M3D: Dataset Condensation by Minimizing Maximum Mean Discrepancy

Hansong Zhang[1, 2*], **Shikun Li**[1, 2*], **Pengju Wang**[1, 2], **Dan Zeng**[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]Department of Communication Engineering, Shanghai University, Shanghai 200040, China
{zhanghansong,lishikun,wangpengju,geshiming}@iie.ac.cn, dzeng@shu.edu.cn

Abstract

Training state-of-the-art (SOTA) deep models often requires extensive data, resulting in substantial training and storage costs. To address these challenges, dataset condensation has been developed to learn a small synthetic set that preserves essential information from the original large-scale dataset. Nowadays, optimization-oriented methods have been the primary method in the field of dataset condensation for achieving SOTA results. However, the bi-level optimization process hinders the practical application of such methods to realistic and larger datasets. To enhance condensation efficiency, previous works proposed Distribution-Matching (DM) as an alternative, which significantly reduces the condensation cost. Nonetheless, current DM-based methods still yield less comparable results to SOTA optimization-oriented methods. In this paper, we argue that existing DM-based methods overlook the higher-order alignment of the distributions, which may lead to sub-optimal matching results. Inspired by this, we present a novel DM-based method named M3D for dataset condensation by **M**inimizing the **M**aximum **M**ean **D**iscrepancy between feature representations of the synthetic and real images. By embedding their distributions in a reproducing kernel Hilbert space, we align all orders of moments of the distributions of real and synthetic images, resulting in a more generalized condensed set. Notably, our method even surpasses the SOTA optimization-oriented method IDC on the high-resolution ImageNet dataset. Extensive analysis is conducted to verify the effectiveness of the proposed method. Source codes are available at https://github.com/Hansong-Zhang/M3D.

Introduction

In the era of deep learning, the utilization of large-scale datasets comprising millions of samples has become an indispensable prerequisite for achieving state-of-the-art (SOTA) models (Zhao and Bilen 2021a; Xia et al. 2022). However, the associated storage expenses and computational costs involved in training these models present formidable challenges, often rendering them beyond the reach of startups and non-profit organizations (Wang et al. 2018; Coleman et al. 2019; Sorscher et al. 2022).

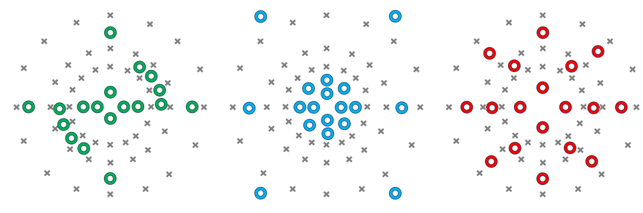


Figure 1: Illustration of the importance of the higher-order alignment of distributions, where circles represent the representations of synthesized examples while crosses represent the representations of original examples. (a) The misaligned distributions with different second-order moments; (b) the misaligned distributions with different third-order moments; (c) the aligned distributions.

To alleviate the challenges associated with larger datasets, Dataset Condensation (DC) (Wang et al. 2018) has emerged to reduce the training cost by synthesizing a compact set of informative images. Since its proposal, DC has attracted significant attention for addressing the challenges posed by the data burden (Cazenavette et al. 2022; Zhao and Bilen 2021b; Kim et al. 2022; Wang et al. 2022; Zhao and Bilen 2023). Typically, DC condenses the dataset by minimizing the distance between real and synthetic images via a pre-defined metric. Based on whether to perform a costly bi-level optimization (Liu et al. 2021), these methods can be generally categorized into two groups: (1) *Optimization-Oriented* methods (Zhao and Bilen 2021b; Kim et al. 2022; Zhao and Bilen 2021a; Cazenavette et al. 2022), which usually generate condensed examples by conducting performance matching or parameter matching via a bi-level optimization (Yu, Liu, and Wang 2023); (2) *Distribution-Matching(DM)-based* methods (Wang et al. 2022; Zhao and Bilen 2023), which focus on aligning the feature distributions between real and synthetic data. Optimization-oriented methods have faced criticism for their inefficiency, primarily due to the involvement of bi-level optimization modules and time-consuming network updating processes (Zhang et al. 2023; Wang et al. 2022; Zhao and Bilen 2023). In contrast, DM-based methods do not involve such nested optimization of models, which significantly reduces the computa-

*These authors contributed equally.
†Corresponding Author

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

tional cost associated with dataset condensation. Nevertheless, the informativeness of the condensed examples generated by current DM-based methods may not be as comparable to those produced by optimization-oriented methods.

In this paper, we address a crucial oversight in existing DM-based methods, which is their neglect of higher-order moments of the distribution. As illustrated in Fig. 1, despite sharing the same first moment, the representation distributions of original and synthetic examples with misaligned second-order moments (Fig. 1a) or third-order moments (Fig. 1b) can exhibit much distinct characteristics. Motivated by this issue, we propose a novel DM-based method involving **M**inimizing the **M**aximum **M**ean **D**iscrepancy (**M3D**) between the representation distributions of the real and synthetic images. Unlike previous DM-based methods that solely embed images in a feature representation space and align the first moment, our method further embeds the distribution of feature representations into a reproducing kernel Hilbert space. This transformation allows us to represent the infinite order of moments in a kernel-function form. By leveraging empirical estimation, we can readily align both first- and higher-order moments of the real and synthetic data with theoretical guarantees. Our method not only maintains the efficiency of the DM-based method but also exhibits significant improvements. Remarkably, the efficiency of our method makes it easily applicable to realistic and larger datasets like ImageNet (Deng et al. 2009).

Before delving into technical details, we clearly emphasize our contribution as:

- We reveal the importance of the alignment of higher-order moments for distribution matching, which is overlooked by previous DM-based methods.
- We propose a theoretical-guaranteed method for dataset condensation named M3D, which applies the classical kernel method to represent an infinite number of moments in a kernel-function form, enabling the improved alignment of the higher-order moments of the representation distributions.
- We conduct extensive experiments to demonstrate the effectiveness and efficiency of our proposed method, where M3D yields SOTA performance with strong generalization across various scenarios.

Background

Problem Fromulation. Dataset Condensation (DC) (Wang et al. 2018), also called dataset distillation, targets to condense a large-scale dataset $\mathcal{T} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{|\mathcal{T}|}$ into a tiny dataset $\mathcal{S} = \{(\boldsymbol{s}_j, y_j)\}_{j=1}^{|\mathcal{S}|}$, so that an arbitrary model trained on $\mathcal{S}$ achieves comparable performance to the one trained on $\mathcal{T}$. Typically, the condensed $\mathcal{S}$ is obtained by minimizing the information loss between the synthesized and the original examples, which can be formulated as:

$$\mathcal{S}^\star = \arg\min_{\mathcal{S}} \boldsymbol{D}(\phi(\mathcal{T}), \phi(\mathcal{S})), \tag{1}$$

where $\boldsymbol{D}$ represents a distance metric such as Mean Square Error (MSE), and ϕ denotes the matching objective. As mentioned before, various objectives can lead to different optimization processes (Yu, Liu, and Wang 2023), and based on whether to perform a costly bi-level optimization, existing methods can be mainly divided into optimization-oriented methods and Distribution-Matching (DM)-based methods. [1]

