Dataset Distillation via the Wasserstein Metric

Haoyang Liu[1], Yijiang Li[2], Tiancheng Xing[3], Peiran Wang[4], Vibhu Dalal[5], Luwei Li[1]
Jingrui He[1], Haohan Wang[1]

[1]University of Illinois at Urbana-Champaign
[2]University of California San Diego
[3]National University of Singapore
[4]University of California, Los Angeles
[5]Sri Aurobindo International Centre of Education
{hl57, haohanw}@illinois.edu

Figure 1. Synthetic images distilled from ImageNet-1K using our WMDD method with ResNet-18, capturing essential class features aligned with human perception. We randomly sampled one image for each of the chosen categories from our output in the 10 IPC setting.

Abstract

Dataset Distillation (DD) aims to generate a compact synthetic dataset that enables models to achieve performance comparable to training on the full large dataset, significantly reducing computational costs. Drawing from optimal transport theory, we introduce WMDD (Wasserstein Metric-based Dataset Distillation), a straightforward yet powerful method that employs the Wasserstein metric to enhance distribution matching.

We compute the Wasserstein barycenter of features from a pretrained classifier to capture essential characteristics of the original data distribution. By optimizing synthetic data to align with this barycenter in feature space and leveraging per-class BatchNorm statistics to preserve intra-class variations, WMDD maintains the efficiency of distribution matching approaches while achieving state-of-the-art results across various high-resolution datasets. Our extensive experiments demonstrate WMDD's effectiveness and adaptability, highlighting its potential for advancing machine learning applications at scale. Code is available at https://github.com/Liu-Hy/WMDD *and website at* https://liu-hy.github.io/WMDD/.

1 Introduction

Dataset distillation [46, 61] aims to create compact synthetic datasets that train models to perform similarly to those trained on full-sized original datasets. This technique promises to address the escalating computational

costs associated with growing data volumes, enables efficient model development across various applications [16, 27, 28, 34, 40, 56], and helps mitigate bias [7, 49], robustness [50] and privacy [11, 38] concerns in training data.

The central challenge in dataset distillation lies in capturing the distributional characteristics of an entire dataset within a small set of synthetic samples [25, 34]. Existing methods often struggle to balance computational efficiency with distillation quality. Some researchers formulate dataset distillation as a bi-level optimization problem [30, 33, 48], which has inspired innovative approaches such as gradient matching [57, 61], trajectory matching [3], and curvature matching [39]. These methods align the optimization dynamics between models trained on synthetic and original datasets. However, they typically require second-order derivative computation, becoming prohibitively expensive for large datasets like ImageNet-1K [9]. Alternative approaches directly align synthetic and original data distributions using metrics like Maximum Mean Discrepancy (MMD) [18, 42]. Despite their computational efficiency, these methods typically underperform compared to optimization-based approaches [25, 34]. We conjecture that this performance gap is due to MMD's limitations in quantifying distributional differences in ways that provide meaningful signals for generating effective synthetic images.

In this paper, we introduce the Wasserstein distance as an effective measure of distributional differences for Dataset Distillation. Wasserstein distance is known for comparing distributions by quantifying the minimal movement required to transform one probability distribution into another within a given metric space [44]. Grounded in Optimal Transport theory [22], it provides a geometrically meaningful approach to quantifying differences between distributions. The Wasserstein barycenter [1] represents the centroid of multiple distributions while preserving their essential characteristics. Fig. 2 illustrates this advantage by simulating distributions spread on circles and crosses on a 2D plane (Fig. 2a), and their barycenters computed with different distribution metrics. While KL divergence (Fig. 2b) and MMD (Fig. 2c) barycenters produce a rigid mix-up of input distributions, the Wasserstein barycenter (Fig. 2d) creates a natural interpolation that preserves the structural characteristics of the original distributions.

Motivated by these advantages, we develop a straightforward yet effective DD method using Wasserstein distance for distribution matching. Unlike prior work using MMD [18, 42], the Wasserstein barycenter [1] avoids reliance on heuristically designed kernels and naturally accounts for distribution geometry and structure. This allows us to statistically summarize real datasets within a fixed number of representative and diverse synthetic images that enable classification models to achieve higher performance.

Furthermore, to address challenges in optimizing high-dimensional data for DD, we present WMDD (**W**asserstein **M**etric-based **D**ataset **D**istillation), an algorithm that balances performance and computational feasibility on large datasets. We embed synthetic data into the feature space of a pre-trained image classifier following [53, 60, 62], and use the Wasserstein barycenter as a compact summary of intra-class data distribution. To leverage prior knowledge in pretrained models, we propose a regularization method using Per-Class BatchNorm statistics (PCBN) for more precise distribution matching, inspired by previous work addressing data heterogeneity [17] and long-tail problems [5] with variants of batch normalization [21]. By implementing an efficient algorithm [8] for Wasserstein barycenter computation, our method maintains the efficiency of distribution matching-based approaches [60] and can scale to large, high-resolution datasets like ImageNet-1K [9]. Our experiments demonstrate that WMDD achieves state-of-the-art performance across various benchmarks. Our contributions include:

- A novel dataset distillation technique that integrates distribution matching with Wasserstein metrics, bridging dataset distillation with insights from optimal transport theory.
- A balanced solution leveraging the computational feasibility of distribution-matching based methods to ensure scalability to large datasets.
- Comprehensive experimental results across diverse high-resolution datasets demonstrating significant performance improvements over existing methods, highlighting our approach's practical applicability in the big data era.

2 Related work

2.1 Data Distillation

Dataset Distillation (DD) aims to create compact synthetic training sets that enable models to achieve performance comparable to those trained on larger original datasets [47]. Current DD methods fall into three major categories [54]: *Performance Matching* seeks to minimize loss of the synthetic dataset by aligning the performance of models trained on synthetic and original datasets, methods include DD [47], FRePo [64], AddMem [10], KIP [33], RFAD [30]; *Parameter Matching* is an approach to train two neural networks on the real and synthetic datasets respectively, with the aim to promote similarity in their parameters, methods include DC [61], DSA [57], MTT [3], HaBa [29], FTD [13], TESLA [6]; *Distribution Matching* aims to obtain synthetic data that closely matches the distribution of real data, methods include DM [60], IT-GAN [59], KFS [24], CAFE [45], SRe^{2}L [53], IDM [62], G-VBSM [36], and SCDD [36].

