Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)
Survey Track

A Survey on Dataset Distillation: Approaches, Applications and Future Directions

Jiahui Geng[1], **Zongxiong Chen**[*2], **Yuandou Wang**[3], **Herbert Woisetschläger**[4], **Sonja Schimmler**[2], **Ruben Mayer**[4], **Zhiming Zhao**[3] and **Chunming Rong**[1]
[1]University of Stavanger
[2] Fraunhofer FOKUS
[3] University of Amsterdam
[4] Technical University of Munich
{jiahui.geng, chunming.rong}@uis.no, {herbert.woisetschlaeger, ruben.mayer}@tum.de,
{z.zhao, y.wang8}@uva.nl, {zongxiong.chen, sonja.schimmler}@fokus.fraunhofer.de

Abstract

Dataset distillation is attracting more attention in machine learning as training sets continue to grow and the cost of training state-of-the-art models becomes increasingly high. By synthesizing datasets with high information density, dataset distillation offers a range of potential applications, including support for continual learning, neural architecture search, and privacy protection. Despite recent advances, we lack a holistic understanding of the approaches and applications. Our survey aims to bridge this gap by first proposing a taxonomy of dataset distillation, characterizing existing approaches, and then systematically reviewing the data modalities, and related applications. In addition, we summarize the challenges and discuss future directions for this field of research.

1 Introduction

High-quality and large-scale datasets are crucial for the success of deep learning, not only enabling the development of end-to-end learning systems [Schmidhuber, 2015; Bahdanau *et al.*, 2015], but also serving as benchmarks to evaluate different machine learning architectures [Deng *et al.*, 2009; Koehn, 2005]. However, the explosion of deep learning dataset sizes has posed considerable challenges concerning processing, storage, and transfer. Training neural networks often require thousands of iterations on the entire dataset, which consumes significant computational resources and power. Tasks such as hyperparameter optimization [Maclaurin *et al.*, 2015] and neural architecture search (NAS) [Such *et al.*, 2020] are even more resource-intensive. One promising solution is to use smaller datasets with high information density to reduce resource consumption while preserving model performance.

Research in the area of curriculum learning [Graves *et al.*, 2017], active learning [Konyushkova *et al.*, 2017], and coreset selection [Sener and Savarese, 2017] has shown that it is possible to sample a subset of the original data to train neural networks, resulting in models with competitive performance. This also implies that we can train high-performance models with less effort while downstream tasks like continual learning (CL) [Castro *et al.*, 2018; Prabhu *et al.*, 2020], neural architecture search (NAS) will also benefit. Nevertheless, creating an algorithm-agnostic, efficient, and unbiased small dataset to replace the original is still challenging. For instance, coreset selection is typically an NP-hard problem, making it computationally intractable and difficult to apply to large datasets.

An alternative approach to coreset is dataset distillation, which aims to distill the original data onto a smaller synthetic dataset [Wang *et al.*, 2018]. Dataset distillation techniques have continued to evolve, with various methods such as gradient matching [Zhao *et al.*, 2021], trajectory matching [Cazenavette *et al.*, 2022], and kernel ridge regression [Nguyen *et al.*, 2020] being proposed to optimize the distilled data, resulting in improved distillation performance in terms of both the accuracy of the trained model on the test set and the generalization capability across different network architectures. However, there remain challenges regarding optimization stability and computational efficiency.

Despite the recent advancements in dataset distillation, a comprehensive overview summarizing its advances and applications is currently not available. This paper aims to fill this gap by presenting a taxonomy of dataset distillation. To our knowledge, it is the first work that provides a systematic categorization of the different methods and techniques used in dataset distillation. The paper mainly makes the following contributions:

- We propose a novel taxonomy of dataset distillation, which can help researchers to better understand the research landscape and find their areas of interest.
- We present existing distillation approaches in detail, discussing their strengths and weaknesses;
- We discuss important challenges in this domain, highlighting promising directions for future research.

The paper is organized as follows. In Section 2, we first present our taxonomy of dataset distillation. Then, we introduce the learning frameworks and common enhancement methods in Section 3 and Section 4, respectively. Section 5

