation from a meta learning perspective. Later, DC (Zhao, Mopuri, and Bilen 2021) expands the idea by matching gradients from the original and synthetic data. DSA (Zhao and Bilen 2021) further enhances performance by incorporating differentiated Siamese augmentation. Instead of matching gradients, MTT (Cazenavette et al. 2022) aligns model parameters over training trajectories. FTD (Du et al. 2023) mitigates trajectory drift by reducing cumulative errors across steps. DATM (Guo et al. 2024) scales trajectory matching by adapting sample difficulty to dataset size. Despite improved performance, their bilevel optimization incurs significant computational cost.

Distribution-Matching (DM)-Based Methods directly align the feature distributions of synthetic and original data in latent space, bypassing bilevel optimization to improve efficiency. DM (Zhao and Bilen 2023) introduces the concept by aligning class-wise centroids. Although fast, its performance is limited by oversimplified feature matching. DataDAM (Sajedi et al. 2023) extends this idea by incorporating multi-layer alignment through attention mechanisms. IDM (Zhao et al. 2023) adopts a dynamic model queue to extract more informative representations, but incurs extra training costs. M3D (Zhang et al. 2024b) further projects features into a Reproducing Kernel Hilbert Space for finer alignment. DANCE (Zhang et al. 2024a) interpolates initial and converged models to create pseudo-intermediate feature extractors. Although distribution matching shortens distillation time, it ignores representation evolution and lacks inter-class compactness, leading to scattered features and poor separability. We address these gaps with trajectory-aware alignment and compactness regularization to produce highly discriminative synthetic data.

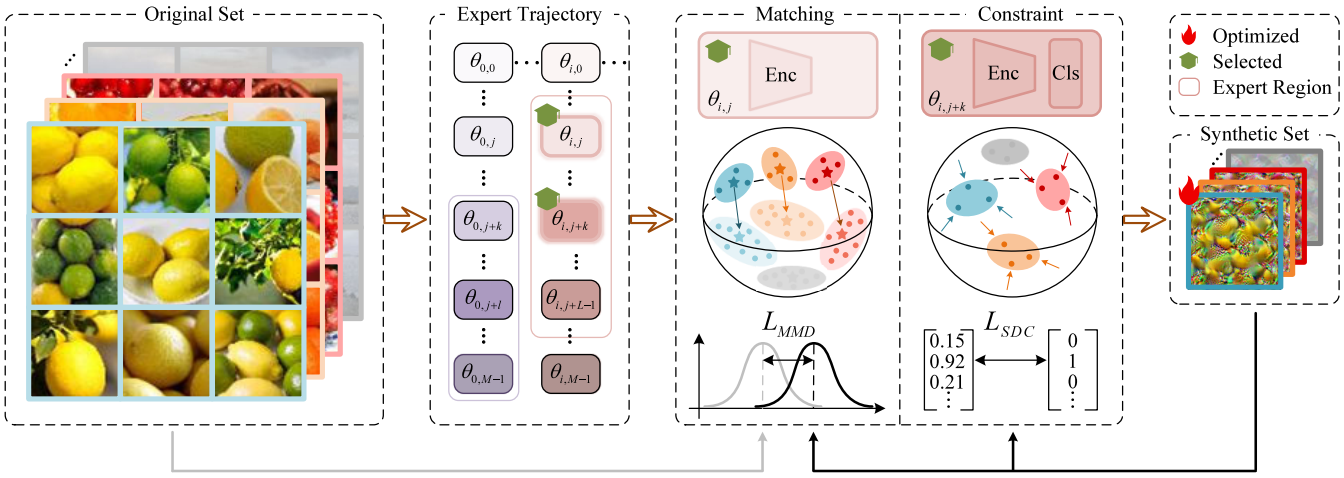


Figure 3: The illustration of our proposed method. First, we pretrain N expert trajectories with original dataset, each comprising M network snapshots. Then, one snapshot is randomly sampled as encoder for distribution matching between original and synthetic dataset, another snapshot in the expert region is chosen to impose distribution constraints on the synthetic dataset.

Method

Preliminary

Dataset Distillation Given a large source training dataset $T = \{(x_i, y_i)\}|_{i=1}^{|T|}$, the objective of dataset distillation is to synthesize a smaller target synthetic dataset $S = \{(s_j, s_j)\}|_{j=1}^{|S|}$, where $|S| \ll |T|$. When we evaluate the original data on models trained with T and S respectively, we hope to have a similar generalization performance. To be specific, let (x_i, y_i) be the data pair sampled from the original data distribution, l be the classification loss function such as cross entropy, θ^T and θ^S be the models trained from T and S respectively, our aim is to minimize the generalization performance gap:

$$E_{x\sim P_D}[(l_{\theta^T}(x), y)] \simeq E_{x\sim P_D}[(l_{\theta^S}(x), y)] \tag{1}$$