2.2 Distribution Matching

Distribution Matching (DM) techniques, initially proposed in [58], aim to directly align the probability distributions

(a) (b) KL divergence (c) MMD distance (d) Wasserstein distance

Figure 2. The capability of Wasserstein barycenter in condensing the core characteristics of distributions: (a) distributions defined on $\mathbb{R}^2$, concentrated on outlines of circles (blue) and crosses (green). Barycenters computed using: (b) KL divergence, (c) Maximum Mean Discrepancy (MMD), which operates in a kernel-induced feature space, and (d) Wasserstein distance, which preserves geometric structure through optimal transport. Color intensity represents probability density, while color hue shows different types of source distributions.

of the original and synthetic datasets [15, 34]. The fundamental premise underlying these methods is that when two datasets exhibit similarity based on a specific distribution divergence metric, they lead to comparably trained models [26]. DM typically employs parametric encoders for projecting data onto a lower dimensional latent space and approximates the Maximum Mean Discrepancy for assessing distribution mismatch [41, 45, 53, 55, 58, 62]. Notably, DM avoids reliance on model parameters and bi-level optimization, diverging from gradient and trajectory matching approaches. This distinction reduces memory requirements. However, the empirical evidence so far suggests that DM may underperform compared to the other approaches [26, 55].

3 Preliminaries

We introduce the fundamental concepts of Dataset Distillation and Wasserstein barycenters that form the foundation of our approach.

3.1 Dataset Distillation

Notations Let $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be the real training set that contains n distinct input–label pairs and let $\mu_\mathcal{T}$ be its empirical distribution, i.e. $\mathbf{x}_i \sim \mu_\mathcal{T}$. Similarly, let $\mathcal{S} = \{(\tilde{\mathbf{x}}_j, \tilde{y}_j)\}_{j=1}^m$ be the synthetic set with *at most* m distinct pairs and empirical distribution $\mu_\mathcal{S}$. Each data point lies in an ambient space $\Omega = \mathbb{R}^d$. Denote by $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\tilde{\mathbf{X}} \in \mathbb{R}^{m\times d}$ the matrices that stack the unique positions in $\mathcal{T}$ and $\mathcal{S}$, respectively. The probability mass associated with the synthetic samples is stored in the weight vector $\mathbf{w} \in \Delta^{m-1}$, where w_j is the weight of $\tilde{\mathbf{x}}_j$ and Δ^{m-1} is the $(m-1)$–simplex. Consequently, we can compactly write the synthetic dataset as the tuple $\mathcal{S} = (\tilde{\mathbf{X}}, \mathbf{w})$. Throughout, $\ell(\mathbf{x}, y; \boldsymbol{\theta})$ denotes the loss incurred by a model with parameters $\boldsymbol{\theta}$ on a single sample $(\mathbf{x}, y)$.

Dataset Distillation (DD) aims at finding the optimal synthetic set $\mathcal{S}^*$ for a given $\mathcal{T}$ by solving a bi-level optimization problem as below:

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mu_\mathcal{T}} \ell\left(\mathbf{x}, y; \boldsymbol{\theta}(\mathcal{S})\right) \tag{1}$$

$$\text{subject to } \boldsymbol{\theta}(\mathcal{S}) = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{m} \ell(\tilde{\mathbf{x}}_i, \tilde{y}_i; \boldsymbol{\theta}). \tag{2}$$

Directly solving the bi-level optimization problem poses significant challenges. As a viable alternative, a prevalent approach [35, 45, 53, 60] seeks to align the distribution of the synthetic dataset with that of the real dataset. This strategy is based on the assumption that *the optimal synthetic dataset should be the one that is distributionally closest to the real dataset subject to a fixed number of synthetic data points*. We label this as **Assumption A1**. While recent methods [45, 59, 60] grounded on this premise have shown promising empirical results, they often struggle to balance strong performance with scalability to large datasets like ImageNet-1K.

3.2 Wasserstein barycenters

Our method computes representative features using Wasserstein barycenters [1], extending the concept of "averaging" to distributions while respecting their geometric properties. This approach relies on the Wasserstein distance to quantify distributional differences.

Definition 1 (Wasserstein distance). Let (Ω, D) be a metric space and denote by $P(\Omega)$ the set of Borel probability measures on Ω. For $\mu, \nu \in P(\Omega)$ the p-Wasserstein distance is

$$W_p(\mu,\nu) := \left(\inf_{\pi\in\Pi(\mu,\nu)} \int_{\Omega^2} D(x,y)^p \,\mathrm{d}\pi(x,y)\right)^{1/p}, \tag{3}$$

where $\Pi(\mu,\nu)$ is the set of couplings (joint distributions with the prescribed marginals). Intuitively, W_p measures the minimum "work"—mass times distance—required to morph μ into ν; hence it is also known as the earth–mover distance.

Definition 2 (Wasserstein barycenter). Given N distributions $\{\nu_i\}_{i=1}^N \subseteq P(\Omega)$, their p-Wasserstein barycenter is

any solution of

$$\arg\min_{\mu \in P(\Omega)} f(\mu) := \frac{1}{N} \sum_{i=1}^{N} W_p^p(\mu, \nu_i). \tag{4}$$

The barycenter can be viewed as the "center of mass" of the input distributions: it minimizes the average transportation cost (squared when p=2) to all u_i.

4 Method

The Wasserstein distance offers an intuitive and geometrically meaningful way to quantify differences between distributions, as demonstrated by its superior performance in preserving structural characteristics (Fig. 2). We leverage these strengths to bridge the performance gap in dataset distillation and potentially surpass current state-of-the-art techniques. This section establishes the connection between Wasserstein barycenters and dataset distillation, presents the efficient computation approach, and introduces our complete method design.