*Corresponding author: zongxiong.chen@fokus.fraunhofer.de

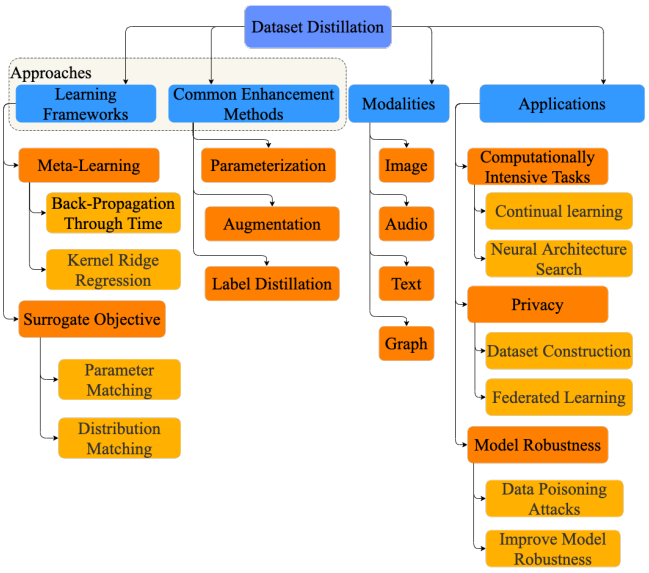

Figure 1: Taxonomy of dataset distillation.

summarizes the advances in different data modalities. In Section 6, we categorize the related applications according to the dataset distillation properties. Finally, we conclude this paper with future directions in Section 7.

2 Taxonomy

2.1 Basics of Dataset Distillation

We begin by introducing the key notations used in this paper. $\mathcal{D}$ represents a general dataset, f_θ represents a neural network with parameters θ, and $f_\theta(x)$ denotes the model's prediction for data point x. The expected loss for dataset $\mathcal{D}$ in relation to θ is defined as

$$\mathcal{L}_\mathcal{D}(\theta) = \mathbb{E}_{(x,y)\sim P_\mathcal{D}}[\ell(f_\theta(x), y)], \tag{1}$$

where x and y are the input data and label pair from $\mathcal{D}$, $\ell(f_\theta(x), y)$ is the given loss value between the prediction and ground truth.

Dataset distillation aims to reduce the size of large-scale training input and label pairs $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{T}|}$ by creating smaller synthetic pairs $\mathcal{S} = \{(\hat{x}_j, \hat{y}_j)\}_{j=1}^{|\mathcal{S}|}$, so that models trained on both $\mathcal{T}$ and $\mathcal{S}$ can achieve similar performance, which can be formulated as:

$$\mathcal{L}_\mathcal{T}(\theta^\mathcal{S}) \simeq \mathcal{L}_\mathcal{T}(\theta^\mathcal{T}), \tag{2}$$

where $\theta^\mathcal{S}$ and $\theta^\mathcal{T}$ are the parameters of the models trained on $\mathcal{S}$ and $\mathcal{T}$ respectively.

2.2 Taxonomy Explanation

The taxonomy of dataset distillation is illustrated in Figure 1. In this taxonomy, we classify the research about dataset distillation from three perspectives: approaches, data modalities and applications. The approaches can be decomposed into two parts. In the learning framework, we explain how dataset distillation can be modeled, optimized and solved in different ways, such as using meta-learning [Andrychowicz *et al.*, 2016] or surrogate objectives (see Section 3.2). Meta-learning can be further divided into using back-propagation through time and using kernel ridge regression. Surrogate objective can be subdivided into parameter matching and distribution matching. We categorize the common enhancement methods, which can be plugged into a learning framework, mainly into parameterization (see Section 4.1), augmentation (see Section 4.2) and label distillation (see Section 4.3). Existing work can be classified into four types of data: image, audio, text, and graph, based on data modality. Applications can be further divided into three categories: computationally intensive tasks such as continual learning and neural architecture search, privacy protection including dataset construction and federated learning, and model robustness, encompassing data poisoning attacks and improving robustness. Corresponding to our taxonomy, some representative papers, together with their characteristics, have been listed in Table 1. It comprehensively compares learning frameworks, enhancement methods, data modality, and applications.

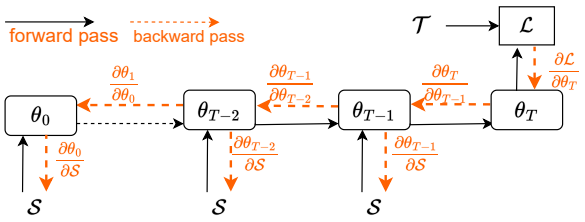

Figure 2: Back-Propagation Through Time. The gradient $\nabla_\mathcal{S}\mathcal{L}$ is calculated via back-propagation through time (see orange dashed line).

3 Learning Frameworks

According to the learning goals, the current learning frameworks can mainly be divided into two categories: meta-learning methods based on inner model performance and methods using surrogate objectives.

