DANCE: Dual-View Distribution Alignment for Dataset Condensation

Hansong Zhang[1,2], **Shikun Li**[1,2], **Fanzhao Lin**[1,2], **Weiping Wang**[1,2]
Zhenxing Qian[3], **Shiming Ge**[1,2*]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100092, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
[3]School of Computer Science, Fudan University, Shanghai 200433, China
{zhanghansong,lishikun,linfanzhao,wangweiping,geshiming}@iie.ac.cn, zxqian@fudan.edu.cn

Abstract

Dataset condensation addresses the problem of data burden by learning a small synthetic training set that preserves essential knowledge from the larger real training set. To date, the state-of-the-art (SOTA) results are often yielded by optimization-oriented methods, but their inefficiency hinders their application to realistic datasets. On the other hand, the Distribution-Matching (DM) methods show remarkable efficiency but sub-optimal results compared to optimization-oriented methods. In this paper, we reveal the limitations of current DM-based methods from the inner-class and inter-class views, *i.e.*, *Persistent Training* and *Distribution Shift*. To address these problems, we propose a new DM-based method named **D**ual-view distribution **A**lig**N**ment for dataset **C**ond**E**nsation (DANCE), which exploits a few pre-trained models to improve DM from both inner-class and inter-class views. Specifically, from the inner-class view, we construct multiple "mid encoders" to perform pseudo long-term distribution alignment, making the condensed set a good proxy of the real one during the whole training process; while from the inter-class view, we use the expert models to perform distribution calibration, ensuring the synthetic data remains in the real class region during condensing. Experiments demonstrate the proposed method achieves a SOTA performance while maintaining comparable efficiency with the original DM across various scenarios. Source codes are available at https://github.com/Hansong-Zhang/DANCE.

1 Introduction

Recently, the reliance on large-scale datasets, which may include millions or even billions of examples, has become essential for developing state-of-the-art (SOTA) models [Zhao and Bilen, 2021a; Xia *et al.*, 2022; Li *et al.*, 2023; Li *et al.*, 2024; Zhang *et al.*, 2024b]. However, this reliance brings significant challenges, primarily due to the substantial storage costs and computational expenses required for training such models. These challenges pose formidable obstacles, particularly for startups and non-profit organizations, making advanced model training often unattainable [Coleman *et al.*, 2020; Sorscher *et al.*, 2022; Zheng *et al.*, 2023; Jin *et al.*, 2022; Yang *et al.*, 2023; Geng *et al.*, 2023; Xia *et al.*, 2024].

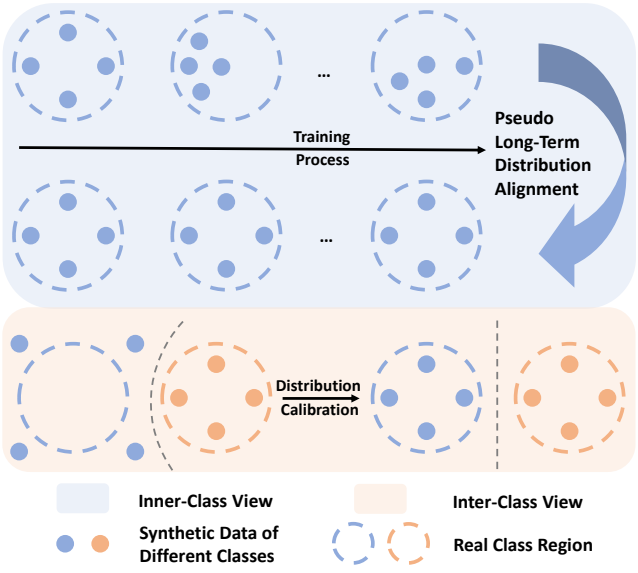


Figure 1: **Two views of the proposed DANCE.** For inner-class view, it ensures that the synthetic data remains a faithful proxy of the real data throughout the training process. For inter-class view, it also prevents the synthetic data from falling outside the real class region (the domain where all real data points of that class reside), which may change the decision boundary of the learned classifier.

As a remedy, *Dataset Condensation* (DC), also known as *Dataset Distillation*, has emerged as a prominent solution to address the challenges of data burden [Wang *et al.*, 2018; Cui *et al.*, 2022; Yu *et al.*, 2024]. It involves learning a small condensed training set to replicate the performance of models trained on larger real datasets. Pioneer methods in this area typically focus on matching either the gradients [Zhao and Bilen, 2021a; Zhao and Bilen, 2021b; Kim *et al.*, 2022; Wang *et al.*, 2023] or parameters [Cazenavette *et al.*, 2022; Du *et al.*, 2023; Guo *et al.*, 2024; Liu *et al.*, 2022] between real and synthetic data, which can be categorized as *Optimization-*

*Shiming Ge is the corresponding author.

Oriented methods. While these methods have shown success, their reliance on the bi-level optimization or nested gradients often results in prohibitively high computational costs [Zhang *et al.*, 2023; Sajedi *et al.*, 2023; Liu *et al.*, 2021; Zhang *et al.*, 2024a], limiting their practical application in wider scenarios.

To address the scalability challenges in DC, *Distribution Matching* (DM) [Zhao and Bilen, 2023] has been proposed. It focuses on aligning the latent representations extracted by randomly-initialized encoders, based on the rationale that the condensed set should represents the real training set in the feature space. Unlike previous *Optimization-Oriented* methods, DM avoids the computationally intensive nested optimization loops, significantly reducing the time required for condensation and thereby enhancing its applicability in diverse scenarios [Loo *et al.*, 2022; Zhou *et al.*, 2023; Cazenavette *et al.*, 2023; Liu *et al.*, 2023a; Nguyen *et al.*, 2021]. However, despite these advantages, DM's performance still falls short of SOTA optimization-oriented methods such as MTT [Cazenavette *et al.*, 2022], IDC [Kim *et al.*, 2022], and DREAM [Liu *et al.*, 2023b].

In this paper, we conduct an in-depth analysis of DM from the inner-class and inter-class views, pointing out the limitations of current DM-based methods and provide our corresponding remedies. Specifically, from the **inner-class view**, to ensure an alignment during the whole training process, previous works like IDM [Zhao *et al.*, 2023] and CAFE [Wang *et al.*, 2022] naively use the models trained from scratch to extract the latent representations. While effective, the *Persistent Training*, i.e. numerous model updating steps, is very time-consuming thus greatly hinders their efficiency. To counter this, we introduce Pseudo Long-Term Distribution Alignment (PLTDA), where we use the convex combination of initialized and trained expert models to perform inner-class distribution alignment, eliminating the need for persistent training. From the **inter-class view**, we reveal the *Distribution Shift* phenomenon in DM, i.e., the synthetic data will diverge from the real class region during condensation, which may change the decision boundary of the learned classifier. To address this, we employ expert models for Distribution Calibration, ensuring the synthetic data remains within the real class region. We term the proposed method as **D**ual-view distribution **A**lignme**N**t for dataset **C**ond**E**nsation (DANCE), for we enhance DM by utilizing the knowledge of expert models from the above two views, which is illustrated in Fig. 1. As will be shown in the experiments, DANCE can achieve comparable results to SOTA optimization-oriented methods even with only a single expert model.