Distribution Matching Prior methods for dataset distillation mostly rely on OO-based techniques like gradient or trajectory matching. As these methods require simultaneous optimization of the network and synthetic dataset, they suffer from high computational overhead and struggle to scale to large datasets. To improve computational efficiency, the DM-based approaches constrain the embedding features of the original and synthetic datasets through the maximum mean discrepancy (MMD) (Gretton et al. 2012), requiring only optimization of the synthetic dataset. We have:

$$S^* = \arg\min_S \mathbb{E}_{\theta\sim P_{\theta_0}} \left\| \frac{1}{T}\sum_{i=1}^{|T|} \psi_\theta(x_i) - \frac{1}{S}\sum_{j=1}^{|S|} \psi_\theta(s_j) \right\|^2 \tag{2}$$

Here $\theta \sim P_{\theta_0}$, most distribution matching methods employ randomly initialized networks as feature extractors.

Expert Trajectory Construction

Expert trajectories comprise model snapshots captured at different epochs using original data, defining the performance upper bound for models trained on the dataset. Trajectory-matching methods leverage these trajectories as their foundation. Despite achieving strong performance, such methods incur substantial storage overhead because of preserving a large number of trajectories. For instance, MTT (Cazenavette et al. 2022) stores 200 trajectories.

In contrast, conventional DM-based approaches typically employ randomly initialized models, overlooking the potential of expert trajectories. We demonstrate that strategic integration of expert trajectories significantly enhances distribution matching for dataset distillation. Our framework therefore incorporates expert trajectories by training N randomly initialized neural networks that converge within M epochs, thereby forming expert trajectories as:

$$\mathbf{P} = \{p_{i,j} \mid 0 \le i \le N,\ 0 \le j \le M\}. \tag{3}$$

Since expert trajectories require only original data for training, we can pretrain models before distillation. This approach retains efficiency of distribution matching, significantly faster than OO-based methods that train networks during distillation. Moreover, our method achieves competitive performance with even one trajectory, substantially reducing storage overhead compared to trajectory matching techniques that require massive trajectory preservation.

Stage-wise Distribution Matching

Conventional DM-based methods typically employ randomly initialized networks as feature extractors, aligning features between synthetic and original data. However, these homogeneous initialization strategies capture only early-stage feature distributions, neglecting evolving patterns throughout the training trajectory.

Inspired by OO-based methods that perform gradient matching or trajectory matching across the whole training process, we leverage pretrained expert trajectories for feature distribution at different training stages. This approach eliminates network training during distillation for efficiency

while providing diverse feature representations spanning all optimization phases.

During each distillation iteration, we randomly select an expert trajectory $\mathbf{P}_i$ and then sample a pretrained model $\theta_{ext} = p_{i,j}$ at arbitrary training stage. The encoder component of this model computes representations for the original dataset and the synthetic dataset. We then align these representations through distribution matching.

We formulate the goal of the stage-wise distribution matching as follows:

$$L_{\text{MMD}} = \sum_{c=1}^{C} \left\| \frac{1}{|B_c^T|} \sum_{i=1}^{|B_c^T|} \psi_{\theta_{\text{ext}}}(x_i) - \frac{1}{|B_c^S|} \sum_{i=1}^{|B_c^S|} \psi_{\theta_{\text{ext}}}(s_i) \right\|^2 \tag{4}$$

where θ_{ext} is the feature extractor network weight sampled from the expert trajectories, B_c^T and B_c^S are mini-batch pairs sampled from T and S respectively for class c.

Stage-wise Distribution Constraint

Conventional distribution matching methods rely exclusively on Maximum Mean Discrepancy (MMD) for feature alignment. However, MMD constrains only mean of feature distributions, providing insufficient regularization. This leads to overly dispersed synthetic data distributions and severe class boundary confusion.

Effective distribution constraints are therefore crucial for enhancing inter-class discriminability in synthetic datasets. By fully exploiting expert trajectories, we introduce a distribution constraint regularization that operates on expert regions. Using different experts in training iterations, the method achieves ensemble-like performance without incurring additional training cost.

Specifically, at each step of the distillation process, once the feature extractor $\theta_{ext} = p_{i,j}$ is determined, we construct an expert region consisting of L pretrained expert networks $P_{er} = \{p_{i,j}, p_{i,j+1}, ..., p_{i,j+L-1}\}$. The expert model $\theta_{exp} = p_{i,j+k}$ is then randomly selected from the region and employed to impose the distributional constraint. We calculate the stage-wise distribution constraint loss as:

$$L_{SDC} = \frac{1}{B_c^S} \sum_{c=1}^{C} \sum_{i=1}^{|B_c^S|} l(\phi_{exp}(s_i), y_i) \tag{5}$$

where θ_{exp} is the expert network weight sampled from the expert region, B_c^S are mini-batch pairs sampled from S for class c.