3.1 Meta-Learning

Meta-learning [Andrychowicz *et al.*, 2016] refers to learning about learning, and often refers to machine learning algorithms that learn from the output of other machine learning algorithms. In this problem, the distilled data are treated as hyperparameters and the objective is to optimize the distilled data in a bi-level optimization problem as follows:

$$\mathcal{S}^* = \arg\min_\mathcal{S} \mathcal{L}_\mathcal{T}(\theta^\mathcal{S}) \text{ s.t. } \theta^\mathcal{S} = \arg\min_\theta \mathcal{L}_\mathcal{S}(\theta), \tag{3}$$

where the inner loop, optimizing $\theta^\mathcal{S}$, trains a model on the synthetic dataset until convergence, and the outer loop, optimizing $\mathcal{S}$, subsequently optimizes the synthetic dataset, so that the model has good generalization capability and can perform well on the real dataset. The distillated dataset is optimized using the meta-gradient:

$$\mathcal{S} \leftarrow \mathcal{S} - \alpha\nabla_\mathcal{S}\mathcal{L}_\mathcal{T}(\theta^\mathcal{S}), \tag{4}$$

where α is the learning rate for updating the synthetic dataset.

Paper	Learning Framework	Enhancement Methods	Data Modality				Applications		
			Image	Text	Audio	Graph	Computationally Intensive Tasks	Privacy	Robustness
DD [Wang *et al.*, 2018]	Back-Propagation Through Time		✓						Data Poisoning
SLDD, TDD [Sucholutsky and Schonlau, 2021]		LD	✓	✓					
Addressable Memory [Deng, 2022]		Factorization, LD	✓				CL		
KIP [Nguyen *et al.*, 2020; Nguyen *et al.*, 2021]	Kernel Ridge Regression	LD	✓					ρ-corruption	
FRePo [Zhou *et al.*, 2022]		LD	✓				CL	MIA	
RFAD [Loo *et al.*, 2022]		LD	✓					ρ-corruption	
DC [Zhao *et al.*, 2021]	Parameter Matching		✓				CL, NAS		
DSA [Zhao Bo, 2021]		DSA	✓				CL, NAS		
MTT [Cazenavette *et al.*, 2022]		DSA	✓						
IDC [Kim *et al.*, 2022]		Factorization, DSA	✓		✓		CL		
HaBa [Liu *et al.*, 2022b]		Factorization	✓		✓		CL		
PSG [Chen *et al.*, 2022]			✓					MIA, DP	
GCond [Jin *et al.*, 2021]						✓	NAS		
DosCond [Jin *et al.*, 2022]						✓			
DM [Zhao and Bilen, 2021]	Distribution Matching	DSA	✓				CL, NAS		
CAFE [Wang *et al.*, 2022]			✓						
IT-GAN [Zhao and Bilen, 2022]		DSA, GAN	✓						
GCDM [Liu *et al.*, 2022a]						✓			
KFS [Lee *et al.*, 2022]			✓						

Table 1: Summary of existing dataset distillation works. CL – Continual Learning, NAS – Neural Architecture Search, MIA – Membership Inference Attack, DP – Differential Privacy, and LD – Label Distillation. Note: ✓– if it uses such data modality.

Back-Propagation Through Time

Computing the meta-gradient $\nabla_{\mathcal{S}}\mathcal{L}_{\mathcal{T}}(\theta^{\mathcal{S}})$ requires differentiating through inner optimization. When the model is learned in an iterative way, i.e.,

$$\theta_{t+1} = \theta_t - \eta\nabla_{\theta_t}\ell(f_\theta(\hat{x}), \hat{y}), \tag{5}$$

where η is the learning rate for inner loop and meta-gradient is calculated by back-propagation through time (BPTT):

$$\nabla_{\mathcal{S}}\mathcal{L}_{\mathcal{T}}(\theta^{\mathcal{S}}) = \frac{\partial\mathcal{L}}{\partial\mathcal{S}} = \frac{\partial\mathcal{L}}{\partial\theta_T}\Big(\sum_{t=0}^{t=T}\frac{\partial\theta_T}{\partial\theta_t}\cdot\frac{\partial\theta_t}{\partial\mathcal{S}}\Big) \tag{6}$$

which is illustrated in Figure 2. It is evident that the computation overhead is high due to the recursive calculation of the meta-gradient using Equation 6.

To make the implementation of Equation 6 feasible, [Wang *et al.*, 2018] suggest using the Truncated Back-Propagation Through Time (TBPTT) method, which involves unrolling the inner-loop optimization steps as a single step of gradient descent optimization,

$$\hat{x}, \hat{\eta} = \arg\min_{\hat{x},\hat{\eta}} \ell(f_{\theta_1}(x), y), \text{s.t.}\theta_1 = \theta_0 - \eta\nabla_{\theta_0}\ell(f_{\theta_0}(\hat{x}), \hat{y}), \tag{7}$$

where $\hat{x}$, $\hat{y}$ are synthetic dataset and $\hat{\eta}$ the learning rate for the optimizer.