Our main contributions are outlined as follows:

[C1]: We identify and analyze the limitations of current DM-based dataset condensation methods from inner- and inter-class views, which reveals two major issues: persistent training and distribution shift.

[C2]: We introduce DANCE by incorporating two modules to effectively mitigate the above two issues inherent in DM-based methods.

[C3]: We conduct extensive experiments across a variety of datasets under different resolutions. The results demonstrate that DANCE establishes a strong baseline in dataset condensation, significantly advancing both performance and efficiency, particularly in the realm of distribution matching.

2 Preliminaries

In this section, we initially formalize the concept of dataset condensation (DC) and then recap the DM method [Zhao and Bilen, 2023], which is pivotal as it represents the pioneering work in the realm of distribution matching within DC and lays the groundwork for our research.

Problem Definition. Given a large real training set $\mathcal{D}_{\mathsf{real}} = \{(\boldsymbol{x}_i^{\mathsf{real}}, y_i^{\mathsf{real}})\}_{i=1}^{|\mathcal{D}_{\mathsf{real}}|}$, *Dataset Condensation* or *Dataset Distillation* aims to generate a small training set $\mathcal{D}_{\mathsf{syn}} = \{(\boldsymbol{x}_j^{\mathsf{syn}}, y_j^{\mathsf{syn}})\}_{j=1}^{|\mathcal{D}_{\mathsf{syn}}|}$ ($|\mathcal{D}_{\mathsf{syn}}| \ll |\mathcal{D}_{\mathsf{real}}|$), so that the model trained on $\mathcal{D}_{\mathsf{syn}}$ and the model trained on $\mathcal{D}_{\mathsf{real}}$ (denoted as $\boldsymbol{\theta}_{\mathsf{syn}}$ and $\boldsymbol{\theta}_{\mathsf{real}}$ respectively) will have similar performance on the unseen data. Formally, let P_D represent the distribution of the real data, ℓ be the loss operation such as cross-entropy, the synthetic training set can be obtained by minimizing the performance gap between the two models:

$$\mathcal{D}_{\mathsf{syn}}^{\star} = \underset{\mathbb{E}_{(\boldsymbol{x},y)\sim P_D}}{\arg\min} \; ||\ell(\boldsymbol{\theta}_{\mathsf{syn}}(\boldsymbol{x}), y) - \ell(\boldsymbol{\theta}_{\mathsf{real}}(\boldsymbol{x}), y)||. \quad (1)$$

Distribution Matching. To solve Eq. (1), previous optimization-oriented methods have attempted to 1) update the $\mathcal{D}_{\mathsf{syn}}$ using a meta-learning framework. 2) match the gradient or parameter induced by $\mathcal{D}_{\mathsf{syn}}$ and $\mathcal{D}_{\mathsf{real}}$. However, both the above methods involves a bi-level optimization, which is computationally inefficient due to the calculation of nested gradients. To improve the condensing efficiency, DM [Zhao and Bilen, 2023] first proposed Distribution Matching, which learns the condensed set by aligning the feature distributions of $\mathcal{D}_{\mathsf{syn}}$ and $\mathcal{D}_{\mathsf{real}}$. Specifically, the condensed set in DM is optimized by:

$$\mathcal{D}_{\mathsf{syn}}^{\star} = \underset{\mathbb{E}_{\boldsymbol{\phi}_0\sim P_{\boldsymbol{\phi}_0}}}{\arg\min} \left\| \frac{\sum_{i=1}^{|\mathcal{D}_{\mathsf{real}}|} \boldsymbol{\phi}_0(\boldsymbol{x}_i^{\mathsf{real}})}{|\mathcal{D}_{\mathsf{real}}|} - \frac{\sum_{j=1}^{|\mathcal{D}_{\mathsf{syn}}|} \boldsymbol{\phi}_0(\boldsymbol{x}_j^{\mathsf{syn}})}{|\mathcal{D}_{\mathsf{syn}}|} \right\|^2, \quad (2)$$

where $\boldsymbol{\phi}_0 \sim P_{\boldsymbol{\phi}_0}$ denotes the randomly-initialized feature extractor (instanced by a random DNN $\boldsymbol{\theta}_0$ without the linear classification layer). Compared to optimization-oriented methods, DM significantly enhances the computational efficiency and has shown better generalization ability across different architectures [Zhao and Bilen, 2023].

3 Methodology

While DM [Zhao and Bilen, 2023] brings remarkable efficiency and cross-architecture performance, the quality of the condensed set it generates typically falls short of those produced by SOTA optimization-oriented methods like IDC [Kim *et al.*, 2022] and MTT [Cazenavette *et al.*, 2022]. In this paper, we aim to enhance the alignment between the distributions of $\mathcal{D}_{\mathsf{real}}$ and $\mathcal{D}_{\mathsf{syn}}$, considering both the **inner-class view** and the **inter-class view**. Sections 3.1 and 3.2 will detail the limitations of current DM-based methods from these two perspectives and introduce our proposed solutions. Subsequently, we describe our overall training algorithm in Section 3.3. Our method, termed **D**ual-view distribution **A**lig**N**ment for dataset **C**ond**E**nsation (DANCE), is depicted in Fig. 2.

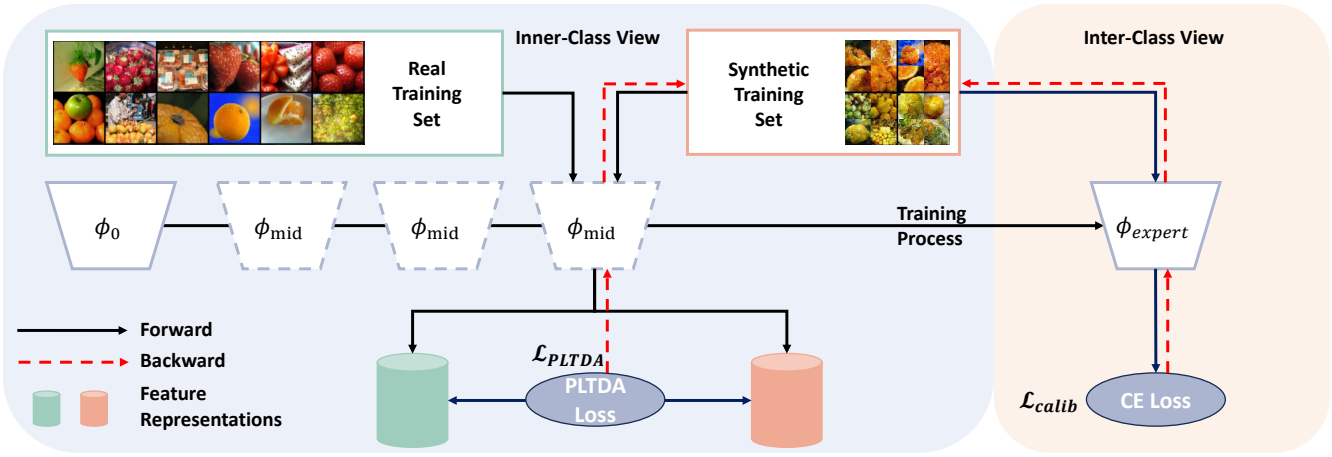


Figure 2: **The framework of DANCE**. From the inner-class view, multiple mid encoders are constructed to perform Pseudo Long-Term Distribution Alignment so that the synthetic set can remain a good proxy of its class during training. From the inter-class view, the Distribution Calibration is performed, ensuring the synthetic data stay within the real class region during condensing process.