4.1 Wasserstein barycenter in dataset distillation

We begin by representing both real and synthetic datasets as empirical distributions. For the real dataset $\mathcal{T}$, assuming no prior knowledge and no repetitive samples, we adopt a discrete uniform distribution over the observed data points, $\mu_{\mathcal{T}} = \frac{1}{n} \sum_{i=1}^{n} \delta_{\mathbf{x}_i}$, where $\delta_{\mathbf{x}_i}$ represents the Dirac delta function centered at position $\mathbf{x}_i$. This function is zero everywhere except at $\mathbf{x}_i$ and integrates to one.

For the synthetic dataset $\mathcal{S}$, we define its empirical distribution as: $\mu_{\mathcal{S}} = \sum_{j=1}^{m} w_j \delta_{\tilde{\mathbf{x}}_j}$, where the weights satisfy $w_j \geq 0$ and $\sum_{j=1}^{m} w_j = 1$. Learning these probabilities provides additional flexibility in approximating the real distribution.

Following Assumption A1 and our choice of the Wasserstein metric, the optimal synthetic dataset $\mathcal{S}^*$ should generate an empirical distribution that minimizes the Wasserstein distance to the real data distribution:

$$\mu_{\mathcal{S}^*} = \mu_{\mathcal{S}}^* = \arg\min_{\mu_{\mathcal{S}} \in P_m} W_p^p(\mu_{\mathcal{S}}, \mu_{\mathcal{T}}), \tag{5}$$

where $\mu_{\mathcal{S}^*}$ is the empirical distribution of the optimal dataset, $\mu_{\mathcal{S}}^*$ is the optimal empirical distribution, and $P_m \subset P(\Omega)$ denotes the set of distributions supported on at most m atoms in $\mathbb{R}^d$. This is a special case of (4) with $N = 1$. Since the synthetic set $\mathcal{S}$ is fully specified by positions $\tilde{\mathbf{X}}$ and weights $\mathbf{w}$, we can find the optimal set $\mathcal{S}^*$ by minimizing the below function:

$$f(\tilde{\mathbf{X}}, \mathbf{w}) := W_p^p(\mu_{\mathcal{S}}, \mu_{\mathcal{T}}). \tag{6}$$

4.2 Computing the Wasserstein barycenter

To efficiently optimize $f(\tilde{\mathbf{X}}, \mathbf{w})$, we adapt the barycenter computation method from [8], employing an alternating optimization approach that iterates between optimizing weights and positions. This approach leverages the convex structure of the optimal transport problem to ensure computational efficiency.

Weight optimization with fixed positions With fixed synthetic data positions $\tilde{\mathbf{X}}$, we first construct a cost matrix $\mathbf{C} \in \mathbb{R}^{n \times m}$ where each $c_{ij} = \|\tilde{\mathbf{x}}_j - \mathbf{x}_i\|^2$ represents the squared Euclidean distance between points in the two distributions. The Wasserstein distance calculation transforms into finding the optimal transport plan $\mathbf{T} \in \mathbb{R}^{n \times m}$, where each t_{ij} represents the mass moved from position i to position j:

$$\min_{\mathbf{T}} \langle \mathbf{C}, \mathbf{T} \rangle_F \quad \text{subject to} \quad \sum_{j=1}^{m} t_{ij} = \frac{1}{n}, \ \forall i, \tag{7}$$

$$\sum_{i=1}^{n} t_{ij} = w_j, \ \forall j, \quad t_{ij} \geq 0, \ \forall i, j, \tag{8}$$

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product. The dual formulation introduces variables α_i and β_j that correspond to the marginal constraints:

$$\max_{\alpha, \beta} \left(\sum_{i=1}^{n} \frac{\alpha_i}{n} + \sum_{j=1}^{m} w_j \beta_j \right) \tag{9}$$

$$\text{subject to} \quad \alpha_i + \beta_j \leq c_{ij}, \forall i, j. \tag{10}$$

Through strong duality [2], the optimal dual variables β_j provide the subgradient of the objective with respect to $\mathbf{w}$. This elegant property allows us to efficiently optimize weights using projected subgradient descent, guiding mass toward locations that minimize transportation cost.

Position optimization with fixed weights With $\mathbf{w}$ fixed, the objective is quadratic in each $\tilde{\mathbf{x}}_j$; its (classical) Hessian is $\nabla^2_{\tilde{\mathbf{x}}_j} f = 2w_j \mathbf{I}$. Performing one Newton step therefore amounts to

$$\tilde{\mathbf{x}}_j \leftarrow \tilde{\mathbf{x}}_j - \frac{1}{w_j} \sum_{i=1}^{n} t_{ij} (\tilde{\mathbf{x}}_j - \mathbf{x}_i). \tag{11}$$

Intuitively, this update pulls each synthetic point toward real data points based on the optimal transport plan, with the "pull strength" weighted by the transport allocation. Points with higher transport allocation exert stronger influence on the synthetic positions.

By alternating between these two optimization steps, we converge to a local optimum that represents the Wasserstein barycenter of the real data distribution. Remarkably, we find that even a small number of iterations produces high-quality synthetic data. Further details on this method are available in Appendix C.

4.3 Barycenter Matching in the Feature Space

Our above discussion shows that dataset distillation can be cast as the problem of finding the barycenter of the real data

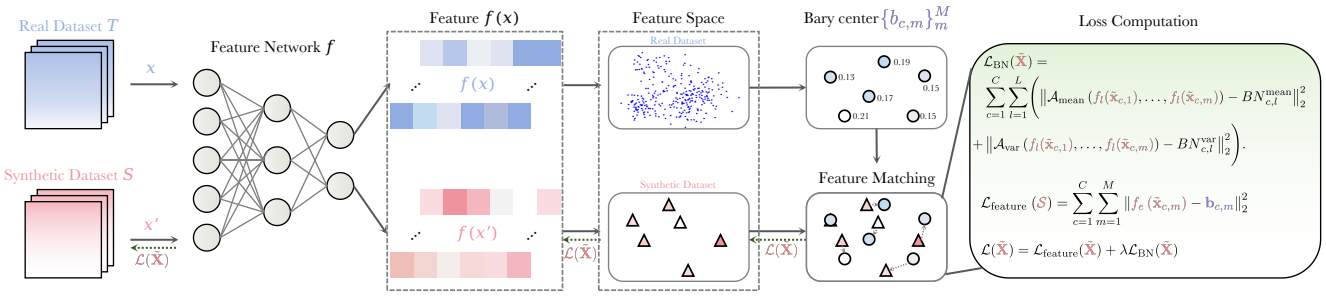


Figure 3. Diagram of our WMDD method. Real dataset T and synthetic dataset S pass through the feature network f to obtain features. The features of the real dataset are used to compute the Wasserstein Barycenter. The synthetic dataset is optimized via feature matching and loss computation (combining feature loss and BN regularization) to align with the Barycenter, generating high-quality synthetic data for efficient model training.