Distribution Matching. Although optimization-oriented methods can achieve the SOTA performance, the inefficiency of them poses a significant obstacle to their application in realistic and larger datasets (Zhang et al. 2023). In response, DM-based methods have been developed as an alternative. In their pioneering work, DM (Zhao and Bilen 2023) introduces a surrogate matching objective that focuses on aligning the representation distributions of $\mathcal{S}$ and $\mathcal{T}$. This objective can be formulated as:

$$\mathcal{S}^\star = \arg\min_{\mathcal{S}} E_{\theta\sim P_\theta}\left[\boldsymbol{D}(g_\theta(\mathcal{S}), g_\theta(\mathcal{T}))\right], \tag{2}$$

where g_θ is the deep encoder network parameterized as θ, which is instanced by the model f_θ without the output layer. With MSE as the distance metric, the training objective of DM can be reformulated as:

$$\mathcal{S}^\star = \arg\min_{\mathcal{S}} E_{\theta\sim P_\theta}\Big\|\frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|} g_\theta(\boldsymbol{x}_i) - \frac{1}{|\mathcal{S}|}\sum_{j=1}^{|\mathcal{S}|} g_\theta(\boldsymbol{s}_j)\Big\|^2, \tag{3}$$

which works as minimizing the gap between empirical first moment of the representation distributions between $\mathcal{S}$ and $\mathcal{T}$. Compared to previous optimization-oriented methods, DM (Zhao and Bilen 2023) eliminates the need for network updating, relying instead on randomly initialized encoders. Furthermore, the costly bi-level optimization is avoided in DM, leading to significantly improved training efficiency.

Remark. Given the lower effectiveness of DM compared to optimization-oriented SOTA methods, efforts have been made to enhance DM and generate more informative examples in previous works (Zhao et al. 2023a; Sajedi et al. 2023). For instance, IDM (Zhao et al. 2023a) enhances DM through techniques such as partitioning, enriched model sampling, and class-aware regularization. Similarly, DataDAM (Sajedi et al. 2023) improves DM by incorporating attention matching. In contrast to these methods where only the first-order moment is matched, our focus is on enhancing DM through distribution embedding and higher-order moments, which are also noticed but not addressed explicitly by IDM (Zhao et al. 2023a).

Reproducing Kernel Hilbert Space. We provide a brief recap of the Reproducing Kernel Hilbert Space (RKHS) (Muandet et al. 2017; Smola et al. 2007; Borgwardt et al. 2006) here, which serves as the foundation of our method.

Definition 1 *Given a kernel $\mathcal{K}$, $\mathcal{H}$ is a Hilbert space of functions $\mathcal{X} \to \mathbb{R}$ with dot product $\langle\cdot,\cdot\rangle$, if $\forall\phi$, satisfying the reproducing property:*

$$\langle\phi(\cdot), \mathcal{K}(x,\cdot)\rangle = \phi(x). \tag{4}$$

[1]Note that the introduction about more dataset condensation works can be found in the Appendix.

	1st-order (mean)	2nd-order (variance)	3rd-order (skewness)	Test Acc.
DM	4.91	7.37	6.76	48.9
+2nd Reg.	4.51	6.96	6.35	52.1 (↑ 3.2)
+2nd & 3rd Reg.	3.69	6.14	5.94	53.9 (↑ 5.0)
M3D	**0.82**	**1.23**	**1.64**	**63.5** (↑ 14.6)

Table 1: The distance between the moments of the condensed set and the original training set. "+2nd(3rd) Reg." denotes adding the regularization of aligning the 2nd(3rd)-order moment to the original loss of DM.

That is to say, with the RKHS, we can map a function f on $\mathcal{X}$ to its value at x as an inner product. In addition to the reproducing property mentioned above, the kernel function $\mathcal{K}$ must also satisfy the following two properties:

$$\textbf{Symmetry}: \quad \mathcal{K}(x, x') = \mathcal{K}(x', x)$$
$$\textbf{Positive}: \quad \mathcal{K}(\cdot, \cdot) \geq 0$$

Commonly used kernel function include the polynomial kernel $\mathcal{K}(x, x') = (x^\intercal x' + c)^d$, the Gaussian RBF kernel $\mathcal{K}(x, x') = \exp(-\lambda\|x - x'\|^2)$, and the Linear kernel $\mathcal{K}(x, x') = x^\intercal x'$.

Methodology

In this section, we begin by analyzing the importance of the alignment of higher-order moments for distribution matching. Subsequently, we propose our method M3D by exploiting the classical kernel method (Altun and Smola 2006; Borgwardt et al. 2006) to align the higher-order moments of the representation distributions between real and synthesized data with theoretical guarantees.

Importance of the Higher-Order Alignment

As shown in Eq. (3), it is evident that DM (Zhao and Bilen 2023) only considers aligning the first moment (mean) of the representation distributions, while neglecting higher-order moments. At a high level, it may lead to the higher-order misalignment of the representation distributions of its condensed data and original data.

To investigate this misalignment issue and highlight the importance of the higher-order alignment, we assessed the moment distances between the condensed set and the original training set on CIFAR-10 with 10 images per class. This was done by incorporating higher-order moment regularization terms into the original loss of DM (Zhao and Bilen 2023). The results, presented in Table 1, reveal that adding second-order regularization notably decreases the distance between higher-order moments of the condensed and original data, underscoring the inadequacy of aligning only the first moment. Furthermore, performing more regularization enhances the condensed dataset's performance through improved higher-order alignment. These results underscore the critical role of higher-order moment alignment in distribution matching, which is neglected in previous works.

Minimizing Maximum Mean Discrepancy

From the preceding analysis, it becomes evident that perfecting distribution matching necessitates the consideration of higher-order moments. While incorporating higher-order regularizations directly aids in aligning these moments, it is limited to finite moments. Moreover, tuning the regularization coefficient becomes increasingly challenging with a growing number of regularization terms. In this subsection, we represent a new DM-based method that aligns the infinite order of moments in a kernel-function form. We depict the framework of the proposed M3D in Fig. 2.

Embedding Distribution in RKHS. Denoting the distribution of representations for real and synthetic examples as $g_\theta(\mathcal{T}) \sim \boldsymbol{P}_\mathcal{T}$ and $g_\theta(\mathcal{S}) \sim \boldsymbol{P}_\mathcal{S}$ respectively, where g_θ denotes the representation extractor parameterized by θ. As the order of moments extends infinitely, it is impractical to explicitly align an infinite number of moments. To address this, we need to first embed the distribution in an RKHS $\mathcal{H}$:

$$\mu[\boldsymbol{P}_{\mathcal{T}/\mathcal{S}}] := E_{\mathcal{T}/\mathcal{S}}[\mathcal{K}(g_\theta(\boldsymbol{x}/\boldsymbol{s}), \cdot)], \tag{5}$$

which has been proven to be a valid embedding for distance based on the following theorem:

Theorem 1 *(Fukumizu, Bach, and Jordan 2004) If the kernel function $\mathcal{K}$ is universal, then the mean map $\mu := \boldsymbol{P} \to \mu[\boldsymbol{P}]$ is injective.*