Training Algorithm

The overall loss function is described in equation 6

$$L_{overall} = L_{MMD} + \alpha L_{SDC} \tag{6}$$

where $\mathcal{L}_{\text{MMD}}$ is the distribution matching loss, $\mathcal{L}_{\text{SDC}}$ is the distribution constraint loss and α is the regularization coefficient. Both feature extraction and constraint networks are sampled from the expert trajectories. Algorithm 1 details our proposed method.

Algorithm 1: Trajectory Guided Dataset Distillation with Balanced Distribution

Input: original dataset D^T, regularization coefficient α, number of iterations $Iter$, expert region distance L, synthetic dataset learning rate η

1: train N expert trajectories, each consists of M snapshots, $\mathbf{P} = \{p_{i,j} \mid 0 \le i \le N,\ 0 \le j \le M\}$
2: **for** $i = 1, 2, \cdots, Iter$ **do**
3: Initialize synthetic dataset D^S with random selected data from D^T
4: Sample an expert trajectory $\mathbf{P}_i$
5: Sample a feature extract network $p_{i,j}$ from $\mathbf{P}_i$
6: Calculate the MMD loss with Equation 4
7: Sample an expert network $p_{i,j+k}$ from $p_{i,j}$ to $p_{i,j+L-1}$
8: Calculate the SDC loss with Equation 5
9: Calculate the total loss with Equation 6 and update D^S with $D^S = D^S - \eta \nabla_{D^S} L_{overall}$
10: **end for**

Output: synthetic dataset D_{syn};

Experiments

Experimental Setup

Datasets. We assessed dataset distillation across multiple datasets, including the low-resolution SVHN (Netzer et al. 2011), CIFAR-10 and CIFAR-100 (32×32) (Krizhevsky, Hinton et al. 2009), the medium-resolution Tiny ImageNet (64×64) (Le and Yang 2015), and high-resolution ImageNet subsets (128×128)(Deng et al. 2009), namely ImageNette, ImageWoof, ImageFruit, ImageMeow, ImageSquawk, and ImageYellow.

Networks. Following previous work, we adopted ConvNet (Gidaris and Komodakis 2018) for dataset distillation. For SVHN, CIFAR-10 and CIFAR-100, a 3-layer ConvNet was used. Each layer had a 128 kernel 3×3 convolutional kernel, instance normalization (Ulyanov, Vedaldi, and Lempitsky 2016), ReLU activation (Nair and Hinton 2010), and a 3×3 average pooling layer with a stride of 2. For Tiny ImageNet and ImageNet subsets, a 4-layer and 5-layer ConvNet was applied respectively.

Implementation Details. During distillation, the optimizer's learning rate is set to 0.1 for ImageNet subsets and 0.01 for others, scaled by the images per class. In training, an SGD optimizer with a 0.01 learning rate, 0.9 momentum, and 0.0005 weight decay is used. Regarding the hyperparameter α , we set 2.5 for 1 and 10 images per class, and 0.5 for 50 images per class. Hyperparameter L is set to 7 in all settings. We trained 5 expert trajectories with 60 epochs for SVHN, CIFAR-10, CIFAR-100 and Tiny ImageNet, and 80 epochs for ImageNet subsets. Following previous research (Zhao and Bilen 2021), we used differential augmentation like color transformation, random crop, cutout, random flip, scale and rotate transformation. We also use multiformation parameterization as in (Kim et al. 2022), where the factor parameter ρ is set to 3 for ImageNet subsets and 2 for other datasets. Each experiment was repeated 5 times.

Datasets	SVHN			CIFAR-10			CIFAR-100			TinyImageNet		
IPC	1	10	50	1	10	50	1	10	50	1	10	50
Ratio (%)	0.02	0.14	0.7	0.02	0.2	1	0.02	0.2	1	0.2	2	10
Whole	95.4 ± 0.1			84.8 ± 0.1			56.2 ± 0.3			37.6 ± 0.4		
Random	14.4 ± 0.2	26.0 ± 1.2	43.4 ± 1.0	14.4 ± 0.2	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	15.0 ± 0.4
Herding	21.5 ± 1.2	31.6 ± 0.7	40.4 ± 0.6	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	16.7 ± 0.3
Forgetting	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 ± 0.2	15.1 ± 0.3	30.5 ± 0.3	1.6 ± 0.1	5.1 ± 0.2	15.0 ± 0.3
DC	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	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	-	-	-
MTT	-	-	-	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	28.0 ± 0.3
FTD	-	-	-	46.8 ± 0.3	66.6 ± 0.3	73.8 ± 0.2	25.2 ± 0.2	43.4 ± 0.3	50.7 ± 0.3	10.4 ± 0.3	24.5 ± 0.2	-
CAFE	42.6 ± 3.3	75.9 ± 0.6	81.3 ± 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	-	-	-
DM	-	-	-	26.0 ± 0.8	48.9 ± 0.6	63 ± 0.4	11.4 ± 0.3	29.7 ± 0.3	43.6 ± 0.4	3.9 ± ±0.2	12.9 ± 0.4	24.1 ± 0.3
IDM	-	-	-	45.6 ± 0.7	58.6 ± 0.1	67.5 ± 0.1	20.1 ± 0.3	45.1 ± 0.1	50 ± 0.2	10.1 ± 0.2	21.9 ± 0.2	27.7 ± 0.3
M3D	**62.8 ± 0.5**	85.0 ± 0.1	86.2 ± 0.3	45.3 ± 0.3	63.5 ± 0.2	69.9 ± 0.5	26.2 ± 0.3	42.4 ± 0.2	50.9 ± 0.7	-	-	-
DSDM	60.2 ± 0.2	85.4 ± 0.3	91.3 ± 0.2	45.0 ± 0.4	66.5 ± 0.3	75.8 ± 0.3	19.5 ± 0.2	46.2 ± 0.3	54.0 ± 0.2	-	-	-
DANCE	-	-	-	47.1 ± 0.2	70.8 ± 0.2	76.1 ± 0.1	27.9 ± 0.2	49.8 ± 0.1	52.8 ± 0.1	11.6 ± 0.2	26.4 ± 0.3	28.9 ± 0.4
Ours	59.0 ± 0.7	**88.2 ± 0.3**	**92.8 ± 0.3**	**47.6 ± 0.2**	**71.9 ± 0.3**	**76.5 ± 0.2**	**28.5 ± 0.2**	**51.3 ± 0.2**	**54.6 ± 0.3**	**13.6 ± 0.1**	**29.3 ± 0.3**	**30.9 ± 0.4**