3.1 Inner-Class View

Limitation of DM. For data from the same class, DM [Zhao and Bilen, 2023] employs various randomly-initialized deep encoders to extract latent representations of $\mathcal{D}_{\text{real}}$ and $\mathcal{D}_{\text{syn}}$. It then minimizes the discrepancy of feature distributions to ensure they are aligned (Eq. (2)). However, we contend that aligning feature distributions from randomly initialized extractors, which are sampled from a probability distribution over parameters P_{ϕ_0}, does not ensure that $\mathcal{D}_{\text{syn}}$ remains a reliable proxy for $\mathcal{D}_{\text{real}}$ throughout all training stages. This divergence may ultimately cause the model trained on $\mathcal{D}_{\text{syn}}$ to deviate from the one trained on $\mathcal{D}_{\text{real}}$. To illustrate this divergence throughout the entire training process, we compute the discrepancy between $\mathcal{D}_{\text{syn}}$ and $\mathcal{D}_{\text{real}}$ at various stages of training. As depicted in Fig. 3a, DM fails to maintain the informativeness of the condensed set throughout the training procedure. During training, the distribution of the data condensed by DM increasingly deviates from that of the real data.

Remark. The misalignment issue is also noted in CAFE [Wang *et al.*, 2022] and IDM [Zhao *et al.*, 2023]. As a remedy, these approaches involve training multiple models from scratch to extract features of $\mathcal{D}_{\text{syn}}$ and $\mathcal{D}_{\text{real}}$ during the condensation process. While such *Persistent Training* yields improved performance, it is hampered by the following two drawbacks:

Drawbacks. 1) *Hyper-parameter tuning*: The models used for condensation need to be **persistently** trained during the condensing process in CAFE and IDM to enrich the distilled knowledge. This process, however, involves meticulous tuning of multiple parameters, including the number of training steps, iteration count, and learning rate during the model-updating phase. Moreover, as model performance often sees a significant rise at the onset of training, the models at early stages are prone to be skipped due to overdoing the update steps, thereby hampering the effectiveness of the condensed images. 2) *Inefficiency*: Both CAFE and IDM necessitate optimizing hundreds of random models each time a dataset is condensed, which is impractical, especially for datasets with larger resolution.

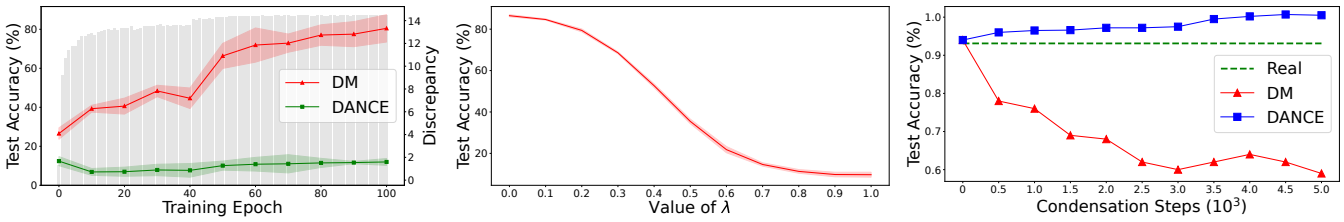
Pseudo Long-Term Distribution Alignment (PLTDA). To tackle the misalignment issue from the inner-class perspective, we introduce a straightforward and effective module, termed **P**seudo **L**ong-**T**erm **D**istribution **A**lignment (PLTDA). Specifically, rather than relying on models trained on real data, we employ a convex combination of randomly-initialized encoders and their corresponding trained counterparts. We refer to these trained encoders as "expert encoders" (ϕ_{expert}), because their corresponding "expert models" ($\boldsymbol{\theta}_{\text{expert}}$) represent the upper bound of the performance of $\mathcal{D}_{\text{syn}}$ and the end of the training process. We term this combination as "mid encoders" (ϕ_{mid}) of "mid models" ($\boldsymbol{\theta}_{\text{mid}}$), which is calculated by:

$$\phi_0 \quad \rightarrow \quad \underbrace{\lambda \cdot \phi_0 + (1-\lambda) \cdot \phi_{\text{expert}}}_{\phi_{\text{mid}}} \quad \rightarrow \quad \phi_{\text{expert}} \tag{3}$$

where $\lambda \sim U(0,1)$ is a randomly generated coefficient for encoder combination. After obtaining the mid encoder, we calculate the loss in PLTDA as:

$$\mathcal{L}_{\text{PLTDA}} = \left\| \frac{\sum_{i=1}^{|\mathcal{D}_{\text{real}}|} \phi_{\text{mid}}(\boldsymbol{x}_i^{\text{real}})}{|\mathcal{D}_{\text{real}}|} - \frac{\sum_{j=1}^{|\mathcal{D}_{\text{syn}}|} \phi_{\text{mid}}(\boldsymbol{x}_j^{\text{syn}})}{|\mathcal{D}_{\text{syn}}|} \right\|^2. \tag{4}$$

Compared to CAFE and IDM, which update the model during the condensation process, our expert encoders do not introduce additional hyper-parameters. Furthermore, as illustrated in Fig. 3b, the performance of the mid models changes more gradually across different values of λ, allowing for the generation of models at various training stages. This approach is advantageous because the standard model training often neglects early-stage models, a phenomenon evident in Fig. 3a (the model performance reaches at a high-level after a few epochs). Additionally, the mid encoders rely solely



(a) (b) (c)

Figure 3: (a) **The distance between the distribution of $\mathcal{D}_{\text{syn}}$ and $\mathcal{D}_{\text{real}}$ across the whole training process.** (b) **The performance of the mid model ϕ_{mid} at different value of λ.** (c) **Accuracy (%) of the expert model on the real test data, and the synthesized data of DM and DANCE.** The evaluations are conducted on CIFAR-10, where (a) and (c) adopts 10 images per class.

on the randomly-initialized encoder ϕ_0 and the expert encoders ϕ_{expert}. Both can be pre-trained offline and reused for different values of images per class (IPCs). Notably, our method does not require a large number of pre-trained expert models like IDC [Kim *et al.*, 2022] or MTT [Cazenavette *et al.*, 2022]. As we will demonstrate in Sec. 4.3, our approach achieves state-of-the-art results across various scenarios with just a single expert model.

3.2 Inter-Class View

Limitation of DM. As depicted in [Zhao and Bilen, 2023], DM aligns the distribution between $\mathcal{D}_{\text{syn}}$ and $\mathcal{D}_{\text{real}}$ in a class-wise manner, while overlooking inter-class constraints. Since many existing DM-based methods [Sajedi *et al.*, 2023; Zhang *et al.*, 2024a] follow the class-wise learning manner of DM, these approaches may have the following drawback:

Drawback. *Distribution Shift*: The synthetic data may fall outside the real class region during condensing, which will affect the decision boundary of the learned classifier. As shown in Fig.3c, the expert model trained by real training data has an excellent performance on real test data, while it cannot achieve a high classification accuracy on the synthetic data generated by DM. It can be inferred from this phenomenon that, many of the synthetic examples are not within the real class region and even cross the decision boundary of the expert model.