Maximum Mean Discrepancy. Via the reproducing property of $\mathcal{H}$, $\forall\phi$, we have

$$\langle \phi, \mu[\boldsymbol{P}_{\mathcal{T}/\mathcal{S}}]\rangle = E_{\mathcal{T}/\mathcal{S}}[\phi(g_\theta(\boldsymbol{x}/\boldsymbol{s}))], \tag{6}$$

which indicate that we can compute expectations w.r.t. $\boldsymbol{P}_{\mathcal{T}/\mathcal{S}}$ by taking the inner product with the distribution kernel embedding $\mu[\boldsymbol{P}_{\mathcal{T}/\mathcal{S}}]$. This property is favorable because it helps us to calculate the Maximum Mean Discrepancy (MMD) between $\boldsymbol{P}_\mathcal{T}$ and $\boldsymbol{P}_\mathcal{S}$:

$$\begin{aligned}\mathrm{MMD}(\boldsymbol{P}_\mathcal{T}, \boldsymbol{P}_\mathcal{S}) :&= \sup(E_\mathcal{T}[\phi(g_\theta(\boldsymbol{x}))] - E_\mathcal{S}[\phi(g_\theta(\boldsymbol{s}))]) \\ &= \sup\langle \phi, \mu[\boldsymbol{P}_\mathcal{T}] - \mu[\boldsymbol{P}_\mathcal{S}]\rangle,\end{aligned}$$

where $\phi \in \mathcal{H}$ and $\|\phi\|_\mathcal{H} \leq 1$. In addition, based on the Cauchy-Schwarz inequality, we have $\langle \phi, \mu[\boldsymbol{P}_\mathcal{T}] - \mu[\boldsymbol{P}_\mathcal{S}]\rangle \leq \|\phi\|_\mathcal{H}\|\mu[\boldsymbol{P}_\mathcal{T}] - \mu[\boldsymbol{P}_\mathcal{S}]\|_\mathcal{H} \leq \|\mu[\boldsymbol{P}_\mathcal{T}] - \mu[\boldsymbol{P}_\mathcal{S}]\|_\mathcal{H}$, hence the MMD can be further simplified as:

$$\mathrm{MMD}(\boldsymbol{P}_\mathcal{T}, \boldsymbol{P}_\mathcal{S}) = \|\mu[\boldsymbol{P}_\mathcal{T}] - \mu[\boldsymbol{P}_\mathcal{S}]\|. \tag{7}$$

It should be noted that $\mu[\boldsymbol{P}_\mathcal{T}]$ and $\mu[\boldsymbol{P}_\mathcal{S}]$ are characterized by infinite-dimensional spaces, which renders direct computation unattainable. However, we can leverage the reproducing property of the RKHS to transform them into a more tractable form using the kernel function $\mathcal{K}$. This transformation can be formally expressed as:

$$\mathrm{MMD}^2(\boldsymbol{P}_\mathcal{T}, \boldsymbol{P}_\mathcal{S}) = \mathcal{K}_{\mathcal{T},\mathcal{T}} + \mathcal{K}_{\mathcal{S},\mathcal{S}} - 2\mathcal{K}_{\mathcal{T},\mathcal{S}}, \tag{8}$$

where $\mathcal{K}_{X,Y} = E_{X,Y}[\mathcal{K}(g_\theta(x), g_\theta(y))]$ with $x \sim X, y \sim Y$. Due to limited page, we provide the derivation of Eq. (8) in the Appendix. Last, note that we only have access to

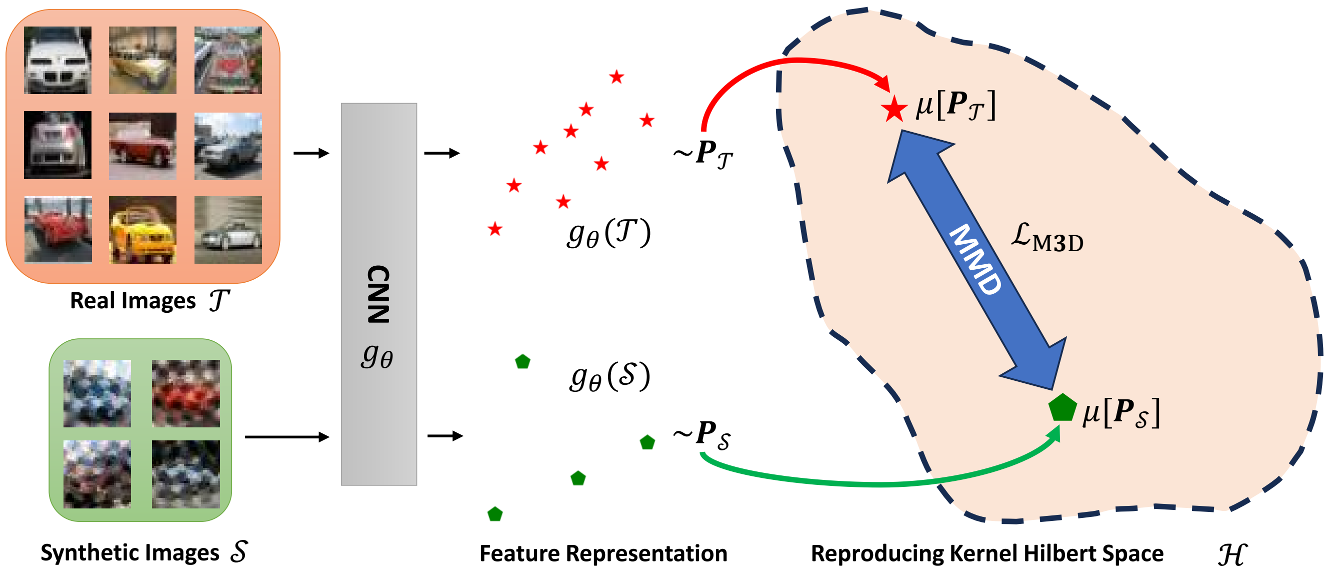


Figure 2: The framework of M3D. After extracting the representations via a encoder network, the distributions of real and synthetic representations are further embedded in the Reproducing Kernel Hilbert Space (RKHS), where the M3D loss $\mathcal{L}_{\text{M3D}}$ is calculated to guild the update of synthetic examples for higher-order distribution alignment.

the datasets $\mathcal{T}$ and $\mathcal{S}$ rather than their underlying distributions. In order to tackle this issue, denoting the empirical approximation of $\mu[\boldsymbol{P}_{\mathcal{T}}]$ and $\mu[\boldsymbol{P}_{\mathcal{S}}]$ as $\mu[\mathcal{T}] = \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\mathcal{K}(g_\theta(\boldsymbol{x}_i),\cdot)$, $\mu[\mathcal{S}] = \frac{1}{|\mathcal{S}|}\sum_{j=1}^{|\mathcal{S}|}\mathcal{K}(g_\theta(\boldsymbol{s}_j),\cdot)$ respectively, we introduce the following theorem:

Theorem 2 *(Altun and Smola 2006) Assume that $\|\phi\|_\infty \leq R$ for all $\phi \in \mathcal{H}$ with $\|\phi\|_{\mathcal{H}} \leq 1$. Then with probability at least $1-\delta$, $\|\mu[\boldsymbol{P}_{\mathcal{T}/\mathcal{S}}] - \mu[\mathcal{T}/\mathcal{S}]\| \leq 2\bar{R}(\mathcal{H}, \boldsymbol{P}_{\mathcal{T}/\mathcal{S}}) + R\sqrt{-|\mathcal{T}/\mathcal{S}|^{-1}\log(\delta)}$, where $\bar{R}(\mathcal{H}, \boldsymbol{P}_{\mathcal{T}/\mathcal{S}})$ is the Rademacher average which is ensured to yield error of $\mathcal{O}(\sqrt{|\mathcal{T}/\mathcal{S}|^{-1}})$.*

Theorem 2 guarantees that the empirical approximations $\mu[\mathcal{T}/\mathcal{S}]$ are good proxies for $\mu[\boldsymbol{P}_{\mathcal{T}/\mathcal{S}}]$. Therefore, we can modify Eq. (8) to the following empirical form as the M3D loss:

$$\mathcal{L}_{\text{M3D}} = \widehat{\text{MMD}}^2(\boldsymbol{P}_{\mathcal{T}}, \boldsymbol{P}_{\mathcal{S}}) = \hat{\mathcal{K}}_{\mathcal{T},\mathcal{T}} + \hat{\mathcal{K}}_{\mathcal{S},\mathcal{S}} - 2\hat{\mathcal{K}}_{\mathcal{T},\mathcal{S}}, \quad (9)$$

where $\hat{\mathcal{K}}_{X,Y} = \frac{1}{|X|\cdot|Y|}\sum_{i=1}^{|X|}\sum_{j=1}^{|Y|}\mathcal{K}(g_\theta(x_i), g_\theta(y_j))$ with $\{x_i\}_{i=1}^{|X|} \sim X, \{y_j\}_{j=1}^{|Y|} \sim Y$. Based on the analysis above, we have successfully achieved the transformation of an infinite number of moments into a finite form using RKHS. As shown in Table 1, this transformation allows us to effectively align the distributions between $\mathcal{T}$ and $\mathcal{S}$ during the condensing process.

Training Algorithm of M3D

The pseudo-code of M3D is provided in the Appendix. In addition to the kernel method, we exploit the following two techniques to enhance the distribution matching.

Factor & Up-sampling. The factor technique (Kim et al. 2022), also termed as partitioning and expansion augmentation in IDM (Zhao et al. 2023a), aims to increase the number of representations extracted from $\mathcal{S}$ without additional storage cost. Specifically, with the factor parameter being l, each image $\boldsymbol{s}_i \in \mathcal{S}$ is factorized into $l \times l$ mini-examples and then up-sampled to its original size in training:

$$\boldsymbol{s}_i \xrightarrow{\text{Factor}} \begin{bmatrix} \boldsymbol{s}_i^{1,1} & \dots & \boldsymbol{s}_i^{1,l} \\ \vdots & \ddots & \vdots \\ \boldsymbol{s}_i^{l,1} & \dots & \boldsymbol{s}_i^{l,l} \end{bmatrix} \xrightarrow{\text{Up-sample}} \{\boldsymbol{s}_i'^{1}, \boldsymbol{s}_i'^{2}, \dots, \boldsymbol{s}_i'^{l\times l}\}. \quad (10)$$

In this way, the storage space of $\mathcal{S}$ can be further leveraged. Following previous works, the same factor technique is incorporated into our framework, where we further exploit its benefits in aligning distributions in higher-order moments.

Iteration per Random Model. Following DM (Zhao and Bilen 2023), we employ multiple randomly initialized models to extract representation embeddings from both $\mathcal{T}$ and $\mathcal{S}$. In contrast to DM, where only a single-step iteration is performed for each model, we posit that relying solely on the representation distributions of one batch of real and synthetic examples may introduce matching biases. To address this, without incurring additional memory usage, we empirically observe that conducting multiple iterations per model (IPM) enhances the performance of the condensed set.

Experiments

In this section, we begin by comparing our proposed M3D with SOTA baselines on multiple benchmark datasets. Subsequently, we conduct an in-depth examination of M3D through ablation analysis.

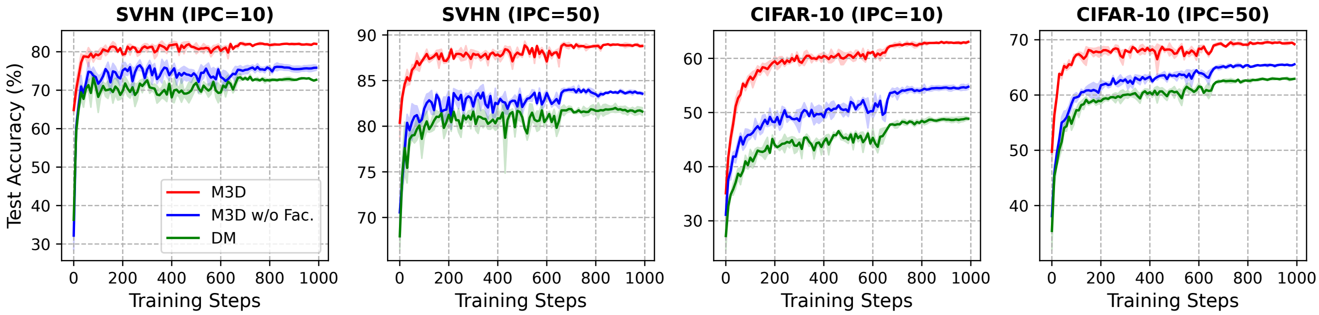


Figure 3: Performance comparison between M3D and DM across varying training steps. M3D w/o Fac denotes the M3D without using the factor technique.