[Deng, 2022] further improves the learning framework by incorporating a momentum term and extending the length of unrolled trajectories. Empirical results show that the momentum term can consistently improve performance and that

longer unrolled trajectories can lead to better model parameters that produce more efficient gradients for compressed representation learning.

BPTT methods have been criticized for several issues, as noted in [Zhou *et al.*, 2022]: 1) high computational cost and memory overhead; 2) bias in short unrolls; 3) gradients exploding or vanishing in long unrolls; and 4) chaotic conditioned loss landscapes.

Kernel Ridge Regression

[Nguyen *et al.*, 2020] transform dataset distillation into a kernel ridge regression (KRR) problem, where the synthetic set is used as the support set and the original set as the target set. Their approach result in a closed-form solution in terms of convex optimization, simplifying the expensive nested optimization into first-order optimization (see Figure 3). They introduce the Kernel-Inducing Point (KIP) algorithm which utilizes neural tangent kernel (NTK) [Jacot *et al.*, 2018] ridge regression to compute the exact outputs of an infinite-width neural network trained on the synthetic set, bypassing the need for gradient and back-propagation computation on any neural network. The KRR loss function for a given kernel and batch data from synthetic set $(X_\mathcal{S}, y_\mathcal{S})$ evaluated on batch data from real set $(X_\mathcal{T}, y_\mathcal{T})$ can be formulated as,

$$\arg\min_{X_\mathcal{S}, y_\mathcal{S}} \frac{1}{2}\|y_\mathcal{T} - K_{X_\mathcal{T}X_\mathcal{S}}(K_{X_\mathcal{S}X_\mathcal{S}} + \lambda I)^{-1} y_\mathcal{S}\|^2, \tag{8}$$

where $K_{X_\mathcal{T}X_\mathcal{S}}$ is the Gram matrix of $X_\mathcal{S}$ and $X_\mathcal{T}$, and $K_{X_\mathcal{S}X_\mathcal{S}}$ is the Gram matrix of $X_\mathcal{S}$.

[Zhou *et al.*, 2022] propose a novel method, neural feature regression with pooling (FRePo), which utilizes a more flexible conjugate kernel with neural features to replace the NTK in KIP [Nguyen *et al.*, 2020]. This approach breaks down the traditional KRR training pipeline into two components: a feature broke f_θ and a linear classifier. When calculating the meta-gradient of $\mathcal{S}$, FRePo fixes the feature extractor parameters and updates $\mathcal{S}$ T times according to Equation 8, where T is a hyperparameter that helps prevent the support/synthetic dataset from memorizing a specific network. Additionally, a model pool is employed to alleviate overfitting in the distillation process.

$$K^\theta_{X_\mathcal{T}X_\mathcal{S}} = f_\theta(X_\mathcal{T}) f_\theta(X_\mathcal{S})^\top, \tag{9}$$

$$K^\theta_{X_\mathcal{S}X_\mathcal{S}} = f_\theta(X_\mathcal{S}) f_\theta(X_\mathcal{S})^\top \tag{10}$$

[Loo *et al.*, 2022] propose to use random feature approximation for distillation (RFAD), which utilizes random feature approximation of the Neural Network Gaussian Process (NNGP) kernel to replace the NTK used in KIP. This approach reduces the computation of the Gram matrix to $\mathcal{O}(|\mathcal{S}|)$, which is linear with the size of the synthetic set, compared to $\mathcal{O}(|\mathcal{S}|^2)$, the complexity of accurately calculating the NTK kernel matrix. They also suggest using cross-entropy loss with Platt scaling [Platt and others, 1999] to provide a more accurate probabilistic interpretation for classification tasks.

3.2 Surrogate Objective

Instead of optimizing directly based on model performance, surrogate objective approaches optimize a proxy objective, such as the parameters or gradients of the model. These approaches assert that the effectiveness of a model trained on a full dataset and a distilled dataset can be inferred from their corresponding parameters and gradients.

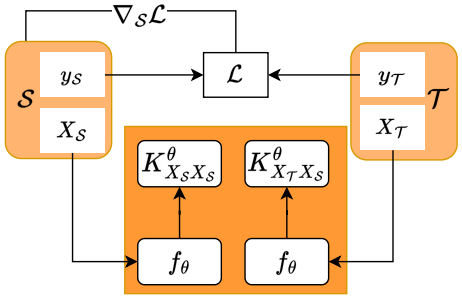

Figure 3: Kernel Ridge Regression. The figure shows the workflow of kernel ridge regression. The details refer to Equation 8 and 9. The key difference is that KIP [Nguyen *et al.*, 2020] uses NTK kernel, RFAD [Loo *et al.*, 2022] uses Neural Network Gaussian Process (NNGP) kernel. Feature extractor f_θ in FRePo [Zhou *et al.*, 2022] is parameterized during training.