Algorithm 1: Wasserstein Metric-based Dataset Distillation (WMDD)

Input: Real dataset $\mathcal{T} = \{\mathbf{x}_{k,i}\}_{i=1,\dots,n_k}^{k=1,\dots,g}$, teacher model f with feature extractor f_e (before the linear classifier), number of iterations K

Train model f on $\mathcal{T}$;
for *each class* k **do**
  for *each sample* i **do**
    Perform forward pass: $f(\mathbf{x}_{k,i})$;
    Store feature: $f_e(\mathbf{x}_{k,i})$;
  Compute $\mathrm{BN}_{k,l}^{\mathrm{mean}}$, $\mathrm{BN}_{k,l}^{\mathrm{var}}$;
for *each class* k **do**
  $\{\mathbf{b}_{k,j}\}_{j=1,\dots,m_k}, \{\mathbf{w}_{k,j}\}_{j=1,\dots,m_k} \leftarrow$ barycenter $(\{f_e(\mathbf{x}_{k,i})\}_{i=1,\dots,n_k})$, according to Algorithm 2 (in Appendix D) with K iterations;
  Optimize $\{\tilde{\mathbf{x}}_{k,j}\}_{j=1,\dots,m_k}$ according to Eq. 15;

Output: Synthetic dataset $\mathcal{S}$ with positions $\{\tilde{\mathbf{x}}_{k,j}\}_{j=1,\dots,m_k}^{k=1,\dots,g}$ and weights $\{\mathbf{w}_{k,j}\}_{j=1,\dots,m_k}^{k=1,\dots,g}$.

distribution, and there is an efficient approach for computing this barycenter. However, for high dimensional data such as images, it is beneficial to use some prior to learning synthetic images that encode meaningful information from the real dataset. Inspired by recent works [51, 53], we use a pretrained classifier to embed the images into the feature space, in which we compute the Wasserstein barycenter to learn synthetic images. This subsection details our concrete algorithm design, which is illustrated in Fig. 3, and summarized in Algorithm 1.

Suppose the real dataset $\mathcal{T}$ has g classes, with n_k images for class k (hence $n = \sum_{k=1}^{g} n_k$). Let us re-index the samples by classes and denote the training set as $\mathcal{T} = \{\mathbf{x}_{k,i}\}_{i=1,\dots,n_k}^{k=1,\dots,g}$. Suppose that we want to distill m_k images for class k. Denote the synthetic set $\mathcal{S} = \{\tilde{\mathbf{x}}_{k,j}\}_{j=1,\dots,m_k}^{k=1,\dots,g}$, where $m_k \ll n_k$ for all k.

First, we employ the pretrained model to extract features for all samples within each class in the original dataset $\mathcal{T}$. More specifically, we use the pretrained model f to obtain the feature set $\{f_e(\mathbf{x}_{k,i})\}_{i=1,\dots,n_k}$ for each class k, where $f_e(\cdot)$ returns the representation immediately before the linear classifier.

Next, we compute the Wasserstein barycenter for each feature set computed in the previous step. We treat the feature set for each class as an empirical distribution, and adapt the algorithm in [8] to compute the free support barycenters with m_k points for class k, denoted as $\{\mathbf{b}_{k,j}\}_{j=1,\dots,m_k}$, and the associated weights $\{\mathbf{w}_{k,j}\}_{j=1,\dots,m_k}$, which are used to weight the synthetic images.

Then, in the main distillation process, we use iterative gradient descent to learn the positions of synthetic images by jointly considering two objectives. We match the features of the synthetic images with the corresponding data points in the learned barycenter:

$$\mathcal{L}_{\text{feature}}(\tilde{\mathbf{X}}) = \sum_{k=1}^{g} \sum_{j=1}^{m_k} \| f_e(\tilde{\mathbf{x}}_{k,j}) - \mathbf{b}_{k,j} \|_2^2, \qquad (12)$$

where $f_e(\cdot)$ is the function to compute features of the last layer.

To further leverage the capability of the pretrained model in aligning the distributions, previous DD works [51, 53] have used BatchNorm statistics of the real data to regularize synthetic images. However, the gradient on each synthetic sample for optimizing global BN alignment in a batch of mixed classes may not synergize well with the gradient on the same sample for matching its class-specific objective like the CE loss. Intuitively, the BN statistics within different data classes may vary, and simply encouraging alignment of global BN statistics does not provide enough information about how synthetic samples from different classes should contribute differently to the global BN statistics, potentially leading to suboptimal distillation quality.