Distribution Calibration. To address the above issue from the inter-class view, we integrate a module called *Distribution Calibration* into our approach. This module utilizes expert models $\boldsymbol{\theta}_{\text{expert}}$ to calibrate the inter-class distribution of $\mathcal{D}_{\text{syn}}$ after PLTDA. Specifically, once the inner-class matching is completed, we impose the following calibration loss, computed using the chosen $\boldsymbol{\theta}_{\text{expert}}$, to prevent the synthetic data from straying from their respective categories:

$$\mathcal{L}_{\text{calib}} = \frac{1}{|\mathcal{D}_{\text{syn}}|} \sum_{j=1}^{|\mathcal{D}_{\text{syn}}|} \ell(\boldsymbol{\theta}_{\text{expert}}(\boldsymbol{x}_j^{\text{syn}}), y_j^{\text{syn}}). \tag{5}$$

It is important to note that, although "Discrimination Loss" in CAFE [Wang *et al.*, 2022] and "Distribution Regularization" in IDM [Zhao *et al.*, 2023] employ a similar concept, they utilize un-converged models for calculating their losses, instead of expert models. This may lead to sub-optimal outcomes due to the relatively poorer generalization ability of such models compared to expert models.

Algorithm 1 Dual-View Distribution Alignment for Dataset Condensation

Input: Real training set $\mathcal{D}_{\text{real}}$
Parameter: Number of expert models N; Number of condensation iterations I; Learning rate of the condensed set η; Calibration interval I_c
Output: The condensed set $\mathcal{D}_{\text{syn}}$

Initialize $\mathcal{D}_{\text{syn}}$ with randomly selected real data
Pre-train N expert encoders $\{\phi_{\text{expert}}^n\}_{n=1}^N$ and save their corresponding initial encoders $\{\phi_0^n\}_{n=1}^N$
for $i = 1, 2, \ldots, I$ **do**
  Randomly select an expert encoder ϕ_{expert}^n and generate the mid encoder ϕ_{mid}^n by Eq. (3)
  Calculate the matching loss $\mathcal{L}_{\text{PLTDA}}$ by Eq. (4)
  Update the $\mathcal{D}_{\text{syn}}$ by $\mathcal{D}_{\text{syn}} = \mathcal{D}_{\text{syn}} - \eta \nabla_{\mathcal{D}_{\text{syn}}} \mathcal{L}_{\text{PLTDA}}$
  if $i \% I_c = 0$ **then**
    Calculate the calibration loss $\mathcal{L}_{\text{calib}}$ by Eq. (5)
    Update the $\mathcal{D}_{\text{syn}}$ by $\mathcal{D}_{\text{syn}} = \mathcal{D}_{\text{syn}} - \eta \nabla_{\mathcal{D}_{\text{syn}}} \mathcal{L}_{\text{calib}}$
  end if
end for
Return: $\mathcal{D}_{\text{syn}}$

3.3 Training Algorithm

The pseudo-code of DANCE is detailed in Algorithm 1. Besides incorporating the PLTDA (Sec. 3.1) and Distribution Calibration (Sec. 3.2), our approach also integrates a prevalent data augmentation technique known as "Factoring & Upsampling". In this technique, each image space in $\mathcal{D}_{\text{syn}}$ is divided into $l \times l$ smaller sections in order to host multiple synthetic images. These mini-images are subsequently up-sampled to their original dimensions during model training. This augmentation strategy was initially introduced by IDC [Kim *et al.*, 2022] and has since been widely employed in various dataset condensation works [Liu *et al.*, 2023b; Zhao *et al.*, 2023].

4 Experiments

4.1 Experimental Setup

Datasets. We assess our method using three low-resolution datasets: Fashion-MNIST [Xiao *et al.*, 2017] with a resolution of 28×28, and CIFAR-10/100 [Krizhevsky, 2009] with a resolution of 32×32. For medium-resolution data,