Parameter Matching

In contrast to optimizing directly based on the loss value corresponding to the distilled data, it aims to make the model approximate the original model in the parameter space, i.e. $\theta^\mathcal{S} \approx \theta^\mathcal{T}$. Empirically, the trajectory of parameters vary with its initial state θ_0. Therefore, the objective of parameter matching should be agnostic to the initialization. When distance between the model parameters trained on the synthetic dataset and the real dataset are consistently close, the distilled dataset can be considered as a good alternative of original whole dataset. Let $\theta^\mathcal{S}(\theta_0)$, $\theta^\mathcal{T}(\theta_0)$ denote the trained models from the same initialization θ_0, the objective function can be expressed as:

$$\min_\mathcal{S} \mathbb{E}_{\theta_0 \sim P_{\theta_0}}[D(\theta^\mathcal{S}(\theta_0), \theta^\mathcal{T}(\theta_0))], \tag{11}$$

where $D(\cdot, \cdot)$ is a distance function.

To enable a more guided optimization and apply the incomplete training, DC [Zhao *et al.*, 2021] synthesizes images by minimizing the gradient matching loss at each training step t:

$$\min_\mathcal{S} \mathbb{E}_{\theta_0 \sim P_{\theta_0}}[\sum_{t=0}^{T-1} D(\nabla_\theta \mathcal{L}_\mathcal{S}(\theta_t), \nabla_\theta \mathcal{L}_\mathcal{T}(\theta_t))] \tag{12}$$

where T is the hyperparameter for the number of training iterations.

[Cazenavette *et al.*, 2022] suggest overcoming bias accumulated from one-step gradient by matching training trajectories (MTT). MTT considers the training trajectories $\theta^{T-1}_{t=0}$ on real data as the expert models, the model $\hat{\theta}$ trained on the synthetic dataset as the student model. It randomly samples $\theta^\mathcal{T}_t$ from the expert model to initialize the student model, and the objective is to make the student model $\hat{\theta}^\mathcal{S}_{t+N}$ approximate the expert model $\theta^\mathcal{T}_{t+M}$ after N iterations. The optimization objective is given by

$$D = \frac{\|\hat{\theta}^\mathcal{S}_{t+N} - \theta^\mathcal{T}_{t+M}\|_2^2}{\|\theta^\mathcal{T}_t - \theta^\mathcal{T}_{t+M}\|_2^2}, \tag{13}$$

where M, N are the hyperparameters.

Parameter matching methods are often criticized for: 1) high bias it introduces [Wang *et al.*, 2022]. The synthetic set learned by gradient matching is extremely biased towards

those large gradient samples, which will decrease its generalization capability on unseen architectures; 2) expensive bi-level optimization. For example training 50 images/class using DC [Zhao *et al.*, 2021] requires 500K epochs of updating network parameter θ_t and 50K updating of $\mathcal{S}$; and 3) fragile hyper-parameters [Zhao and Bilen, 2021] tuning. e.g. how often to update θ_t and $\mathcal{S}$ in DC, as well as M, N in MTT [Cazenavette *et al.*, 2022] are critical.

Distribution Matching

The objective of distribution matching is essentially to learn synthetic samples so that the distribution of the synthetic samples is similar to that of real samples in the feature space. They use an empirical estimate of the maximum mean discrepancy (MMD) as a metric to evaluate the distance of spatial distribution. Due to the high computational complexity and difficulty in optimization caused by high dimensionality, Zhao [Zhao and Bilen, 2021] use different randomly initialized neural networks as feature extractors to reduce the input dimension to low-dimensional space.

$$\min_{\mathcal{S}} \mathbb{E}_{\theta\sim P_\theta}\left\| \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|} f_\theta(\hat{x}_i) - \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|} f_\theta(x_i)\right\|^2, \qquad (14)$$

where f_θ is parameterized by θ, and θ is sampled from a random distribution P_θ. $|\mathcal{S}|$ and $|\mathcal{T}|$ are the cardinality of dataset $\mathcal{S}$ and $\mathcal{T}$, respectively.

To better capture the whole dataset distribution, [Wang *et al.*, 2022] propose to use layer-wise feature alignment in CAFE to learn a more comprehensive characteristic of the distribution. They also introduce a loss function to improve the discriminative ability of the learned samples. The classification loss is calculated using the feature centers of real sample and averaged synthetic samples of each class.