Methods	ImageNette				Tiny ImageNet				ImageNet-1K				ImageNet-21K	
	1	10	50	100	1	10	50	100	1	10	50	100	10	20
Random [60]	23.5 ± 4.8	47.7 ± 2.4	-	-	1.5 ± 0.1	6.0 ± 0.8	16.8 ± 1.8	-	0.5 ± 0.1	3.6 ± 0.1	15.3 ± 2.3	-	-	-
DM [60]	32.8 ± 0.5	58.1 ± 0.3	-	-	3.9 ± 0.2	12.9 ± 0.4	24.1 ± 0.3	-	1.5 ± 0.1	-	-	-	-	-
MTT [3]	**47.7** ± 0.9	63.0 ± 1.3	-	-	**8.8** ± 0.3	23.2 ± 0.2	28.0 ± 0.3	-	-	-	-	-	-	-
DataDAM [35]	34.7 ± 0.9	59.4 ± 0.4	-	-	8.3 ± 0.4	18.7 ± 0.3	28.7 ± 0.3	-	2.0 ± 0.1	6.3 ± 0.0	15.5 ± 0.2	-	-	-
SRe^{2}L [53]	20.6† ± 0.3	54.2† ± 0.4	80.4† ± 0.4	85.9† ± 0.2	-	-	41.1 ± 0.4	49.7 ± 0.3	-	21.3 ± 0.6	46.8 ± 0.2	52.8 ± 0.4	18.5 ± 0.2	21.8 ± 0.1
CDA‡ [52]	-	-	-	-	-	-	48.7	53.2	-	-	53.5	58.0	22.6 ± 0.2	26.4 ± 0.1
G-VBSM [36]	-	-	-	-	-	-	47.6 ± 0.3	51.0 ± 0.4	-	31.4 ± 0.5	51.8 ± 0.4	55.7 ± 0.4	-	-
SCDD [63]	-	-	-	-	-	31.6 ± 0.1	45.9 ± 0.2	-	-	32.1 ± 0.2	53.1 ± 0.1	57.9 ± 0.1	-	-
WMDD	40.2 ± 0.6	**64.8** ± 0.4	**83.5** ± 0.3	**87.1** ± 0.3	7.6 ± 0.2	**41.8** ± 0.1	**59.4** ± 0.5	**61.0** ± 0.3	**3.2** ± 0.3	**38.2** ± 0.2	**57.6** ± 0.5	**60.7** ± 0.2	**24.5** ± 0.1	**29.3** ± 0.2

Table 1. Comparison of various dataset distillation methods. We used the reported results for prior methods when available. We replicated the result of SRe^{2}L on the ImageNette dataset, marked by †. Results of CDA did not include error bars, and the row is marked by ‡.

Thus, to better capture the intra-class data distribution, we propose the Per-Class BatchNorm (PCBN) regularization method, using BatchNorm statistics of the real data *within each class separately* to regularize synthetic data. While conceptually similar to previous BatchNorm variants for feature distribution heterogeneity [17] and long-tail problems [5], it is fundamentally different in technical design. Specifically, we regularize synthetic images with

$$\mathcal{L}_{\text{BN}}(\tilde{\mathbf{X}}) = \sum_{k=1}^{g}\sum_{l=1}^{L}\Big(\|\mathcal{A}_{\text{mean}}(\{f_l(\tilde{\mathbf{x}}_{k,j})\}_{j=1}^{m_k}, \{w_{k,j}\}_{j=1}^{m_k}) - \text{BN}_{k,l}^{\text{mean}}\|_2^2 + \|\mathcal{A}_{\text{var}}(\{f_l(\tilde{\mathbf{x}}_{k,j})\}_{j=1}^{m_k}, \{w_{k,j}\}_{j=1}^{m_k}) - \text{BN}_{k,l}^{\text{var}}\|_2^2\Big). \tag{13}$$

Here, L is the number of BatchNorm layers, and $f_l(\cdot)$ is the function that computes the feature map that feeds the l-th BatchNorm layer. $BN_{k,l}^{\text{mean}}$ and $BN_{k,l}^{\text{var}}$ denote the per-channel mean and variance of class k, obtained from one pass over the real data. The weighted aggregate operators $\mathcal{A}_{\text{mean}}$ and $\mathcal{A}_{\text{var}}$ compute statistics of synthetic samples while respecting the optimal transport weights. For feature tensor $\mathbf{F}$ with spatial dimensions $H \times U$ (height and width), these operators compute channel-wise statistics:

$$\mathcal{A}_{\text{mean}}(\mathbf{F}, \mathbf{w})_c := \frac{1}{HU\sum_{j=1}^{m_k} w_{k,j}} \sum_{j=1}^{m_k} w_{k,j} \sum_{h=1}^{H}\sum_{u=1}^{U} F_{j,c,h,u},$$

$$\mathcal{A}_{\text{var}}(\mathbf{F}, \mathbf{w})_c := \frac{1}{HU\sum_{j=1}^{m_k} w_{k,j}} \sum_{j=1}^{m_k} w_{k,j} \sum_{h=1}^{H}\sum_{u=1}^{U} \big(F_{j,c,h,u} - \mathcal{A}_{\text{mean}}(\mathbf{F}, \mathbf{w})_c\big)^2. \tag{14}$$

Here, $F_{j,c,h,u}$ denotes the activation at position (h, u) in channel c for synthetic sample j. Each expression computes statistics for channel c; concatenating across all channels yields the complete mean and variance vectors.

Combining these objectives above, we employ the below loss function for learning the synthetic data:

$$\mathcal{L}(\tilde{\mathbf{X}}) = \mathcal{L}_{\text{feature}}(\tilde{\mathbf{X}}) + \lambda \mathcal{L}_{\text{BN}}(\tilde{\mathbf{X}}), \tag{15}$$

where λ is a regularization coefficient. The synthetic set $\mathcal{S}$ therefore comprises the positions $\tilde{\mathbf{X}}$ and their associated weights $\{w_{k,j}\}_{j=1}^{m_k}$, which are used in the FKD stage following previous DD works [36, 51, 53].

5 Experiments

5.1 Experiment Setup

We systematically evaluated our method on three high-resolution datasets: ImageNette [20], Tiny ImageNet [23], and ImageNet-1K [9]. We tested synthetic image budgets of 1, 10, 50, and 100 images per class (IPC). For each dataset, we trained a ResNet-18 model [19] on the real training set, distilled the dataset using our method, then trained a ResNet-18 model from scratch on the synthetic data. We measured performance using the top-1 accuracy of the trained model on the validation set. Results report the mean and standard deviation from 3 repeated runs. Our barycenter algorithm implementation used the Python Optimal Transport library [14]. We maintained most hyperparameter settings from [53] but adjusted our loss terms' regularization coefficient λ. For barycenter computation (Algorithm 1), we found $K = 10$ iterations sufficient for high-performance synthetic data generation. Increasing K yielded only marginal improvements, so we kept $K = 10$ to balance efficiency and performance. We provide full implementation details in Appendix E.