Table 1: Comparison with previous coreset selection and dataset distillation methods on low-resolution and medium-resolution datasets. IPC: images per class. Ratio (%): the ratio of synthetic images to the original dataset. Whole: the accuracy of the model trained with the whole training set.

Datasets	ImageNette		ImageWoof		ImageFruit		ImageMeow		ImageSquawk		ImageYellow	
IPC	1	10	1	10	1	10	1	10	1	10	1	10
Ratio(%)	0.105	1.05	0.11	1.1	0.077	0.077	0.077	0.077	0.077	0.077	0.077	0.077
Whole	87.4 ± 1.0		67.0 ± 1.3		63.9 ± 2.0		66.7 ± 1.1		87.5 ± 0.3		84.4 ± 0.6	
Random	23.5 ± 4.8	47.7 ± 2.4	14.2 ± 0.9	27.0 ± 1.9	13.2 ± 0.8	21.4 ± 1.2	13.8 ± 0.6	29.0 ± 1.1	21.8 ± 0.5	40.2 ± 0.4	20.4 ± 0.6	37.4 ± 0.5
MTT	47.7 ± 0.9	63.0 ± 1.3	28.6 ± 0.8	35.8 ± 1.8	26.6 ± 0.8	40.3 ± 1.3	30.7 ± 1.6	40.4 ± 2.2	39.4 ± 1.5	52.3 ± 1.0	45.2 ± 0.8	60.0 ± 1.5
FTD	52.2 ± 1.0	67.7 ± 0.7	30.1 ± 1.0	38.8 ± 1.4	29.1 ± 0.9	44.9 ± 1.5	33.8 ± 1.5	43.3 ± 0.6	-	-	-	-
DM	32.8 ± 0.5	58.1 ± 0.3	21.1 ± 1.2	31.4 ± 0.5	-	-	-	-	31.2 ± 0.7	50.4 ± 1.2	-	-
DANCE	57.2 ± 0.5	80.2 ± 0.7	30.6 ± 0.3	57.8 ± 1.1	30.6 ± 0.8	52.8 ± 0.7	39.4 ± 0.8	60.4 ± 1.1	52.0 ± 0.5	77.2 ± 0.3	51.8 ± 1.1	**78.8 ± 0.7**
Ours	**61.8 ± 0.7**	**82 ± 0.5**	**34.6 ± 0.3**	**58.4 ± 0.7**	**34.8 ± 0.5**	**57.8 ± 0.6**	**41.4 ± 0.5**	**62.8 ± 0.9**	**53.2 ± 0.6**	**78 ± 0.3**	**55.8 ± 0.7**	76.6 ± 0.8

Table 2: Comparison with previous coreset selection and dataset distillation methods on ImageNet Subset