	Fashion-MNIST			CIFAR-10			CIFAR-100			TinyImageNet		
Resolution	28×28			32×32			32×32			64×64		
IPC	1	10	50	1	10	50	1	10	50	1	10	50
Ratio (%)	0.017	0.17	0.83	0.02	0.2	1	0.02	0.2	1	0.2	2	10
Random	$51.4_{\pm 3.8}$	$73.8_{\pm 0.7}$	$82.5_{\pm 0.7}$	$14.4_{\pm 2.0}$	$26.0_{\pm 1.2}$	$43.4_{\pm 1.0}$	$4.2_{\pm 0.3}$	$14.6_{\pm 0.5}$	$30.0_{\pm 0.4}$	$1.4_{\pm 0.1}$	$5.0_{\pm 0.2}$	$15.0_{\pm 0.4}$
Herding	$67.0_{\pm 1.9}$	$71.1_{\pm 0.7}$	$71.9_{\pm 0.8}$	$21.5_{\pm 1.2}$	$31.6_{\pm 0.7}$	$40.4_{\pm 0.6}$	$8.4_{\pm 0.3}$	$17.3_{\pm 0.3}$	$33.7_{\pm 0.5}$	$2.8_{\pm 0.2}$	$6.3_{\pm 0.2}$	$16.7_{\pm 0.3}$
K-Center	$66.9_{\pm 1.8}$	$54.7_{\pm 1.5}$	$68.3_{\pm 0.8}$	$21.5_{\pm 1.3}$	$14.7_{\pm 0.9}$	$27.0_{\pm 1.4}$	$8.3_{\pm 0.3}$	$7.1_{\pm 0.2}$	$30.5_{\pm 0.3}$	-	-	-
DC	$70.5_{\pm 0.6}$	$82.3_{\pm 0.4}$	$83.6_{\pm 0.4}$	$28.3_{\pm 0.5}$	$44.9_{\pm 0.5}$	$53.9_{\pm 0.5}$	$12.8_{\pm 0.3}$	$25.2_{\pm 0.3}$	-	$5.3_{\pm 0.1}$	$12.9_{\pm 0.1}$	$12.7_{\pm 0.4}$
DSA	$70.6_{\pm 0.6}$	$84.6_{\pm 0.3}$	**88.7**$_{\pm 0.2}$	$28.8_{\pm 0.7}$	$52.1_{\pm 0.5}$	$60.6_{\pm 0.5}$	$13.9_{\pm 0.3}$	$32.3_{\pm 0.3}$	$42.8_{\pm 0.4}$	$5.7_{\pm 0.1}$	$16.3_{\pm 0.2}$	$5.1_{\pm 0.2}$
IDC	$81.0_{\pm 0.2}$	$86.0_{\pm 0.3}$	$86.2_{\pm 0.2}$	**50.6**$_{\pm 0.4}$	$67.5_{\pm 0.5}$	$74.5_{\pm 0.1}$	-	$45.1_{\pm 0.4}$	-	-	-	-
DREAM	**81.3**$_{\pm 0.2}$	**86.4**$_{\pm 0.3}$	$86.8_{\pm 0.3}$	**51.1**$_{\pm 0.3}$	**69.4**$_{\pm 0.4}$	**74.8**$_{\pm 0.1}$	**29.5**$_{\pm 0.3}$	**46.8**$_{\pm 0.7}$	**52.6**$_{\pm 0.4}$	$10.0_{\pm 0.4}$	-	**29.5**$_{\pm 0.3}$
MTT	-	-	-	$31.9_{\pm 1.2}$	$56.4_{\pm 0.7}$	$65.9_{\pm 0.6}$	$24.3_{\pm 0.3}$	$40.1_{\pm 0.4}$	$47.7_{\pm 0.2}$	$6.2_{\pm 0.4}$	$17.3_{\pm 0.2}$	$26.5_{\pm 0.3}$
CAFE	$77.1_{\pm 0.9}$	$83.0_{\pm 0.4}$	$84.8_{\pm 0.4}$	$30.3_{\pm 1.1}$	$46.3_{\pm 0.6}$	$55.5_{\pm 0.6}$	$12.9_{\pm 0.3}$	$27.8_{\pm 0.3}$	$37.9_{\pm 0.3}$	-	-	-
CAFE+DSA	$73.7_{\pm 0.7}$	$83.0_{\pm 0.3}$	**88.2**$_{\pm 0.3}$	$31.6_{\pm 0.8}$	$50.9_{\pm 0.5}$	$62.3_{\pm 0.4}$	$14.0_{\pm 0.3}$	$31.5_{\pm 0.2}$	$42.9_{\pm 0.2}$	-	-	-
DM	$70.7_{\pm 0.6}$	$83.5_{\pm 0.3}$	$88.1_{\pm 0.6}$	$26.0_{\pm 0.8}$	$48.9_{\pm 0.6}$	$63.0_{\pm 0.4}$	$11.4_{\pm 0.3}$	$29.7_{\pm 0.3}$	$43.6_{\pm 0.4}$	$3.9_{\pm 0.2}$	$12.9_{\pm 0.4}$	$24.1_{\pm 0.3}$
IDM	-	-	-	$45.6_{\pm 0.7}$	$58.6_{\pm 0.1}$	$67.5_{\pm 0.1}$	$20.1_{\pm 0.3}$	$45.1_{\pm 0.1}$	$50.0_{\pm 0.2}$	**10.1**$_{\pm 0.2}$	**21.9**$_{\pm 0.2}$	$27.7_{\pm 0.3}$
DataDAM	-	-	-	$32.0_{\pm 1.2}$	$54.2_{\pm 0.8}$	$67.0_{\pm 0.4}$	$14.5_{\pm 0.5}$	$34.8_{\pm 0.5}$	$49.4_{\pm 0.3}$	$8.3_{\pm 0.4}$	$18.7_{\pm 0.3}$	$28.7_{\pm 0.3}$
DANCE	**81.5**$_{\pm 0.4}$	**86.3**$_{\pm 0.2}$	$86.9_{\pm 0.1}$	$47.1_{\pm 0.2}$	**70.8**$_{\pm 0.2}$	**76.1**$_{\pm 0.1}$	**27.9**$_{\pm 0.2}$	**49.8**$_{\pm 0.1}$	**52.8**$_{\pm 0.1}$	**11.6**$_{\pm 0.2}$	**26.4**$_{\pm 0.3}$	**28.9**$_{\pm 0.4}$
Whole Dataset	$93.5_{\pm 0.1}$			$84.8_{\pm 0.1}$			$56.2_{\pm 0.3}$			$37.6_{\pm 0.4}$		

Table 1: **Comparison with previous coreset selection and dataset condensation methods on low-resolution datasets and medium-resolution datasets.** IPC: image(s) per class. Ratio (%): the ratio of condensed examples to the whole training set. Best results are **highlighted** and the second best results are in **bold**. Note that some entries are marked as "-" because of scalability issues or the results are not reported.

	ImageNette		ImageWoof		ImageFruit		ImageMeow		ImageSquawk		ImageYellow	
IPC	1	10	1	10	1	10	1	10	1	10	1	10
Ratio (%)	0.105	1.050	0.110	1.100	0.077	0.77	0.077	0.77	0.077	0.77	0.077	0.77
Random	$23.5_{\pm 4.8}$	$47.7_{\pm 2.4}$	$14.2_{\pm 0.9}$	$27.0_{\pm 1.9}$	$13.2_{\pm 0.8}$	$21.4_{\pm 1.2}$	$13.8_{\pm 0.6}$	$29.0_{\pm 1.1}$	$21.8_{\pm 0.5}$	$40.2_{\pm 0.4}$	$20.4_{\pm 0.6}$	$37.4_{\pm 0.5}$
MTT	**47.7**$_{\pm 0.9}$	**63.0**$_{\pm 1.3}$	**28.6**$_{\pm 0.8}$	**35.8**$_{\pm 1.8}$	**26.6**$_{\pm 0.8}$	**40.3**$_{\pm 1.3}$	**30.7**$_{\pm 1.6}$	**40.4**$_{\pm 2.2}$	**39.4**$_{\pm 1.5}$	$52.3_{\pm 1.0}$	**45.2**$_{\pm 0.8}$	**60.0**$_{\pm 1.5}$
DM	$32.8_{\pm 0.5}$	$58.1_{\pm 0.3}$	$21.1_{\pm 1.2}$	$31.4_{\pm 0.5}$	-	-	-	-	$31.2_{\pm 0.7}$	$50.4_{\pm 1.2}$	-	-
DataDAM	$34.7_{\pm 0.9}$	$59.4_{\pm 0.4}$	$24.2_{\pm 0.5}$	$34.4_{\pm 0.4}$	-	-	-	-	$36.4_{\pm 0.8}$	**55.4**$_{\pm 0.9}$	-	-
DANCE	**57.2**$_{\pm 0.5}$	**80.2**$_{\pm 0.7}$	**30.6**$_{\pm 0.3}$	**57.8**$_{\pm 1.1}$	**30.6**$_{\pm 0.8}$	**52.8**$_{\pm 0.7}$	**39.4**$_{\pm 0.8}$	**60.4**$_{\pm 1.1}$	**52.0**$_{\pm 0.5}$	**77.2**$_{\pm 0.3}$	**51.8**$_{\pm 1.1}$	**78.8**$_{\pm 0.7}$
Whole Dataset	$87.4_{\pm 1.0}$		$67.0_{\pm 1.3}$		$63.9_{\pm 2.0}$		$66.7_{\pm 1.1}$		$87.5_{\pm 0.3}$		$84.4_{\pm 0.6}$	

Table 2: **Comparison with previous coreset selection and dataset condensation methods on high-resolution** (128×128) **Imagenet-Subsets.** All the datasets are condensed using a 5-layer ConvNet.

we utilize the resized TinyImageNET [Le and Yang, 2015], which has a resolution of 64×64. Furthermore, in alignment with MTT [Cazenavette *et al.*, 2022], we employ various subsets of the high-resolution ImageNet-1K [Deng *et al.*, 2009] dataset (resolution 128×128) in our experiments. These subsets include ImageNette, ImageWoof, ImageFruit, ImageWeow, ImageSquawk, and ImageYellow. Additional details about the datasets are provided in the Appendix.