5.2 Comparison with Other Methods

With this experimental setup, we now evaluate how our Wasserstein metric-based approach performs against existing dataset distillation methods.

We compared our method against several baselines and recent strong dataset distillation (DD) approaches, including distribution matching-based methods like DataDAM [35], SRe^{2}L [53], CDA [52], G-VBSM [36], and SCDD [63], selected for their scalability to large, high-resolution datasets. Table 1 presents our experimental results alongside reported results from these methods under identical settings. Our method consistently achieved state-of-the-art performance in most settings across different datasets. Compared to MTT [3] and DataDAM [35], which show good performance in fewer IPC settings, the performance of our method increases more rapidly with the number of synthetic images. Notably, in the 100 IPC setting, our method achieved top-1 accuracies of 87.1%, 61.0%, and

60.7% across the three datasets, respectively. These results approach those of pretrained classifiers (89.9%, 63.5%, and 63.1%) trained on full datasets. This superior performance highlights the effectiveness and robustness of our approach in achieving higher accuracy across different datasets. Our method is scalable to even larger datasets. We demonstrate the scalability of WMDD on ImageNet-21K with 10 and 20 IPC, where our method consistently outperforms CDA and SRe^{2}L by a large margin.

5.3 Cross-architecture Generalization

Beyond achieving strong performance on the distillation architecture, a critical test for any dataset distillation method is how well the synthetic data generalizes to different model architectures [4, 25]. For this aim, we conducted experiments training various randomly initialized models on synthetic data generated via our ResNet-18-based method. To prevent overfitting on the small synthetic data while ensuring fair comparison, we held out 20% of the distilled data as validation set to find the best training epoch for each experiment. We report the performance of different evaluation models, ResNet-18, ResNet-50, ResNet-101 [19], ViT-Tiny, and ViT-Small [12], in the 50 IPC setting on ImageNet-1K. The results in Table 2 show that our method demonstrates stronger cross-architecture transfer than previous methods. Our synthetic data generalizes well across the ResNet family, where the performance increases with the model capacity. The performance on the vision transformers is relatively lower, probably due to their data-hungry property.

Method	Res18	Res50	Res101	ViT-T	ViT-S
SRe^{2}L	48.02	55.61	60.86	16.56	15.75
CDA	54.43	60.79	61.74	31.22	32.97
G-VBSM	52.28	59.08	59.30	30.30	30.83
WMDD (Ours)	**57.83**	**61.22**	**62.57**	**34.25**	**34.87**

Table 2. Cross-architecture generalization performance on ImageNet-1K in 50 IPC setting. We used ResNet-18 for distillation and different architectures for evaluation: ResNet-{18,50,101}, ViT-Tiny and ViT-Small with a patch size of 16.

5.4 Ablation Study

To understand the individual contributions of our key design choices, we conducted an ablation study examining which factors drive our method's improved performance. We examined two key factors: whether to use our Wasserstein barycenter loss (Eq. 12) or the cross-entropy loss [52, 53] for feature matching; and whether to use standard BatchNorm statistics or our PCBN method for regularization. We evaluated these factors across different datasets using the 10 IPC setting, with results shown in Table 3. As discussed in our method design (Section 4.3), standard BN computes statistics from all-class samples, which does not synergize well with the class-specific matching objective, leading to mixed results with the Wasserstein loss. In contrast, our PCBN method significantly improves performance on all datasets by capturing intra-class distributions. When properly paired with PCBN, our Wasserstein loss yields further significant gains across all datasets. As our WMDD method already achieves high performance (with our 100 IPC results approaching those of full dataset training), these consistent improvements confirm the effectiveness of our design choices.

$\mathcal{L}_{\text{feature}}$	$\mathcal{L}_{\text{reg}}$	ImageNette	Tiny ImageNet	ImageNet-1K
Wass.	**PCBN**	**64.7**$\pm$ **0.2**	**41.8**$\pm$ **0.1**	**38.1**$\pm$ **0.1**
CE	PCBN	63.5$\pm$ 0.1	41.0$\pm$ 0.2	36.4$\pm$ 0.2
Wass.	BN	60.7$\pm$ 0.2	36.6$\pm$ 0.1	26.8$\pm$ 0.3
CE	BN	54.2$\pm$ 0.1	38.0$\pm$ 0.3	35.9$\pm$ 0.2

Table 3. Ablation study on two variables: whether to use our Wasserstein (Wass.) loss or the cross-entropy (CE) loss in previous DD works [52, 53] for feature matching ($\mathcal{L}_{\text{feature}}$), and whether to use standard BatchNorm (BN) or our PCBN method for regularization ($\mathcal{L}_{\text{reg}}$). We report the mean and standard error of performance on 5 repetitive runs.

Additionally, we find that directly replacing the Wasserstein metric in our method with MMD results in near-random performance on Tiny-ImageNet and ImageNet-1K. This motivates a deeper analysis of different distribution metrics, which we provide below.

5.5 Comparison with Alternative Metrics

The MMD Metric. Table 1 shows that our method using the Wasserstein metric outperforms all previous DD methods, including MMD-based methods such as [60]. A more direct comparison between the two distribution metrics is tricky, because existing MMD-based methods require feature spaces from dozens of randomly initialized models, which is incompatible with our algorithm using a single pretrained model. Simply replacing the Wasserstein metric in our method with MMD results in near-random performance. To try to make a fair comparison, we removed engineering tricks from DD methods using both metrics and evaluated their vanilla versions on Tiny-ImageNet. Specifically, we compared our method with a seminal MMD-based method [60] on Tiny-ImageNet, and removed all engineering tricks including fancy augmentations (e.g., rotation, color jitter, and mixup) used in both methods and the FKD [37] used in our method. According to the result in Figure 4, the Wasserstein metric yields better synthetic data in all settings. In 1 IPC setting, the MMD metric yields random performance, likely due to empirical approximation errors and its focus on feature means rather than their geometric properties. In Appendix B, we provide a possible

theoretical explanation for the superior performance of the Wasserstein metric by combining error bound analysis with the practicality of existing MMD-based methods.