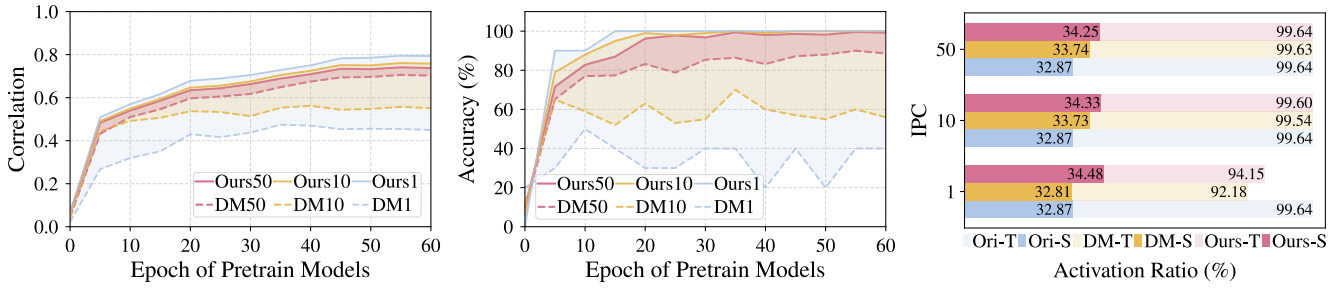
Comparison with Previous Methods

Performance Comparison We compare our method with various baseline approaches across different resolutions and scenarios. For coreset methods, baselines like Random, Herding (Welling 2009), and Forgetting (Toneva et al. 2018) are selected. Among OO-based methods, DC (Zhao, Mopuri, and Bilen 2021), DSA (Zhao and Bilen 2021), MTT (Cazenavette et al. 2022) and FTD (Du et al. 2023) are chosen for comparison. And for DM-based methods, CAFE (Wang et al. 2022), DM (Zhao and Bilen 2023), IDM (Zhao et al. 2023), M3D (Zhang et al. 2024b), DSDM (Li et al. 2024) and DANCE (Zhang et al. 2024a) are included. Table 1 and Table 2 present the comparative results of our method and other baselines across multiple benchmark datasets, such as SVHN, CIFAR-10, CIFAR-100, TinyImageNet, and subsets of ImageNet, demonstrating the effectiveness of our approach. To be Specific, in experiments on CIFAR-10, when IPC is 10 and 50, our method surpasses the classical DM algorithm by 23% and 13.5%, respectively, and it also outperforms OO-based algorithms such as DC and MTT. On the TinyImageNet dataset, our approach achieves remarkable accuracy of 29.3% at IPC-10 and 30.9% at IPC-50, both of which are higher than the current SOTA methods. Furthermore, on the ImageNet subset, our method demonstrates superior performance. For instance, a 5% gain improvement is achieved in ImageFruit at IPC-10.

Cross-Architecture Evaluation We evaluate the performance of synthetic datasets generated by our method across different network architectures. As shown in Table 3, we generate synthetic datasets using a 3-layer ConvNet and evaluate them on ResNet10 (He et al. 2016) and DenseNet121 (Huang et al. 2017). Our method demonstrates superior generalization performance across multiple architectures at IPC-10 and IPC-50.

Effectiveness of Our Method

Distribution Matching To validate the effectiveness of the proposed distribution matching method, we extract features from both the synthetic and original datasets using pre-



(a) Stage-wise Feature Correlation (b) Stage-wise Snapshot Accuracy (c) Neuron activation ratio

Figure 4: Effectiveness study of our method. (a) Distribution similarity via feature correlations. (b) Class separability evaluated by pretrained model accuracy across stages. (c) Information density from per-image and dataset-level neuron activation ratios.

(a) only constraint (b) only matching (c) Ours

Figure 5: Distribution of original images and synthetic images on CIFAR-10 with IPC-50.

trained models at different training stages and compute their similarity. Figure 4(a) illustrates that our method consistently achieves superior feature alignment across all training stages and configurations, demonstrating its improved performance in distribution matching. In contrast, Figure 5(a) shows that replacing the stage-wise feature extractors with randomly initialized networks leads to overly concentrated feature distributions, severely compromising data diversity.

Distribution Constraint To assess the impact of the proposed distribution constraint, we evaluate the synthetic datasets using different pretrained models under various configurations. Figure 4(b) shows that our method consistently achieves superior classification accuracy across all training stages, reflecting improved class separability and reduced overlap between categories in the synthesized data. In contrast, Figure 5(b) highlights substantial degradation when expert models are replaced with randomly initialized ones, dispersed feature representations lead to severe entanglement and poor class distinction.

Information Distillation We quantify information density through the ratio of neurons activated per image and across the entire dataset, where higher activation ratio indicates greater information density. Following previous work (Wang et al. 2024a), neurons with gradients exceeding layer-wise mean values are considered activated. Figure 4(c) demonstrates that at equivalent total activation levels, our method activates more neurons per image, confirming higher information density in individual synthetic samples.

Ablation Study

Effectiveness of Each Component We evaluate the impact of the main modules of the proposed method on performance, namely multiformation augmentation, stage-wise distribution matching, and stage-wise distribution constraints. As illustrated in Table 4, the proposed module achieves performance gains across multiple datasets at various IPC. Optimal performance is achieved when all modules are used in combination, indicating the effectiveness of the proposed method.