Dataset	IPC	Ratio (%)	Coreset Selection			Dataset Condensation							Whole
			Random	Herding	K-Center	DC	DSA	CAFE	CAFE+DSA	DM	IDM	**M3D**	
MNIST	1	0.017	64.9$_{\pm 3.5}$	89.2$_{\pm 1.6}$	89.3$_{\pm 1.5}$	91.7$_{\pm 0.5}$	88.7$_{\pm 0.6}$	93.1$_{\pm 0.3}$	90.8$_{\pm 0.5}$	89.7$_{\pm 0.6}$	-	**94.4**$_{\pm 0.2}$	
	10	0.17	95.1$_{\pm 0.9}$	93.7$_{\pm 0.3}$	84.4$_{\pm 1.7}$	97.4$_{\pm 0.2}$	**97.8**$_{\pm 0.1}$	97.2$_{\pm 0.2}$	97.5$_{\pm 0.1}$	97.5$_{\pm 0.1}$	-	97.6$_{\pm 0.1}$	99.6$_{\pm 0.0}$
	50	0.83	97.9$_{\pm 0.2}$	94.8$_{\pm 0.2}$	97.4$_{\pm 0.3}$	98.8$_{\pm 0.2}$	**99.2**$_{\pm 0.1}$	98.6$_{\pm 0.2}$	98.9$_{\pm 0.2}$	98.6$_{\pm 0.1}$	-	98.2$_{\pm 0.2}$	
F-MNIST	1	0.017	51.4$_{\pm 3.8}$	67.0$_{\pm 1.9}$	66.9$_{\pm 1.8}$	70.5$_{\pm 0.6}$	70.6$_{\pm 0.6}$	77.1$_{\pm 0.9}$	73.7$_{\pm 0.7}$	70.7$_{\pm 0.6}^{\dagger}$	-	**80.7**$_{\pm 0.3}$	
	10	0.17	73.8$_{\pm 0.7}$	71.1$_{\pm 0.7}$	54.7$_{\pm 1.5}$	82.3$_{\pm 0.4}$	84.6$_{\pm 0.3}$	83.0$_{\pm 0.4}$	83.0$_{\pm 0.3}$	83.5$_{\pm 0.3}^{\dagger}$	-	**85.0**$_{\pm 0.1}$	93.5$_{\pm 0.1}$
	50	0.83	82.5$_{\pm 0.7}$	71.9$_{\pm 0.8}$	68.3$_{\pm 0.8}$	83.6$_{\pm 0.4}$	**88.7**$_{\pm 0.2}$	84.8$_{\pm 0.4}$	88.2$_{\pm 0.3}$	88.1$_{\pm 0.6}^{\dagger}$	-	86.2$_{\pm 0.3}$	
SVHN	1	0.014	14.6$_{\pm 1.6}$	20.9$_{\pm 1.3}$	21.0$_{\pm 1.5}$	31.2$_{\pm 1.4}$	27.5$_{\pm 1.4}$	42.6$_{\pm 3.3}$	42.9$_{\pm 3.0}$	30.3$_{\pm 0.1}^{\dagger}$	-	**62.8**$_{\pm 0.5}$	
	10	0.14	35.1$_{\pm 4.1}$	50.5$_{\pm 3.3}$	14.0$_{\pm 1.3}$	76.1$_{\pm 0.6}$	79.2$_{\pm 0.5}$	75.9$_{\pm 0.6}$	77.9$_{\pm 0.6}$	73.5$_{\pm 0.5}^{\dagger}$	-	**83.3**$_{\pm 0.7}$	95.4$_{\pm 0.1}$
	50	0.7	70.9$_{\pm 0.9}$	72.6$_{\pm 0.8}$	20.1$_{\pm 1.4}$	82.3$_{\pm 0.3}$	84.4$_{\pm 0.4}$	81.3$_{\pm 0.3}$	82.3$_{\pm 0.4}$	82.0$_{\pm 0.2}^{\dagger}$	-	**89.0**$_{\pm 0.2}$	
CIFAR-10	1	0.02	14.4$_{\pm 2.0}$	21.5$_{\pm 1.2}$	21.5$_{\pm 1.3}$	28.3$_{\pm 0.5}$	28.8$_{\pm 0.7}$	30.3$_{\pm 1.1}$	31.6$_{\pm 0.8}$	26.0$_{\pm 0.8}$	**45.6**$_{\pm 0.7}$	45.3$_{\pm 0.3}$	
	10	0.2	26.0$_{\pm 1.2}$	31.6$_{\pm 0.7}$	14.7$_{\pm 0.9}$	44.9$_{\pm 0.5}$	52.1$_{\pm 0.5}$	46.3$_{\pm 0.6}$	50.9$_{\pm 0.5}$	48.9$_{\pm 0.6}$	58.6$_{\pm 0.1}$	**63.5**$_{\pm 0.2}$	84.8$_{\pm 0.1}$
	50	1	43.4$_{\pm 1.0}$	40.4$_{\pm 0.6}$	27.0$_{\pm 1.4}$	53.9$_{\pm 0.5}$	60.6$_{\pm 0.5}$	55.5$_{\pm 0.6}$	62.3$_{\pm 0.4}$	63.0$_{\pm 0.4}$	67.5$_{\pm 0.1}$	**69.9**$_{\pm 0.5}$	
CIFAR-100	1	0.2	4.2$_{\pm 0.3}$	8.4$_{\pm 0.3}$	8.3$_{\pm 0.3}$	12.8$_{\pm 0.3}$	13.9$_{\pm 0.3}$	12.9$_{\pm 0.3}$	14.0$_{\pm 0.3}$	11.4$_{\pm 0.3}$	20.1$_{\pm 0.3}$	**26.2**$_{\pm 0.3}$	
	10	2	14.6$_{\pm 0.5}$	17.3$_{\pm 0.3}$	7.1$_{\pm 0.2}$	25.2$_{\pm 0.3}$	32.3$_{\pm 0.3}$	27.8$_{\pm 0.3}$	31.5$_{\pm 0.2}$	29.7$_{\pm 0.3}$	**45.1**$_{\pm 0.1}$	42.4$_{\pm 0.2}$	56.2$_{\pm 0.3}$
	50	10	30.0$_{\pm 0.4}$	33.7$_{\pm 0.5}$	30.5$_{\pm 0.3}$	-	42.8$_{\pm 0.4}$	37.9$_{\pm 0.3}$	42.9$_{\pm 0.2}$	43.6$_{\pm 0.4}$	50.0$_{\pm 0.2}$	**50.9**$_{\pm 0.7}$	

Table 2: Comparison with previous coreset selection and dataset condensation methods on low-resolution datasets. All the datasets are condensed using a 3-layer ConvNet. IPC: image(s) per class. Ratio (%): the ratio of condensed examples to the whole training set. "†" denotes the result is reproduced by us. Best results are in bold. Note that some entries are marked as "-" because of scalability issues or the results are not reported.

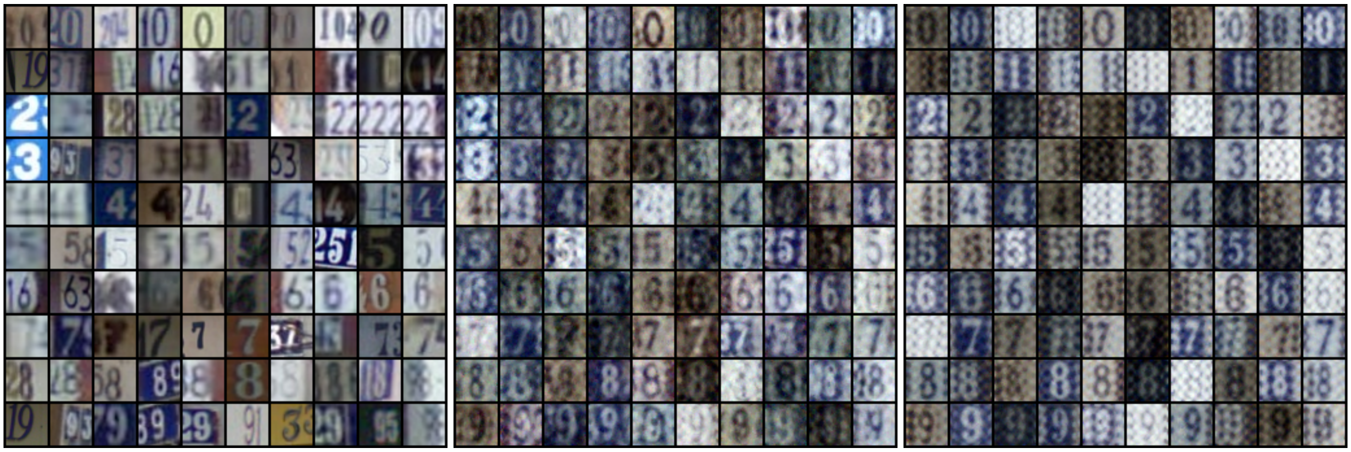
Experimental Setups

Datasets. We evaluate the classification performance of networks trained on synthetic images that have been condensed using various baselines as well as our proposed method M3D. Our evaluation encompasses five low-resolution datasets: MNIST (LeCun et al. 1998), Fashion-MNIST (F-MNIST) (Xiao, Rasul, and Vollgraf 2017), SVHN (Netzer et al. 2011), CIFAR-10 (Krizhevsky, Hinton et al. 2009), and CIFAR-100 (Krizhevsky, Hinton et al. 2009). In addition, we also conduct experiments on the high-resolution dataset ImageNet subsets (Deng et al. 2009). Detailed descriptions of datasets can be found in the Appendix.