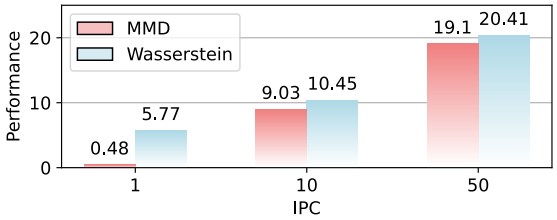


Figure 4. Performance comparison of MMD distance vs. the Wasserstein distance. The evaluation model is ResNet18.

The Sliced Wasserstein Distance. Beyond MMD, we also examined the Sliced Wasserstein (SW) distance [32], which has shown promise in reducing computational cost while retaining key aspects of Wasserstein geometry. In Table 4, we compare our Wasserstein barycenters to those computed with SW and show that the latter achieves comparable accuracy with a modest increase in speed. However, our full barycenter computation is already highly efficient, accounting for only a small fraction of the overall runtime.

Method	Accuracy (%)			Time (hour)		
IPC	1	10	50	1	10	50
WMDD (Ours)	**7.6**	**41.8**	**59.4**	0.71	2.30	5.27
Sliced Wass.	7.4	41.1	58.3	**0.68**	**2.23**	**5.16**

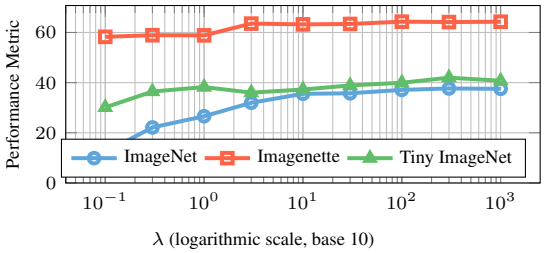
Table 4. Performance and efficiency comparison with Sliced Wass. Distance on Tiny-ImageNet.

5.6 Hyperparameter Sensitivity

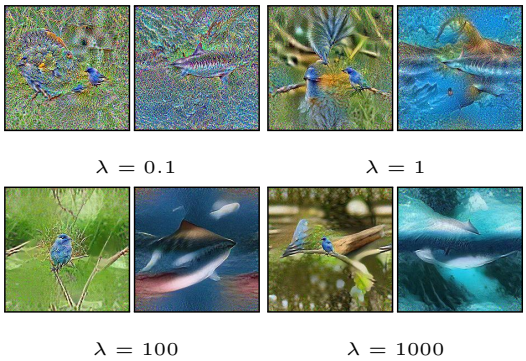
We analyze the sensitivity of key hyperparameters below.

Regularization Strength. To analyze how the regularization term affects our method (Eq. 15), we tested λ values ranging from 10^{-1} to 10^{3} and evaluated performance on three datasets in 10 IPC setting. Figure 5a shows that small λ result in lower performance across all datasets. Performance improves as λ increases, stabilizing around a threshold of approximately 10.0. This demonstrates that while regularization enhances dataset quality, our method remains robust to specific λ values. Figure 5b illustrates the regularization effect on synthetic images of the same class. When λ is too small, synthetic images exhibit high-frequency components, suggesting overfitting to model weights and architecture. In contrast, sufficiently large λ values produce synthetic images that better align with human perception.

Features from Different Layers. Beyond regularization strength, we also examined which network layer provides the most effective features for our Wasserstein barycenter computation. Table 5 shows the performance with features from different layers of ResNet-18 on Tiny-ImageNet. The



(a) Effect of λ on WMDD performance on the three datasets.



(b) Visualization of synthetic images from Imagenet-1K of classes indigo bird (left) and tiger shark (right), with different λ.

Figure 5. Effect of regularization strength λ on our method.

accuracy increases and then stabilizes by Layer 16, indicating WMDD leverages high-level, abstract representations.

Layer	5	10	15	16	17	18
Acc (%)	2.4	11.3	37.6	41.1	41.6	**41.8**

Table 5. Performance of WMDD using features from different layers of the backbone.

6 Conclusion

This work introduces a new dataset distillation approach leveraging Wasserstein metrics, grounded in optimal transport theory, to achieve more precise distribution matching. Our method learns synthetic datasets by matching the Wasserstein barycenter of the data distribution in the feature space of pretrained models, combined with a simple regularization technique to leverage the prior knowledge in these models. Through empirical testing, our approach has demonstrated impressive performance across a variety of benchmarks, highlighting its reliability and practical applicability in diverse scenarios. Findings from our controlled experiments corroborate the utility of Wasserstein metrics for capturing the essence of data distributions. Future work will aim to explore the integration of advanced metrics with generative methods, aligning with the broader goal of advancing data efficiency in computer vision.

References

[1] Martial Agueh and Guillaume Carlier. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011. 2, 3

[2] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 4, 15

[3] 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*, 2022. 2, 6

[4] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Generalizing dataset distillation via deep generative prior. *CVPR*, 2023. 7

[5] Lechao Cheng, Chaowei Fang, Dingwen Zhang, Guanbin Li, and Gang Huang. Compound batch normalization for long-tailed image classification. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1925–1934, 2022. 2, 6

[6] Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Scaling up dataset distillation to imagenet-1k with constant memory. In *International Conference on Machine Learning*, pages 6565–6590. PMLR, 2023. 2

[7] Justin Cui, Ruochen Wang, Yuanhao Xiong, and Cho-Jui Hsieh. Mitigating bias in dataset distillation. *arXiv preprint arXiv:2406.06609*, 2024. 2

[8] Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. In *International conference on machine learning*, pages 685–693. PMLR, 2014. 2, 4, 5, 14

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 6

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

[11] Tian Dong, Bo Zhao, and Lingjuan Lyu. Privacy for free: How does dataset condensation help privacy? In *International Conference on Machine Learning*, pages 5378–5396. PMLR, 2022. 2

[12] 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. *arXiv preprint arXiv:2010.11929*, 2020. 7

[13] 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*, pages 3749–3758, 2023. 2

[14] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. 6

[15] Jiahui Geng, Zongxiong Chen, Yuandou Wang, Herbert Woisetschlaeger, Sonja Schimmler, Ruben Mayer, Zhiming Zhao, and Chunming Rong. A survey on dataset distillation: Approaches, applications and future directions, 2023. 3