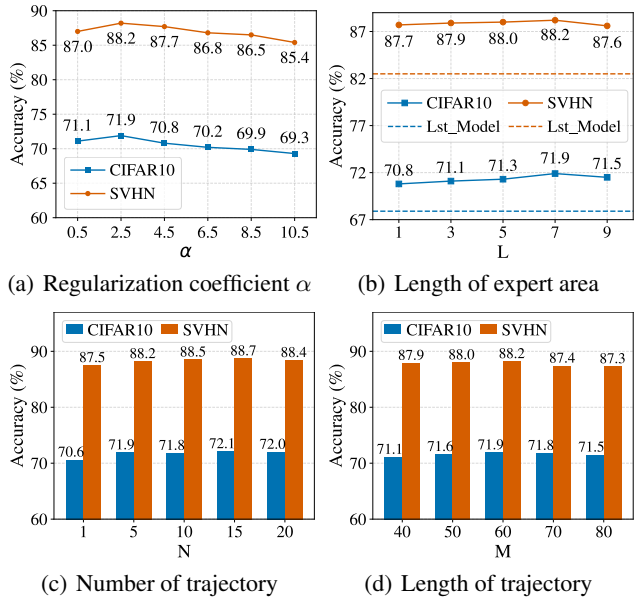
Impact of α The regularization coefficient α reflects the importance of expert distribution constraints in the loss. The figure 6(a) shows how performance varies with α. When α

Method	IPC	ConvNet-3	ResNet-10	DenseNet-121
DSA	10	52.1 ± 0.5	32.9 ± 0.3	34.5 ± 0.1
	50	60.6 ± 0.5	49.7 ± 0.4	49.1 ± 0.2
MTT	10	56.4 ± 0.7	34.5 ± 0.8	41.5 ± 0.5
	50	65.9 ± 0.6	43.2 ± 0.4	51.9 ± 0.3
DM	10	48.9 ± 0.6	42.3 ± 0.5	39.0 ± 0.1
	50	63.0 ± 0.4	58.6 ± 0.3	57.4 ± 0.3
M3D	10	63.5 ± 0.2	56.7 ± 0.3	54.6 ± 0.2
	50	69.9 ± 0.5	66.6 ± 0.3	66.1 ± 0.4
DANCE	10	70.8 ± 0.2	67.0 ± 0.2	64.5 ± 0.3
	50	76.1 ± 0.1	68.0 ± 0.1	64.8 ± 0.3
Ours	10	**71.9** ± **0.3**	**67.7** ± **0.2**	**68.2** ± **0.3**
	50	**76.5** ± **0.2**	**74.9** ± **0.4**	**74.3** ± **0.2**

Table 3: Accuracy on CIFAR-10 with different architectures. Synthetic dataset are condensed using ConvNet-3.

Aug	L_{MMD}	L_{SDC}	CIFAR-10		CIFAR-100	
			10	50	10	50
-	-	-	55.2 ± 0.2	65.3 ± 0.3	33.7 ± 0.2	44.5 ± 0.3
✓	-	-	63.2 ± 0.2	69.5 ± 0.3	40.5 ± 0.2	47.2 ± 0.3
✓	✓	-	65.8 ± 0.3	75.2 ± 0.2	47.0 ± 0.3	53.0 ± 0.2
✓	✓	✓	**71.9** ± **0.3**	**76.5** ± **0.2**	**51.3** ± **0.2**	**54.6** ± **0.3**

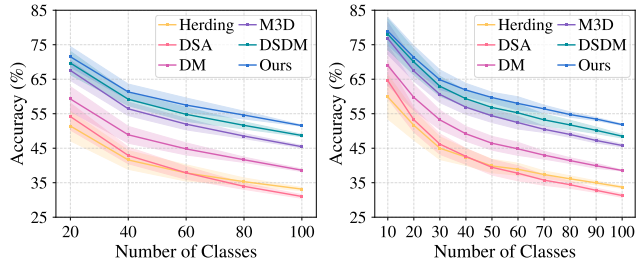
Table 4: Accuracy ablation study of each component on CIFAR-10/100 with IPC-10 and IPC-50.



(a) Regularization coefficient α (b) Length of expert area

(c) Number of trajectory (d) Length of trajectory

Figure 6: Effect of different hyperparameters with IPC-10.



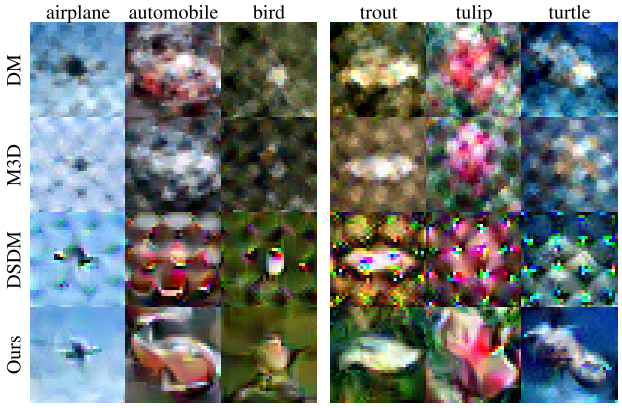
(a) step=5 (b) step=10

Figure 7: Class-incremental learning on CIFAR-100.

ranges from 0.5 to 10.5, the model's performance fluctuates within the range of 2.8%, indicating moderate model sensitivity to this hyperparameter.