Network Architectures. We use a depth-3 ConvNet (Sagun et al. 2017) for the low-resolution datasets, and a ResNetAP-10 (Kim et al. 2022) (ResNet-10 with the strided convolution replaced by average pooling) for the high-resolution ImageNet subsets.

Baselines. We employ an extensive range of methods as baselines for comparison. Regarding coreset selection methods, we consider the following: (1) Random, (2) Herding (Welling 2009), and (3) K-Center (Farahani and Hekmatfar 2009; Sener and Savarese 2017). For optimization-oriented DC methods, we evaluate (4) DC (Zhao and Bilen 2021b), (5) DSA (Zhao and Bilen 2021a), (6) IDC (Kim et al. 2022). On the other hand, for DM-based DC methods, we include (7) CAFE (Wang et al. 2022), (8) its variant CAFE+DSA (Wang et al. 2022), (9) DM (Zhao and Bilen 2023) and (10) IDM (Zhao et al. 2023a). We provide detailed descriptions of baselines in the Appendix.

Metric. Following previous works (Wang et al. 2018; Zhao and Bilen 2023; Kim et al. 2022), we employ the test accuracy of networks trained on condensed examples as the evaluation metric. All the networks are trained from scratch for multiple times — 10 times for low-resolution datasets and



(a) Initialized SVHN images. (b) Condensed images by DM. (c) Condensed images by M3D.

Figure 4: Visualization of the condensed set of SVHN dataset with 10 images per class. The condensed set is generated by (b) DM and (c) M3D. Both DM and M3D use the same initialization as (a) shows.

IPC	ImageNet-10				ImageNet-100			
	10		20		10		20	
	Acc.	Time	Acc.	Time	Acc.	Time	Acc.	Time
Random	46.9	-	51.8	-	20.7	-	29.7	-
Herding	50.4	-	57.5	-	22.6	-	31.1	-
DSA	52.7	27.0h	57.4	51.4h	21.8	9.7h	30.7	23.9h
IDC	72.8	70.1h	76.6	92.8h	46.7	141.0h	53.7	185.0h
DM	52.3	1.4h	59.3	3.6h	22.3	**2.8h**	30.4	**2.8h**
M3D	**73.4**	**1.1h**	**76.8**	**3.1h**	**46.9**	3.5h	**55.5**	4.2h

Table 3: The performance and efficiency comparison on high-resolution ImageNet-subsets. The synthetic examples are condensed using ResNetAP-10. The minimal time required for obtaining the best performance is reported, which is measured on a single RTX-A6000 GPU with same batch size. For ImageNet-100, all methods are splitted into five sub-tasks with 20 classes each for faster optimization.

3 times for ImageNet subsets. We report the average performance and the standard deviation.

Implementation Details. We employ the Gaussian kernel for RKHS by default. The number of iterations is set to 10K for all low-resolution datasets. While for ImageNet subsets, we set 1K iterations. Additionally, the number of iterations per model is consistently set to 5 across all datasets. Regarding the learning rates for the condensed data, we assign a value of 1 for low-resolution datasets including F-MNIST, SVHN and CIFAR-10/100. For ImageNet subsets, we adopt a learning rate of 1e-1. Following IDC (Kim et al. 2022), the factor parameter l is set to 2 for low-resolution datasets and 3 for ImageNet subsets.

Comparison to the SOTA Methods

Table 2 and Table 3 present the comparison of our method with coreset selection and dataset condensation methods. The results show that synthetic examples are more informative than the selected ones, especially when the number of image(s) per class is small. This is attributed to the fact that synthetic examples are not confined to the set of real examples. Furthermore, our method consistently outperforms other baselines across a diverse set of scenarios. Remarkably, M3D achieves over a 5% higher accuracy than the best baseline on SVHN, CIFAR-10 (IPC=10), and CIFAR-100 (IPC=1). Notably, for high-resolution ImageNet subsets (Deng et al. 2009; Kim et al. 2022; Zhang et al. 2023), our method surpasses all baselines in test accuracy, including the current SOTA optimization-oriented IDC (Kim et al. 2022). It is worth noting that IDC (Kim et al. 2022) demands an exceptionally long time to condense ImageNet subsets, e.g., approximately 4 days on ImageNet-10 with IPC=20 (Zhang et al. 2023). In contrast, M3D achieves superior performance in a matter of hours. Additionally, our method eliminates the need for network updates, thereby circumventing the tuning of various hyper-parameters. Consequently, our method can be readily applied to realistic and larger datasets, maintaining efficiency and effectiveness simultaneously.

To further demonstrate the advantages of our method, we provide the test accuracy across varying training steps in Fig. 3. As observed, our method consistently outperforms DM at different training steps. Even without the factor technique, our method still achieves considerable improvement, highlighting the effectiveness of M3D in aligning distributions compared to previous DM-based methods.

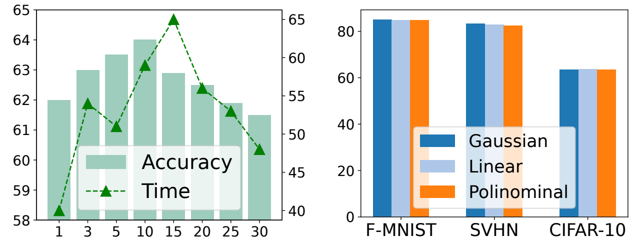
Cross-Architecture Evaluation. We further assess the performance of our condensed examples on different architectures. In Table 4, we present the performance of our condensed examples from CIFAR-10 dataset on ConNet-3, ResNet-10 (He et al. 2016), and DenseNet-121 (Huang et al. 2017). Combining the results from Table 2, we can find that M3D outperforms the compared methods not only on the architecture used for condensation but on unseen ones.

Figure 5: Representative samples condensed by M3D on ImageNet. The corresponding labels, from left to right and top to bottom, are bonnet, green snake, langur, doberman, gyromitra, saluki, vacuum, window screen, and cockroach.

IPC	Evaluation Model	DSA	DM	M3D
10	ConvNet-3	52.1	48.9	**63.5**
	ResNet-10	32.9	42.3	**56.7**
	DenseNet-121	34.5	39.0	**54.6**
50	ConvNet-3	60.6	63.0	**69.9**
	ResNet-10	49.7	58.6	**66.6**
	DenseNet-121	49.1	57.4	**66.1**

Table 4: Cross-architecture generalization performance (%) on CIFAR-10. The synthetic examples is condensed using ConvNet-3 and evaluated using other architectures.