[16] Jack Goetz and Ambuj Tewari. Federated learning via synthetic data. *arXiv preprint arXiv:2008.04489*, 2020. 2

[17] Xinyu Gong, Wuyang Chen, Tianlong Chen, and Zhangyang Wang. Sandwich batch normalization: A drop-in replacement for feature distribution heterogeneity. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2494–2504, 2022. 2, 6

[18] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 2, 13

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6, 7

[20] Jeremy Howard. Imagenette dataset, 2019. Available at: https://github.com/fastai/imagenette. 6

[21] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. PMLR, 2015. 2

[22] Leonid V Kantorovich. Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422, 1960. 2

[23] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 6

[24] 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

[25] Shiye Lei and Dacheng Tao. A comprehensive survey of dataset distillation. *arXiv preprint arXiv:2301.05603*, 2022. 2, 7

[26] Shiye Lei and Dacheng Tao. A comprehensive survey of dataset distillation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):17–32, 2024. 3

[27] Yijiang Li, Wentian Cai, Ying Gao, Chengming Li, and Xiping Hu. More than encoder: Introducing transformer decoder to upsample. In *2022 IEEE international conference on bioinformatics and biomedicine (BIBM)*, pages 1597–1602. IEEE, 2022. 2

[28] Yijiang Li, Ying Gao, and Haohan Wang. Towards understanding adversarial transferability in federated learning. *Transactions on Machine Learning Research*, 2023. 2

[29] Songhua Liu, Kai Wang, Xingyi Yang, Jingwen Ye, and Xinchao Wang. Dataset distillation via factorization. *Advances in Neural Information Processing Systems*, 35:1100–1113, 2022. 2, 12

[30] Noel Loo, Ramin Hasani, Alexander Amini, and Daniela Rus. Efficient dataset distillation using random feature ap-

proximation. In *Advances in Neural Information Processing Systems*, 2022. 2

[31] TorchVision maintainers and contributors. Torchvision: Pytorch's computer vision library. https://github.com/pytorch/vision, 2016. 16

[32] Khai Nguyen and Nhat Ho. Energy-based sliced wasserstein distance. *Advances in Neural Information Processing Systems*, 36:18046–18075, 2023. 8

[33] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. In *International Conference on Learning Representations*, 2021. 2

[34] Noveen Sachdeva and Julian McAuley. Data distillation: A survey, 2023. 2, 3

[35] 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*, pages 17096–17107, 2022. 3, 6

[36] Shitong Shao, Zeyuan Yin, Muxin Zhou, Xindong Zhang, and Zhiqiang Shen. Generalized large-scale data condensation via various backbone and statistical matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16709–16718, 2024. 2, 6, 12

[37] Zhiqiang Shen and Eric Xing. A fast knowledge distillation framework for visual recognition. In *European Conference on Computer Vision*, pages 673–690. Springer, 2022. 7, 12

[38] Shuo Shi, Peng Sun, Xinyi Shang, Tianyu Du, Xuhong Zhang, Jianwei Yin, and Tao Lin. Privacy as a free lunch: Crafting initial distilled datasets through the kaleidoscope. 2

[39] Seungjae Shin, Heesun Bae, Donghyeok Shin, Weonyoung Joo, and Il-Chul Moon. Loss-curvature matching for dataset selection and condensation, 2023. 2

[40] Felipe Petroski Such, Aditya Rawal, Joel Lehman, Kenneth Stanley, and Jeffrey Clune. Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data. In *International Conference on Machine Learning*, pages 9206–9216. PMLR, 2020. 2

[41] Peng Sun, Bei Shi, Daiwei Yu, and Tao Lin. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm, 2023. 3

[42] Ilya O Tolstikhin, Bharath K. Sriperumbudur, and Bernhard Schölkopf. Minimax estimation of maximum mean discrepancy with radial kernels. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2016. 2

[43] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 17

[44] Cédric Villani. *Optimal Transport: Old and New*. Springer Science & Business Media, 2008. 2

[45] 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*, pages 12196–12205, 2022. 2, 3, 14

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

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

[48] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation. arXiv preprint arXiv:2006.08545, 2020. 2

[49] Eric Xue, Yijiang Li, Haoyang Liu, Peiran Wang, Yifan Shen, and Haohan Wang. Towards adversarially robust dataset distillation by curvature regularization. *arXiv preprint arXiv:2403.10045*, 2024. 2

[50] Eric Xue, Yijiang Li, Haoyang Liu, Peiran Wang, Yifan Shen, and Haohan Wang. Towards adversarially robust dataset distillation by curvature regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9041–9049, 2025. 2

[51] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020. 5, 6

[52] Zeyuan Yin and Zhiqiang Shen. Dataset distillation in large data era, 2023. 6, 7

[53] Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective, 2023. 2, 3, 5, 6, 7, 17, 19

[54] Ruonan Yu, Songhua Liu, and Xinchao Wang. Dataset distillation: A comprehensive review, 2023. 2

[55] Hansong Zhang, Shikun Li, Pengju Wang, Dan Zeng, and Shiming Ge. M3d: Dataset condensation by minimizing maximum mean discrepancy, 2024. 3, 14

[56] Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. DENSE: Data-free one-shot federated learning. In *Advances in Neural Information Processing Systems*, 2022. 2

[57] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, 2021. 2

[58] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching, 2022. 2, 3

[59] Bo Zhao and Hakan Bilen. Synthesizing informative training samples with gan. *arXiv preprint arXiv:2204.07513*, 2022. 2, 3

[60] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6514–6523, 2023. 2, 3, 6, 7, 14

[61] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021. 1, 2

[62] Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation, 2023. 2, 3, 12, 14

[63] Muxin Zhou, Zeyuan Yin, Shitong Shao, and Zhiqiang Shen. Self-supervised dataset distillation: A good compression is all you need. *arXiv preprint arXiv:2404.07976*, 2024. 6

[64] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. *arXiv preprint arXiv:2206.00719v2*, 2022. 2