Impact of Expert Region The expert region refers to a set of model snapshots within a range of length L starting from the feature extractor. During distillation, we randomly sample expert models from this region to impose distributional constraints. As shown in Figure 6(b), using the last converged model as the baseline, our expert region design demonstrates consistent performance across various L, and significantly outperforms the single-expert setting.

Impact of Expert Trajectory The expert trajectory, characterized by its number and length, affects both distribution matching and constraint. As illustrated in Figure 6(c), distillation performance improves slightly with an increasing number of trajectories, but gradually reaches saturation. Similarly, Figure 6(d) shows that extending trajectory length yields comparable trends in performance gain. To balance performance and efficiency, we refrain from excessively increasing the number or length of expert trajectories.



(a) CIFAR-10 (b) CIFAR-100

Figure 8: Visualization of synthetic images of CIFAR-10 (partial) and CIFAR-100 (partial) with different method.

Continual Learning

Continual learning aims to train models to adapt to new tasks while preserving performance on previous ones and minimizing catastrophic forgetting (Prabhu, Torr, and Dokania 2020). Dataset distillation, by reducing the volume of original dataset, shows great potential in this area. We compare Herding, DSA, DM, DSDM, M3D and our method. Following prior studies, we keep 20 images per class, set step sizes at 5 and 10, use a 3-layer ConvNet, and run each experiment 5 times to get mean and variance values. As demonstrated in Figure 7, our approach significantly outperforms baselines.

Visualization

To provide an intuitive qualitative assessment of the distilled images, we visualize the synthetic datasets generated by DM, M3D, DSDM and our method in Figure 8. It can be observed that: 1) The synthetic images produced by our method demonstrate superior structural detail preservation from the original dataset, with substantially reduced noise and enhanced visual clarity. 2) Our synthetic images possess more distinct category-discriminative features, thereby improving inter-class separability.

Conclusion

In this paper, we revisit the limitations of DM-based dataset distillation, emphasizing their reliance on static representations and the degradation in downstream performance. To address these issues, we propose Trajectory Guided Dataset Distillation (TGDD), a novel approach that dynamically aligns synthetic data with the evolving feature space of the training trajectory. By jointly applying stage-wise distribution matching and distribution constraints, TGDD improves both the diversity and representativeness of synthetic datasets. Extensive experiments on ten benchmarks demonstrate that TGDD delivers superior performance and efficiency, achieving up to a 5.0% accuracy gain on high-resolution benchmarks. In future work, we will explore how to extend dataset distillation to other modalities and tasks.

Acknowledgments

This work was supported in part by the National Science Foundation of China Joint Key Project under Grants U24B20173 and U24B20182, and by the National Natural Science Foundation of China under Grant 62376046, Grant 62536002, Grant 62561160098 and Grant 62402073. It was also supported by the Natural Science Foundation of Chongqing under Grant CSTB2023NSCQ-MSX0341, and by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJQN202300619.

References

Agarwal, S.; Arora, H.; Anand, S.; and Arora, C. 2020. Contextual diversity for active learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, 137–153. Springer.

Aljundi, R.; Lin, M.; Goujaud, B.; and Bengio, Y. 2019. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32.

Cazenavette, G.; Wang, T.; Torralba, A.; Efros, A. A.; and Zhu, J.-Y. 2022. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4750–4759.

Cazenavette, G.; Wang, T.; Torralba, A.; Efros, A. A.; and Zhu, J.-Y. 2023. Generalizing dataset distillation via deep generative prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3739–3748.

Chai, C.; Wang, J.; Tang, N.; Yuan, Y.; Liu, J.; Deng, Y.; and Wang, G. 2023. Efficient coreset selection with cluster-based methods. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 167–178.

Cui, J.; Wang, R.; Si, S.; and Hsieh, C.-J. 2023. Scaling up dataset distillation to imagenet-1k with constant memory. In *International Conference on Machine Learning*, 6565–6590. PMLR.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Dong, T.; Zhao, B.; and Lyu, L. 2022. Privacy for free: How does dataset condensation help privacy? In *International Conference on Machine Learning*, 5378–5396. PMLR.

Du, J.; Jiang, Y.; Tan, V. Y.; Zhou, J. T.; and Li, H. 2023. Minimizing the accumulated trajectory error to improve dataset distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3749–3758.

Feldman, D. 2020. Core-sets: Updated survey. *Sampling techniques for supervised or unsupervised tasks*, 23–44.

Gidaris, S.; and Komodakis, N. 2018. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4367–4375.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1): 723–773.