4 Common Enhancement Methods

In this section we introduce some techniques that can be integrated into the learning framework presented in the previous section to further enhance distillation performance.

4.1 Parameterization

Dataset parameterization aims to utilize the regularity to guide the synthesis, It helps enhance the interpretability by learning hidden patterns, and control the diversity of the synthetic data. In [Zhao and Bilen, 2022], the authors propose IT-GAN, a method that uses a pre-trained GAN decoder to increase the informativeness distilled data. IT-GAN first obtains latent vectors from training samples using GAN Inversion [Abdal *et al.*, 2019], then it use the distribution matching algorithm to learn the latent vectors. These vectors can be fed into a pre-trained GAN decoder to induce synthetic images of the original size. In addition, most distillation methods processes each synthetic sample independently, ignoring mutual consistency and relationships between samples. Factorization are proposed to decompose images into different parts to better capture the correlation between different samples and improve the diversity. IDC [Kim *et al.*, 2022] utilizes a multi-formation function as the decoder as the decoder to store more information in single sample. [Deng, 2022] propose to learn matrix-based codes and decodes and use matrix multiplication to generate synthetic datasets. [Lee *et al.*, 2022] employ the latent code - decoder mode for factorization. The decoder is designed as an upsampling neural network containing three ConvTranspose2d layers, aiming to restore latent codes compressed in low dimensions into the image pixel space. [Liu *et al.*, 2022b] propose HaBa, which chooses to decompose the image into two parameter spaces of basis and hallucinator. Where hallucinator is an encoder-transformation-decoder structure. Specifically, the encoder is composed of CNN blocks, followed by an affine transformation with scale σ and a decoder of a symmetric CNN architecture.

4.2 Augmentation

In [Zhao Bo, 2021], the authors propose using differentiable siamese augmentation (DSA) when learning synthetic images, which leads to more informative datasets. DSA is a pluggable technique that includes operators like *scale*, *flip*, *crop*, *rotate*, *color jitters*, and *cutout*. It can be easily integrated into various distillation methods and has been widely used in [Zhao and Bilen, 2021; Wang *et al.*, 2022]. In [Cui *et al.*, 2022], DSA is found to achieve the best performance compared to other data augmentation techniques. However, current augmentation techniques are not suitable for discrete data such as graphs and text.

4.3 Label Distillation

Label distillation relaxes the restrictions on labels, allowing them to have richer semantics beyond one-hot vectors. It is first introduced in SLDD [Sucholutsky and Schonlau, 2021] and has been shown to improve not only the storage efficiency but also the distillation performance. Their method only requires to make the label in Equation 7 learnable variables. [Nguyen *et al.*, 2020] also provide a label learning algorithm based on the closed-form solution in KRR. It is reported that only five distilled images from MNIST would enable the model to achieve 92% accuracy [Sucholutsky and Schonlau, 2021].

5 Data Modalities

Dataset distillation, first proposed for images, has been applied to various modalities. In this section, we categorize existing works according to data modality and discuss some of the challenges.

5.1 Image

Most dataset distillation methods to date have been performed on image datasets [Wang *et al.*, 2018; Nguyen *et al.*, 2020; Zhou *et al.*, 2022; Zhao *et al.*, 2021; Kim *et al.*, 2022; Cazenavette *et al.*, 2022]. These works have constructed benchmarks to facilitate fair comparisons of novel approaches. Images have a continuous real-value domain, which allows direct optimization of synthetic images using deep learning optimizers. We find that experimental datasets become increasingly complex, starting from MNIST, CIFAR10, and SVHN, to more challenging datasets like Tiny-ImageNet and ImageNet [Zhou *et al.*, 2022; Cazenavette *et al.*, 2022; Kim *et al.*, 2022]. Furthermore, parameterization methods that capture on the regularity of images are becoming increasingly prevalent in the field, as evidenced by recent