Visualizations. We visualize the condensed images of SVHN and ImageNet in Fig. 4 and Fig. 5, respectively. For SVHN, we initialize the synthetic set $\mathcal{S}$ using random images from the training set $\mathcal{T}$ and then apply the condensation process using DM and M3D. As shown, the condensed images by DM and M3D appear as if the original images have been augmented with a distinct texture. Notably, the condensed images produced by our method exhibit a more pronounced and visually appealing texture compared to DM. While the overall appearance remains similar, our condensed images demonstrate better alignment with the higher-order moments of the original training set. In the case of ImageNet, the condensed images exhibit a texture reminiscent of a sunspot. In contrast to optimization-oriented methods, the images condensed by M3D retain more natural features and are more visually recognizable to humans. More visualization results are provided in the Appendix.



(a) (b)

Figure 6: (a) Ablation of IPM, where the horizontal axis represents the number of IPM, the left and right vertical axis denote test accuracy (%) and the corresponding time cost (mins), respectively. (b) Ablation of the kernel function, where the vertical axis denotes test accuracy (%).

Ablation Study

Impact of the Iteration per Model (IPM). We conduct experiments using various number of iterations per model, and the corresponding performance is depicted in Fig. 6a. We adopt CIFAR-10 with 10 images per class to showcase the impact of IPM. In addition to the test accuracy of condensed examples, we also provide the training time required to achieve the reported accuracy. As shown, increasing the number of IPM may lead to improved performance of the condensed data, but it also increases the training time. Conversely, an excessively large IPM can compromise the generalization ability of the condensed examples.

Impact of the Kernel Function. Different kernel functions construct distinct Reproducing Kernel Hilbert Spaces (RKHS). To investigate their influence, we adopt two additional kernel functions in addition to the Gaussian kernel: the linear kernel and the polynomial kernel. Fig. 6b illustrates the test accuracy under different kernel functions with 10 images per class. As observed, the choice of $\mathcal{K}$ has minimal impact on the performance of the condensed dataset. This indicates that as long as the selected kernel function is valid, our M3D can effectively embed the distributions in the constructed RKHS, resulting in a robust method.

Conclusion

In conclusion, this paper introduces a novel Distribution-Matching (DM)-based method called M3D for dataset condensation. With a theoretical guarantee, our method embeds the representation distributions of real and synthetic examples in a reproducing kernel Hilbert space, minimizing the maximum mean discrepancy between them to align their distributions in both first- and higher-order moments. Extensive experiments show the effectiveness and efficiency of our method. Notably, the efficiency of our method enables its application to more realistic and larger datasets. M3D first studies the alignment of higher-order moments of the representation distributions between real and synthetic examples, and establishes a strong baseline in DM-based methods for dataset condensation, which we believe will be valuable to the research community.

Acknowledgments

This work was partially supported by grants from the National Key Research and Development Plan (2020AAA0140001), and the Beijing Natural Science Foundation (19L2040), and the Open Research Project of National Key Laboratory of Science and Technology on Space-Born Intelligent Information Processing (TJ-02-22-01).

References

Altun, Y.; and Smola, A. 2006. Unifying divergence minimization and statistical inference via convex duality. In *ICCLT*, 139–153.

Borgwardt, K. M.; Gretton, A.; Rasch, M. J.; Kriegel, H.-P.; Schölkopf, B.; and Smola, A. J. 2006. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14): e49–e57.

Cazenavette, G.; Wang, T.; Torralba, A.; Efros, A. A.; and Zhu, J.-Y. 2022. Dataset Distillation by Matching Training Trajectories. In *CVPR*, 4750–4759.

Cazenavette, G.; Wang, T.; Torralba, A.; Efros, A. A.; and Zhu, J.-Y. 2023. Generalizing Dataset Distillation via Deep Generative Prior. In *CVPR*, 3739–3748.

Chen, Z.; Geng, J.; Zhu, D.; Woisetschlaeger, H.; Li, Q.; Schimmler, S.; Mayer, R.; and Chunming, R. 2023. A Comprehensive Study on Dataset Distillation: Performance, Privacy, Robustness and Fairness. *arXiv*.

Coleman, C.; Yeh, C.; Mussmann, S.; Mirzasoleiman, B.; Bailis, P.; Liang, P.; Leskovec, J.; and Zaharia, M. 2019. Selection via proxy: Efficient data selection for deep learning. *arXiv*.

Cui, J.; Wang, R.; Si, S.; and Hsieh, C.-J. 2022. DC-BENCH: Dataset condensation benchmark. *NeurIPS*, 35: 810–822.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255.

Deng, Z.; and Russakovsky, O. 2022. Remember the Past: Distilling Datasets into Addressable Memories for Neural Networks. In *NeurIPS*.

Du, J.; Jiang, Y.; Tan, V. T. F.; Zhou, J. T.; and Li, H. 2023. Minimizing the Accumulated Trajectory Error to Improve Dataset Distillation. In *CVPR*.

Farahani, R. Z.; and Hekmatfar, M. 2009. *Facility location: concepts, models, algorithms and case studies*. Springer Science & Business Media.

Fukumizu, K.; Bach, F. R.; and Jordan, M. I. 2004. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *JMLR*, 5(Jan): 73–99.

Goetz, J.; and Tewari, A. 2020. Federated Learning via Synthetic Data. *arXiv*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *CVPR*, 4700–4708.

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *NeurIPS*, 31.

Jiang, Z.; Gu, J.; Liu, M.; and Pan, D. Z. 2022. Delving into Effective Gradient Matching for Dataset Condensation. *arXiv*.

Jin, W.; Tang, X.; Jiang, H.; Li, Z.; Zhang, D.; Tang, J.; and Yin, B. 2022. Condensing graphs via one-step gradient matching. In *KDD*, 720–730.

Kim, J.-H.; Kim, J.; Oh, S. J.; Yun, S.; Song, H.; Jeong, J.; Ha, J.-W.; and Song, H. O. 2022. Dataset Condensation via Efficient Synthetic-Data Parameterization. In *ICML*, 11102–11118.

Knoblauch, J.; Husain, H.; and Diethe, T. 2020. Optimal continual learning has perfect memory and is np-hard. In *ICML*, 5327–5337.

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

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11): 2278–2324.

Lee, S.; Chun, S.; Jung, S.; Yun, S.; and Yoon, S. 2022. Dataset condensation with contrastive signals. In *ICML*, 12352–12364.

Lei, S.; and Tao, D. 2023. A comprehensive survey to dataset distillation. *arXiv*.

Li, S.; Liu, T.; Tan, J.; Zeng, D.; and Ge, S. 2023a. Trustable Co-label Learning from Multiple Noisy Annotators. *IEEE TMM*, 25: 1045–1057.

Li, S.; Xia, X.; Deng, J.; Ge, S.; and Liu, T. 2023b. Transferring Annotator-and Instance-dependent Transition Matrix for Learning from Crowds. *arXiv*.

Li, S.; Xia, X.; Ge, S.; and Liu, T. 2022a. Selective-Supervised Contrastive Learning with Noisy Labels. In *CVPR*, 316–325.