Network Architectures. Following previous studies [Cazenavette *et al.*, 2022], we implement the condensation process using a ConvNet [Sagun *et al.*, 2018]. The ConvNet we employ consists of three identical convolutional blocks, each featuring a 128-kernel 3×3 convolutional layer, instance normalization, ReLU activation, and 3×3 average pooling with a stride of 2. For low-resolution datasets, we use a three-layer ConvNet, while a four-layer ConvNet is utilized for TinyImageNet. To accommodate the higher resolutions of the high-resolution ImageNet-1K subsets, we employ a five-layer ConvNet.

Evaluation Metric. We utilize the test accuracy of networks trained on the condensed set $\mathcal{D}_{\mathsf{syn}}$ as our primary evaluation metric. Each network is trained from scratch multiple times: 10 times for low-resolution datasets and TinyImageNet, and 3 times for the ImageNet-1K subsets. We report both the average accuracy and the standard deviation. To assess training efficiency, we consider run time per step and peak GPU memory usage as criteria, where the run time is calculated as an average over 1000 iterations.

Implementation Details. For training, we employ an SGD optimizer with a learning rate of 0.01, momentum of 0.9, and weight decay of 0.0005. The expert models $\boldsymbol{\theta}_{\mathsf{expert}}$ are trained for 60 epochs on low-resolution datasets and TinyImageNet, and for 80 epochs on ImageNet-1K subsets. We consistently use 5 expert models for all datasets as the default setting. The

Method	IPC	ConvNet-3	ResNet-10	DenseNet-121
DSA	10	$52.1_{\pm 0.5}$	$32.9_{\pm 0.3}$	$34.5_{\pm 0.1}$
	50	$60.6_{\pm 0.5}$	$49.7_{\pm 0.4}$	$49.1_{\pm 0.2}$
IDC	10	$67.5_{\pm 0.5}$	$63.5_{\pm 0.1}$	$61.6_{\pm 0.6}$
	50	$74.5_{\pm 0.1}$	$\mathbf{72.4}_{\pm 0.5}$	$\mathbf{71.8}_{\pm 0.6}$
MTT	10	$56.4_{\pm 0.7}$	$34.5_{\pm 0.8}$	$41.5_{\pm 0.5}$
	50	$65.9_{\pm 0.6}$	$43.2_{\pm 0.4}$	$51.9_{\pm 0.3}$
DM	10	$48.9_{\pm 0.6}$	$42.3_{\pm 0.5}$	$39.0_{\pm 0.1}$
	50	$63.0_{\pm 0.4}$	$58.6_{\pm 0.3}$	$57.4_{\pm 0.3}$
DANCE	10	$\mathbf{70.8}_{\pm 0.2}$	$\mathbf{67.0}_{\pm 0.2}$	$\mathbf{64.5}_{\pm 0.3}$
	50	$\mathbf{76.1}_{\pm 0.1}$	$68.0_{\pm 0.1}$	$64.8_{\pm 0.3}$

Table 3: **Cross-architecture generalization performance (%) on CIFAR-10.** The synthetic data is condensed using ConvNet-3 and evaluated using other architectures. The best results are in **bold**.

	IPC	DC	DSA	DM	MTT	IDM	**DANCE**
Run Time (Sec)	1	0.16	0.22	0.08	0.36	0.50	0.11
	10	3.31	4.47	0.08	0.40	0.48	0.12
	50	15.74	20.13	0.08	✗	0.58	0.12
GPU Memory (MB)	1	3515	3513	3323	2711	3223	2906
	10	3621	3639	3455	8049	3179	3045
	50	4527	4539	3605	✗	4027	3549

Table 4: **Time and GPU memeory cost comparison of SOTA datasets condensation methods.** Run Time: the time for a single iteration. GPU memory: the peak memory usage during condensing. Both run time and GPU memory is averaged over 1000 iterations. All experiments are conducted on CIFAR-10 with a single NVIDIA-A100 GPU. "✗" denotes out-of-memory issue.

number of iterations for Distribution Calibration is fixed at 1 across all datasets. During the condensing process, the SGD optimizer is set with a learning rate of 0.1 for ImageNet-1K subsets and 0.01 for other datasets, with the learning rate being scaled by the number of images per class (IPC). Following IDC [Kim *et al.*, 2022], we train the networks using a sequence of color transformation, cropping, and CutMix [Yun *et al.*, 2019]. The factor parameter l is set to 2 for low-resolution datasets and Tiny-ImageNet, and 3 for ImageNet-1K subsets. All synthetic data are initially generated from randomly selected real data to expedite optimization. The experiments are conducted on a GPU group comprising GTX 3090, RTX-2080, and NVIDIA-A100 GPUs.

4.2 Comparison to State-of-The-Art Methods

Baselines. We include a comprehensive range of methods as baselines in our study. For coreset selection methods, our choose Random Selection, Herding [Welling, 2009], and K-Center [Farahani and Hekmatfar, 2009]. In the category of Optimization-Oriented methods, we consider DC [Zhao and Bilen, 2021b], DSA [Zhao and Bilen, 2021a], IDC [Kim *et al.*, 2022], DREAM [Liu *et al.*, 2023b], and MTT [Cazenavette *et al.*, 2022]. Additionally, for Distribution-Matching-based methods, our baselines include CAFE and CAFE+DSA [Wang *et al.*, 2022], DM [Zhao and Bilen, 2023], IDM [Zhao *et al.*, 2023], and DataDAM [Sajedi *et al.*, 2023]. Further details about these baseline methods are provided in the Appendix, due to page constraints.

Performance Comparison. Tab. 1 and Tab. 2 present the comparison of our method with coreset selection methods and dataset condensation/distillation methods. The proposed method, DANCE, demonstrates remarkable performance across various datasets and resolutions. On low-resolution datasets such as Fashion-MNIST, CIFAR-10, CIFAR-100, and the medium-resolution dataset TinyImageNet, DANCE consistently outperforms or rivals leading methods in different IPC (images per class) settings. For instance, on Fashion-MNIST, it achieves the highest test accuracy of 81.5% with a single IPC. On CIFAR-10 and CIFAR-100, DANCE sets new benchmarks with 70.8% and 52.8% accuracy respectively at 50 IPC, even surpassing the SOTA optimization-oriented methods DREAM and IDC. Particularly notable is its performance on TinyImageNet, where it attains an accuracy of 11.6% at 1 IPC and 26.4% at 10 IPC, significantly ahead of the next best method, IDM. For the high-resolution ImageNet-1K subsets, DANCE still yield SOTA results across various scenarios. Remarkably, across all ImageNet-1K subsets with 10 images per class, our DANCE brings over 10% accuracy increase compared to the second best results, showcasing its efficacy in handling diverse image complexities. These results show the superiority of DANCE in dataset condensation tasks, especially considering the wide margin by which it leads in many categories. Overall, DANCE not only establishes new standards in dataset condensation but also demonstrates its robustness across varying resolutions and dataset complexities.