Guo, Z.; Wang, K.; Cazenavette, G.; Li, H.; Zhang, K.; and You, Y. 2024. Towards Lossless Dataset Distillation via Difficulty-Aligned Trajectory Matching. In *The Twelfth International Conference on Learning Representations*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Kim, J.-H.; Kim, J.; Oh, S. J.; Yun, S.; Song, H.; Jeong, J.; Ha, J.-W.; and Song, H. O. 2022. Dataset condensation via efficient synthetic-data parameterization. In *International Conference on Machine Learning*, 11102–11118. PMLR.

Kim, Y.; and Shin, B. 2022. In defense of core-set: A density-aware core-set selection for active learning. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 804–812.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90.

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7): 3.

Lee, H.; Kim, S.; Lee, J.; Yoo, J.; and Kwak, N. 2024. Coreset selection for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7682–7691.

Lei, S.; and Tao, D. 2023. A comprehensive survey of dataset distillation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1): 17–32.

Li, H.; Zhou, Y.; Gu, X.; Li, B.; and Wang, W. 2024. Diversified semantic distribution matching for dataset distillation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 7542–7550.

Nair, V.; and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A. Y.; et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, 4. Granada.

Pi, R.; Zhang, W.; Xie, Y.; Gao, J.; Wang, X.; Kim, S.; and Chen, Q. 2023. Dynafed: Tackling client data heterogeneity with global dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12177–12186.

Prabhu, A.; Torr, P. H.; and Dokania, P. K. 2020. Gdumb: A simple approach that questions our progress in continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 524–540. Springer.

Rahaman, N.; Baratin, A.; Arpit, D.; Draxler, F.; Lin, M.; Hamprecht, F.; Bengio, Y.; and Courville, A. 2019. On the spectral bias of neural networks. In *International conference on machine learning*, 5301–5310. PMLR.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2001–2010.

Sajedi, A.; Khaki, S.; Amjadian, E.; Liu, L. Z.; Lawryshyn, Y. A.; and Plataniotis, K. N. 2023. Datadam: Efficient dataset distillation with attention matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 17097–17107.

Schuhmann, C.; Beaumont, R.; Vencu, R.; Gordon, C.; Wightman, R.; Cherti, M.; Coombes, T.; Katta, A.; Mullis, C.; Wortsman, M.; et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35: 25278–25294.

Sener, O.; and Savarese, S. 2017. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*.

Such, F. P.; Rawal, A.; Lehman, J.; Stanley, K.; and Clune, J. 2020. Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data. In *International Conference on Machine Learning*, 9206–9216. PMLR.

Toneva, M.; Sordoni, A.; Combes, R. T. d.; Trischler, A.; Bengio, Y.; and Gordon, G. J. 2018. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.

Wang, J.; Di, S.; Chen, L.; and Ng, C. W. W. 2024a. Learning from emergence: A study on proactively inhibiting the monosemantic neurons of artificial neural networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3092–3103.

Wang, K.; Zhao, B.; Peng, X.; Zhu, Z.; Yang, S.; Wang, S.; Huang, G.; Bilen, H.; Wang, X.; and You, Y. 2022. Cafe: Learning to condense dataset by aligning features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12196–12205.

Wang, T.; Zhu, J.-Y.; Torralba, A.; and Efros, A. A. 2018. Dataset distillation. *arXiv preprint arXiv:1811.10959*.

Wang, Y.; Fu, H.; Kanagavelu, R.; Wei, Q.; Liu, Y.; and Goh, R. S. M. 2024b. An aggregation-free federated learning for tackling data heterogeneity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26233–26242.

Welling, M. 2009. Herding dynamical weights to learn. In *Proceedings of the 26th annual international conference on machine learning*, 1121–1128.

Yang, E.; Shen, L.; Wang, Z.; Liu, T.; and Guo, G. 2023. An efficient dataset condensation plugin and its application to continual learning. *Advances in Neural Information Processing Systems*, 36.

Yang, S.; Cheng, S.; Hong, M.; Fan, H.; Wei, X.; and Liu, S. 2024. Neural spectral decomposition for dataset distillation. In *European Conference on Computer Vision*, 275–290. Springer.

Yu, R.; Liu, S.; and Wang, X. 2023. Dataset distillation: A comprehensive review. *IEEE transactions on pattern analysis and machine intelligence*, 46(1): 150–170.

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.

Zhang, H.; Li, S.; Lin, F.; Wang, W.; Qian, Z.; and Ge, S. 2024a. DANCE: dual-view distribution alignment for dataset condensation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 1679–1687.

Zhang, H.; Li, S.; Wang, P.; Zeng, D.; and Ge, S. 2024b. M3d: Dataset condensation by minimizing maximum mean discrepancy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 9314–9322.

Zhao, B.; and Bilen, H. 2021. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, 12674–12685. PMLR.

Zhao, B.; and Bilen, H. 2023. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 6514–6523.

Zhao, B.; Mopuri, K. R.; and Bilen, H. 2021. Dataset Condensation with Gradient Matching. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Zhao, G.; Li, G.; Qin, Y.; and Yu, Y. 2023. Improved distribution matching for dataset condensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7856–7865.