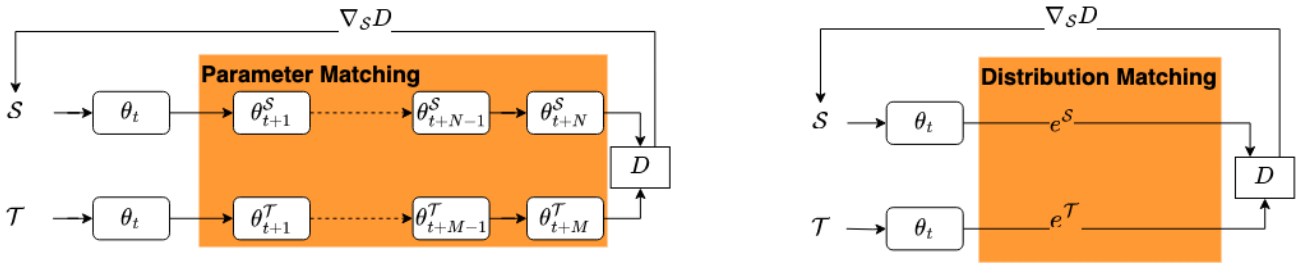

Figure 4: Surrogate Objective. The figure presents the workflow of parameter matching (left) and distribution matching (right). The key difference between algorithm DC [Zhao *et al.*, 2021] and MTT [Cazenavette *et al.*, 2022] is that DC uses information from one-step optimization (gradient) while MTT using parameters after several steps. Definition of D is given as in Equation 11. In distribution matching, the embeddings $e^{\mathcal{S}}$ and $e^{\mathcal{T}}$ in DM [Zhao and Bilen, 2021] are extracted from layer output of ConvNet and the D is maximum mean discrepancy, whereas, $e^{\mathcal{S}}$ and $e^{\mathcal{T}}$ in CAFE [Wang *et al.*, 2022] correspond to layer-wise features and D is a mean square error.

research such as [Kim *et al.*, 2022; Zhao and Bilen, 2022; Liu *et al.*, 2022b].

5.2 Audio

Speech signals also satisfy the regularity condition of a low-rank data subspace, i.e., temporally adjacent signals haves similar spectra. Therefore, many parametrization methods [Liu *et al.*, 2022b; Kim *et al.*, 2022] designed for image dataset can also be applied in this domain. They both experiment with the Speech Commands [Warden, 2018] dataset. In detail, they process the waveform data with a short-time Fourier transform to obtain the magnitude spectrogram and used log-scale magnitude spectrograms for the experiments. Their works show that dataset distillation can achieve consistent performance on downstream tasks of speech signals.

5.3 Text

The discrete nature poses challenges to textual distillation. [Sucholutsky and Schonlau, 2021] first embed the text into a contiguous space using pre-trained GloVe embedding and fill or truncate all sentences to a pre-determined length. In this way, each sentence can be regarded as a single channel image of size length $\times$ embedding dimension. Text distillation also involves finding the nearest embedding in the dictionary for each vector in the optimized matrix, and transforming these embeddings into the corresponding words and finally the sentence.

Current efforts are based on primitive bi-level optimization, which is computationally inefficient. There is a lack of work analyzing factors such as the difficulty of the dataset, sentence length, or cross-architecture generalization. Distilled sentences may consist of unrelated words, which makes it difficult to interpret and further analyze. Exploring ways to leverage regularity and context in text distillation is a promising area of research.

5.4 Graph

Graph data is very common in real life, e.g. social networks, Web relationship analysis, and user-item interaction can all be modeled as graph data containing nodes and edges. [Jin *et al.*, 2021; Jin *et al.*, 2022] design a strategy to simultaneously compress node features and structural information based on gradient matching. [Liu *et al.*, 2022a] adopt the distribution matching to boost the performance and show that the dataset distillation was significantly efficient and in some datasets they reached 95% of the original performance by compressing 99% of the data. Graph distillation is mainly challenged by heterogeneous, abstract, high-level graph representations.

6 Applications

Dataset distillation, initially designed for model training acceleration, has shown potential in various applications due to its properties.

6.1 Computationally Intensive Tasks

Continual Learning

Continual learning (CL) addresses the problem of catastrophic forgetting by using strategies such as experience replay, which stores representative samples from previous tasks as a buffer to recall knowledge. Dataset distillation, which involves highly compressed representations, is an alternative to traditional sampling methods. There are currently two experimental settings for using distillation in CL. [Zhao *et al.*, 2021; Zhao Bo, 2021] use different datasets, SVHN, MNIST, and USPS, three handwritten digit recognition datasets, and take EEIL [Castro *et al.*, 2018] as the baseline for continuous learning. In the study of [Zhao and Bilen, 2021], the experimental settings are changed to incremental class learning on the CIFAR100 dataset. The researchers establish a baseline using the GDumb method [Prabhu *et al.*, 2020] and randomly divided 100 classes into 5 and 10 learning steps, with 20 and 10 classes per step respectively.

Neural Architecture Search

Neural architecture search (NAS) is known to be expensive as it involves training multiple architectures on the entire training dataset and selecting the best-performing one on the validation set. To address this issue, researchers have proposed using a distilled dataset as a proxy of the entire dataset, which can effectively identify the best network. Related experiments on the CIFAR10 dataset have been reported in DC [Zhao *et al.*, 2021], DSA [Zhao Bo, 2021], and DM [Zhao and Bilen, 2021]. These studies construct a search space of 720 ConvNets by varying hyperparameters such as network depth, width, activation function, normalization, and pooling layers over a uniform grid. The effectiveness of the distilled

dataset was evaluated using the Spearman's rank correlation coefficient between the validation accuracies obtained by the proxy dataset and the entire dataset. A higher correlation value indicates that the proxy dataset is more reliable.