Cross-Architecture Evaluation. We also evaluated the performance of our condensed set across various architectures, as detailed in Tab. 3. The results demonstrate that DANCE excels not only on the architecture employed during the condensation process but also exhibits impressive generalization capabilities on a range of unseen architectures

Training Efficiency Evaluation. In the context of dataset condensation, it is of great importance to consider the resource and time costs, as extensively discussed in previous studies [Sajedi *et al.*, 2023; Zhang *et al.*, 2023]. Some of the methods entail significantly higher time costs in comparison to the time required for training the entire dataset, rendering them less than optimal in balancing effectiveness and efficiency. Our evaluation encompasses both time and peak GPU memory costs incurred during the condensation process for various baseline methods and DANCE. As presented in Tab. 4, DANCE exhibits remarkable efficiency compared to optimization-oriented methods such as DC [Zhao and Bilen, 2021b], DSA [Zhao and Bilen, 2021a], and MTT [Cazenavette *et al.*, 2022]. Much like DM [Zhao and Bilen, 2023], our method demonstrates scalability across different IPCs. However, IDM, being rooted in the DM-based approach, displays higher time costs when contrasted with both DM [Zhao and Bilen, 2023] and DANCE.

4.3 Ablation Studies

Effectiveness of Each Module. We evaluate three primal Modules of our method, namely Pseudo Long-Term Distribu-

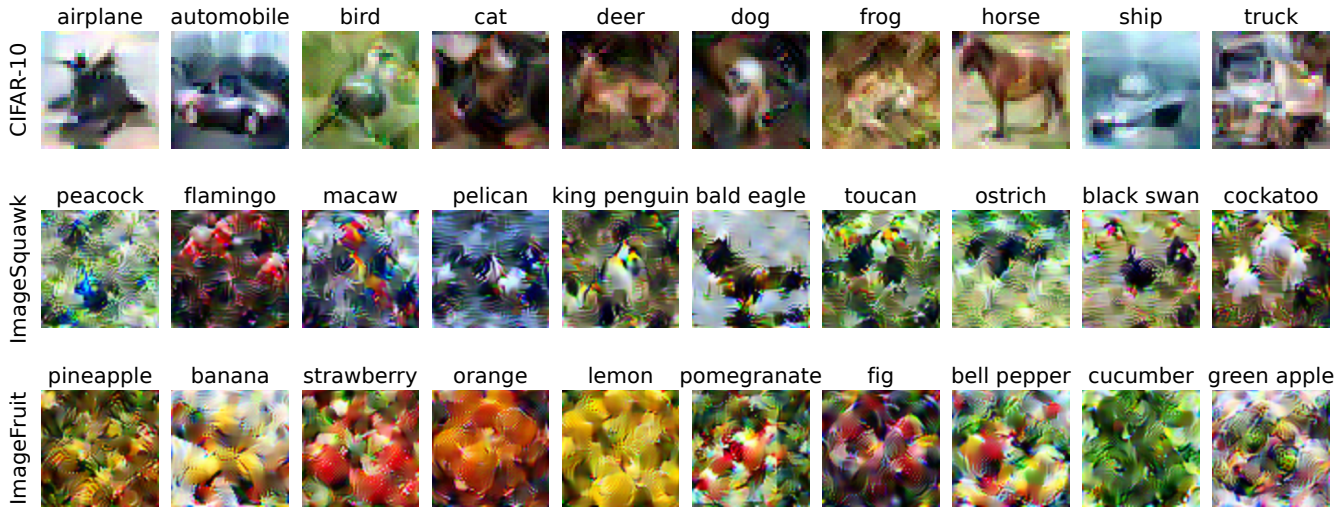


Figure 4: Example condensed images of 32×32 **CIFAR-10**, 128×128 **ImageSquawk**, and 128×128 **ImageFruit**.

Fac.	PLTDA	Dist. Calib.	CIFAR-10 10	CIFAR-10 50	CIFAR-100 10	CIFAR-100 50
-	✔	✔	$56.1_{\pm 0.2}$	$71.4_{\pm 0.4}$	$40.3_{\pm 0.2}$	$50.6_{\pm 0.1}$
✔	-	✔	$64.8_{\pm 0.1}$	$68.2_{\pm 0.1}$	$37.2_{\pm 0.1}$	$45.6_{\pm 0.2}$
✔	✔	-	$65.6_{\pm 0.3}$	$69.8_{\pm 0.2}$	$43.5_{\pm 0.3}$	$47.5_{\pm 0.2}$
✔	✔	✔	$\mathbf{70.8}_{\pm 0.2}$	$\mathbf{76.1}_{\pm 0.1}$	$\mathbf{49.8}_{\pm 0.1}$	$\mathbf{52.8}_{\pm 0.1}$

Table 5: **Ablation study on three main modules of DANCE.** "✔" denotes the module is included, and "-" ortherwise. "Fac." denotes the Factoring technique. "Dist. Calib." denotes the module of Distribution Calibration.

NEM	1	2	3	4	5	10	15	20
Acc.	69.2	70.1	70.2	70.2	70.8	71.2	71.1	71.1

Table 6: **Ablation on the number of expert models (NEM).** The evaluation is conducted on CIFAR-10 with 10 images per class.

tion Alignment (Sec. 3.1), Distribution Calibration (Sec. 3.2), and Factoring & Up-sampling technique (Sec. 3.3). As shown in Tab. 5, both the proposed PLTDA and Distribution Calibration bring significant improvement across various datasets. The most significant improvement is observed when all three modules are included. The results highlight the effectiveness of the three modules, demonstrating their collective importance in enhancing the DANCE framework's performance across different datasets.

Impact on the Number of Expert Models. The expert models $\boldsymbol{\theta}_{\text{expert}}$ are integral to both the PLTDA and Distribution Calibration modules within DANCE. To ascertain their impact, we investigated how the number of expert models (NEM) affects DANCE's performance. As Tab. 6 illustrates, there is a noticeable increase in DANCE's performance with the rise in NEM. Notably, even with just a single expert model, DANCE achieves competitive results, scoring 69.2% on CIFAR-10 with 10 images per class. This underscores DANCE's ability to efficiently leverage the pre-trained knowledge embedded in expert models.

4.4 Visualization Results

In Fig. 4, we present the synthetic images condensed by DANCE, showcasing distinct characteristics across different datasets. For the low-resolution dataset CIFAR-10, the condensed images are quite discernible, with each clearly representing its respective class. In contrast, the condensed images from the high-resolution ImageNet-1K subsets appear more abstract and outlined. Unlike the images produced by DM [Zhao and Bilen, 2023], which feature class-independent textures, our synthetic images encapsulate richer information pertinent to classification tasks. Additional visualizations are available in the Appendix due to page limitations.

5 Conclusion

In this study, we introduce a novel framework called **D**ual-view distribution **A**lignment for dataset **C**ondensation (DANCE), which enhances the Distribution Matching (DM) method by focusing on both inner- and inter-class views. DANCE consists of two meticulously designed modules: Pseudo Long-Term Distribution Alignment (PLTDA) for inner-class view and Distribution Calibration for inter-class view. PLTDA ensures that the data condensed by DANCE effectively represents its class throughout the entire training process while eliminating the need for persistent training. In contrast, Distribution Calibration maintains the synthetic data within its respective class region. Extensive experimental results on various datasets show that DANCE consistently surpasses state-of-the-art methods while requiring less computational costs. This makes DANCE highly suitable for practical and complex scenarios.