Li, S.; Xia, X.; Zhang, H.; Zhan, Y.; Ge, S.; and Liu, T. 2022b. Estimating Noise Transition Matrix with Label Correlations for Noisy Multi-Label Learning. In *NeurIPS*.

Liu, R.; Gao, J.; Zhang, J.; Meng, D.; and Lin, Z. 2021. Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *IEEE TPAMI*, 44(12): 10045–10067.

Liu, S.; Wang, K.; Yang, X.; Ye, J.; and Wang, X. 2022. Dataset distillation via factorization. *NeurIPS*, 35: 1100–1113.

Liu, S.; and Wang, X. 2023. Few-Shot Dataset Distillation via Translative Pre-Training. In *ICCV*, 18654–18664.

Liu, S.; Ye, J.; Yu, R.; and Wang, X. 2023a. Slimmable dataset condensation. In *CVPR*, 3759–3768.

Liu, Y.; Gu, J.; Wang, K.; Zhu, Z.; Jiang, W.; and You, Y. 2023b. DREAM: Efficient Dataset Distillation by Representative Matching. In *ICCV*.

Loo, N.; Hasani, R.; Amini, A.; and Rus, D. 2022. Efficient dataset distillation using random feature approximation. *NeurIPS*, 35: 13877–13891.

Loo, N.; Hasani, R.; Lechner, M.; and Rus, D. 2023. Dataset Distillation with Convexified Implicit Gradients. In *ICML*, 22649–22674.

Masarczyk, W.; and Tautkute, I. 2020. Reducing Catastrophic Forgetting with Learning on Synthetic Data. In *CVPR Workshop*.

Medvedev, D.; and D'yakonov, A. 2021. Learning to Generate Synthetic Training Data using Gradient Matching and Implicit Differentiation. In *AIST*, 138–150.

Muandet, K.; Fukumizu, K.; Sriperumbudur, B.; Schölkopf, B.; et al. 2017. Kernel mean embedding of distributions: A review and beyond. *Found. Trends Mach. Learn.*, 10(1-2): 1–141.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

Nguyen, T.; Chen, Z.; and Lee, J. 2020. Dataset meta-learning from kernel ridge-regression. *arXiv*.

Nguyen, T.; Novak, R.; Xiao, L.; and Lee, J. 2021. Dataset Distillation with Infinitely Wide Convolutional Networks. In *NeurIPS*, 5186–5198.

Pooladzandi, O.; Davini, D.; and Mirzasoleiman, B. 2022. Adaptive second order coresets for data-efficient machine learning. In *ICML*, 17848–17869.

Sagun, L.; Evci, U.; Guney, V. U.; Dauphin, Y.; and Bottou, L. 2017. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv*.

Sajedi, A.; Khaki, S.; Amjadian, E.; Liu, L. Z.; Lawryshyn, Y. A.; and Plataniotis, K. N. 2023. DataDAM: Efficient Dataset Distillation with Attention Matching. In *ICCV*, 17097–17107.

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

Smola, A.; Gretton, A.; Song, L.; and Schölkopf, B. 2007. A Hilbert space embedding for distributions. In *ALT*, 13–31.

Sorscher, B.; Geirhos, R.; Shekhar, S.; Ganguli, S.; and Morcos, A. 2022. Beyond neural scaling laws: beating power law scaling via data pruning. *NeurIPS*, 35: 19523–19536.

Such, F. P.; Rawal, A.; Lehman, J.; Stanley, K.; and Clune, J. 2020. Generative Teaching Networks: Accelerating Neural Architecture Search by Learning to Generate Synthetic Training Data. In *ICML*, 9206–9216.

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

Wang, C.; Sun, J.; Dong, Z.; Li, R.; and Zhang, R. 2023. Gradient Matching for Categorical Data Distillation in CTR Prediction. In *CRS*, RecSys '23, 161–170. New York, NY, USA: Association for Computing Machinery. ISBN 9798400702419.

Wang, K.; Zhao, B.; Peng, X.; Zhu, Z.; Yang, S.; Wang, S.; Huang, G.; Bilen, H.; Wang, X.; and You, Y. 2022. CAFE: Learning to Condense Dataset by Aligning Features. In *CVPR*, 12196–12205.

Wang, T.; Zhu, J.-Y.; Torralba, A.; and Efros, A. A. 2018. Dataset Distillation. *arXiv*.

Welling, M. 2009. Herding dynamical weights to learn. In *ICML*, 1121–1128.

Wiewel, F.; and Yang, B. 2021. Condensed Composite Memory Continual Learning. In *IJCNN*, 1–8.

Xia, X.; Liu, J.; Yu, J.; Shen, X.; Han, B.; and Liu, T. 2022. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *ICLR*.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv*.

Xiong, Y.; Wang, R.; Cheng, M.; Yu, F.; and Hsieh, C.-J. 2023. Feddm: Iterative distribution matching for communication-efficient federated learning. In *CVPR*, 16323–16332.

Yang, S.; Xie, Z.; Peng, H.; Xu, M.; Sun, M.; and Li, P. 2022. Dataset pruning: Reducing training data by examining generalization influence. *arXiv*.

Ying, C.; Klein, A.; Christiansen, E.; Real, E.; Murphy, K.; and Hutter, F. 2019. Nas-bench-101: Towards reproducible neural architecture search. In *ICML*, 7105–7114.

Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; and Darrell, T. 2020. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2636–2645.

Yu, R.; Liu, S.; and Wang, X. 2023. Dataset distillation: A comprehensive review. *arXiv*.

Zhang, H.; Li, S.; Zeng, D.; Yan, C.; and Ge, S. 2024. Coupled Confusion Correction: Learning from Crowds with Sparse Annotations. *AAAI*.

Zhang, L.; Zhang, J.; Lei, B.; Mukherjee, S.; Pan, X.; Zhao, B.; Ding, C.; Li, Y.; and Dongkuan, X. 2023. Accelerating Dataset Distillation via Model Augmentation. In *CVPR*.

Zhao, B.; and Bilen, H. 2021a. Dataset condensation with Differentiable Siamese Augmentation. In *ICML*, 12674–12685.

Zhao, B.; and Bilen, H. 2021b. Dataset Condensation with Gradient Matching. In *ICLR*.

Zhao, B.; and Bilen, H. 2023. Dataset Condensation with Distribution Matching. In *WACV*.

Zhao, G.; Li, G.; Qin, Y.; and Yu, Y. 2023a. Improved distribution matching for dataset condensation. In *CVPR*, 7856–7865.

Zhao, G.; Li, G.; Qin, Y.; and Yu, Y. 2023b. Improved distribution matching for dataset condensation. In *CVPR*, 7856–7865.

Zheng, H.; Liu, R.; Lai, F.; and Prakash, A. 2022. Coverage-centric Coreset Selection for High Pruning Rates. *arXiv*.

Zhou, D.; Wang, K.; Gu, J.; Peng, X.; Lian, D.; Zhang, Y.; You, Y.; and Feng, J. 2023. Dataset quantization. In *ICCV*, 17205–17216.

Zhou, Y.; Nezhadarya, E.; and Ba, J. 2022. Dataset Distillation using Neural Feature Regression. In *NeurIPS*.