Acknowledgments

This work was partially supported by grants from the Pioneer R&D Program of Zhejiang Province (2024C01024).

References

[Cazenavette *et al.*, 2022] George Cazenavette, Tongzhou Wang, Antonio Torralba, et al. Dataset distillation by matching training trajectories. In *CVPR*, pages 4750–4759, 2022.

[Cazenavette *et al.*, 2023] George Cazenavette, Tongzhou Wang, Antonio Torralba, et al. Generalizing dataset distillation via deep generative prior. In *CVPR*, pages 3739–3748, 2023.

[Coleman *et al.*, 2020] Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *ICLR*, 2020.

[Cui *et al.*, 2022] Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Dc-bench: Dataset condensation benchmark. In *NeurIPS*, pages 810–822, 2022.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[Du *et al.*, 2023] Jiawei Du, Yidi Jiang, Vincent T. F. Tan, Joey Tianyi Zhou, and Haizhou Li. Minimizing the accumulated trajectory error to improve dataset distillation. In *CVPR*, pages 3749–3758, 2023.

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

[Geng *et al.*, 2023] Zongxion Geng, Jiahui andg Chen, Yuandou Wang, Herbert Woisetschlaeger, Sonja Schimmler, Ruben Mayer, Zhiming Zhao, and Chunming Rong. A survey on dataset distillation: Approaches, applications and future directions. In *IJCAI*, 2023.

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

[Jin *et al.*, 2022] Wei Jin, Xianfeng Tang, Haoming Jiang, Zheng Li, Danqing Zhang, Jiliang Tang, and Bing Yin. Condensing graphs via one-step gradient matching. In *ACM KDD*, pages 720–730, 2022.

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

[Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

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

[Li *et al.*, 2023] Shikun Li, Tongliang Liu, Jiyong Tan, Dan Zeng, and Shiming Ge. Trustable co-label learning from multiple noisy annotators. *IEEE TMM*, 25:1045–1057, 2023.

[Li *et al.*, 2024] Shikun Li, Xiaobo Xia, Jiankang Deng, Shiming Ge, and Tongliang Liu. Transferring annotator- and instance-dependent transition matrix for learning from crowds. *IEEE TPAMI*, pages 1–15, 2024.

[Liu *et al.*, 2021] Risheng Liu, Jiaxin Gao, Jin Zhang, et al. Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *IEEE TPAMI*, 44(12):10045–10067, 2021.

[Liu *et al.*, 2022] Songhua Liu, Kai Wang, Xingyi Yang, Jingwen Ye, and Xinchao Wang. Dataset distillation via factorization. In *NeurIPS*, pages 1100–1113, 2022.

[Liu *et al.*, 2023a] Songhua Liu, Jingwen Ye, Runpeng Yu, and Xinchao Wang. Slimmable dataset condensation. In *CVPR*, pages 3759–3768, 2023.

[Liu *et al.*, 2023b] Yanqing Liu, Jianyang Gu, Kai Wang, Zheng Zhu, Wei Jiang, and Yang You. DREAM: Efficient dataset distillation by representative matching. In *ICCV*, pages 17268–17278, 2023.

[Loo *et al.*, 2022] Noel Loo, Ramin Hasani, Alexander Amini, and Daniela Rus. Efficient dataset distillation using random feature approximation. In *NeurIPS*, pages 13877–13891, 2022.

[Nguyen *et al.*, 2021] Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. In *NeurIPS*, pages 5186–5198, 2021.

[Sagun *et al.*, 2018] Levent Sagun, Utku Evci, V. Ugur Güney, Yann N. Dauphin, and Léon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. In *ICLR Workshop*, 2018.

[Sajedi *et al.*, 2023] Ahmad Sajedi, Samir Khaki, Ehsan Amjadian, et al. Datadam: Efficient dataset distillation with attention matching. In *ICCV*, pages 17097–17107, 2023.

[Sorscher *et al.*, 2022] Ben Sorscher, Robert Geirhos, Shashank Shekhar, et al. Beyond neural scaling laws: beating power law scaling via data pruning. In *NeurIPS*, pages 19523–19536, 2022.

[Wang *et al.*, 2018] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation. *arXiv*, 2018.

[Wang *et al.*, 2022] Kai Wang, Bo Zhao, Xiangyu Peng, et al. CAFE: Learning to condense dataset by aligning features. In *CVPR*, pages 12196–12205, 2022.

[Wang *et al.*, 2023] Cheng Wang, Jiacheng Sun, Zhenhua Dong, Ruixuan Li, and Rui Zhang. Gradient matching for categorical data distillation in ctr prediction. In *CRS*, pages 161–170, 2023.

[Welling, 2009] Max Welling. Herding dynamical weights to learn. In *ICML*, pages 1121–1128, 2009.

[Xia *et al.*, 2022] Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *ICLR*, 2022.

[Xia *et al.*, 2024] Xiaobo Xia, Jiale Liu, Shaokun Zhang, Qingyun Wu, Hongxin Wei, and Tongliang Liu. Refined coreset selection: Towards minimal coreset size under model performance constraints. In *ICML*, 2024.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv*, 2017.

[Yang *et al.*, 2023] Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence. In *ICLR*, 2023.

[Yu *et al.*, 2024] Ruonan Yu, Songhua Liu, and Xinchao Wang. Dataset distillation: A comprehensive review. *IEEE TPAMI*, 46(1):150–170, 2024.

[Yun *et al.*, 2019] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, et al. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6022–6031, 2019.

[Zhang *et al.*, 2023] Lei Zhang, Jie Zhang, Bowen Lei, et al. Accelerating dataset distillation via model augmentation. In *CVPR*, pages 11950–11959, 2023.

[Zhang *et al.*, 2024a] Hansong Zhang, Shikun Li, Pengju Wang, Dan Zeng, and Shiming Ge. M3d: Dataset condensation by minimizing maximum mean discrepancy. In *AAAI*, 2024.

[Zhang *et al.*, 2024b] Hansong Zhang, Shikun Li, Dan Zeng, Chenggang Yan, and Shiming Ge. Coupled confusion correction: Learning from crowds with sparse annotations. In *AAAI*, 2024.

[Zhao and Bilen, 2021a] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *ICML*, pages 12674–12685, 2021.

[Zhao and Bilen, 2021b] Bo Zhao and Hakan Bilen. Dataset condensation with gradient matching. In *ICLR*, 2021.

[Zhao and Bilen, 2023] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *WACV*, pages 6503–6512, 2023.

[Zhao *et al.*, 2023] Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation. In *CVPR*, pages 7856–7865, 2023.

[Zheng *et al.*, 2023] Haizhong Zheng, Rui Liu, Fan Lai, and Atul Prakash. Coverage-centric coreset selection for high pruning rates. In *ICLR*, 2023.

[Zhou *et al.*, 2023] Daquan Zhou, Kai Wang, Jianyang Gu, et al. Dataset quantization. In *ICCV*, pages 17205–17216, 2023.