6.2 Privacy

Dataset Construction

Machine learning is vulnerable to a variety of privacy attacks, such as membership inference attacks [Shokri *et al.*, 2017], model inversion attacks [Fredrikson *et al.*, 2015; Carlini *et al.*, 2023], and gradient inversion attacks [Geng *et al.*, 2021; Geng *et al.*, 2023], where attackers attempt to infer task-independent private information from the target model, and even recover the original training data. Additionally, data collection and publishing raise privacy and copyright concerns. [Dong *et al.*, 2022; Zhou *et al.*, 2022] have shown that models trained on synthetic data are robust to both loss-based and likelihood-based membership inference attacks. To ensure that the distilled samples cannot be inferred from real ones, [Nguyen *et al.*, 2020] implemented the KIP_ρ variant, which randomly initialized ρ proportion of each image and kept them unchanged during training. This idea was later followed by RFAD_ρ [Loo *et al.*, 2022]. [Chen *et al.*, 2022] added a differential privacy (DP) mechanism [Dwork, 2006] to the distillation process to provide rigorous privacy guarantees. Medical data often requires strict anonymization before publication, [Li *et al.*, 2022] propose to dataset distillation to construct privacy-preserving datasets.

Federated Learning

Federated learning (FL) is an emerging technology that enables different clients to collaboratively train a shared model without sharing their local data. It faces challenges such as high bandwidth requirements for uploading large model updates and a lack of strict privacy guarantees. There are several works that propose to combine dataset distillation in FL. [Hu *et al.*, 2022; Xiong *et al.*, 2022] suggest sharing lightweight synthetic datasets instead of sharing model updates, since the distilled dataset size is generally smaller. However, this may introduce bias and increase the computational load, which can negatively impact the performance and efficiency of FL.

6.3 Robustness

Data Poisoning Attacks

Distilled data lose its fidelity and may not be visually distinguishable from its original contents, making it vulnerable to data poisoning attacks and difficult to detect. Studies have shown that a small number of these poisoned samples can significantly reduce the accuracy of a model's predictions on a specific category. [Wang *et al.*, 2018] propose a study on data poisoning attacks using dataset distillation. [Liu *et al.*, 2023] propose two backdoor attacks on distilled data by injecting triggers into the synthetic data during the distillation process, either in the initial stage or throughout the entire process.

Improve Model Robustness

Dataset distillation can also be used as a means of improving its robustness. Researchers have proposed using optimization techniques to learn a robust distilled dataset, such that a classifier trained on this dataset will have improved resistance to adversarial attacks. [Tsilivis *et al.*, 2022] have combined the KIP method with adversarial training to enhance the robustness of the distilled dataset. [Wu *et al.*, 2022] approached the problem of dataset learning as a tri-level optimization problem to obtain a distilled dataset that minimizes robust error on the data-parameterized classifier.

7 Conclusion and Future Directions

In this paper, we present a systematic review of recent advances in dataset distillation. We introduce a novel taxonomy that categorizes existing works from various perspectives. We find that most existing efforts are geared toward image datasets, whereas the handling of discrete text and graph data remains a significant challenge. There is a limited exploration of robustness, and further research is necessary as the technology gains wider adoption. Our study demonstrates the research landscape in this field and suggests directions for future work.

7.1 Computational Efficiency

The computational efficiency of dataset distillation is an important consideration, as many current methods for dataset distillation can be computationally expensive, particularly for larger datasets. The goal of dataset distillation is to reduce the size of a dataset while preserving its key features and patterns, but this process often requires complex optimization and clustering algorithms, which can be computationally intensive. Methods like MTT [Cazenavette *et al.*, 2022], KIP [Nguyen *et al.*, 2020], and FRePo [Zhou *et al.*, 2022] can cause GPU memory bottlenecks when the number of images per class (IPC) increases. While the DM [Zhao and Bilen, 2021] approach proposes using distribution matching to avoid model training, and RFAD [Loo *et al.*, 2022] proposes using NNGP to reduce the computational complexity of kernel ridge regression, the computational efficiency of distillation still requires improvement, particularly for larger datasets.

7.2 Performance Degradation on Larger IPC

According to [Cui *et al.*, 2022], current dataset distillation methods perform well only when the number of images per class (IPC) is relatively small. As the IPC increases, the performance of most distillation methods deteriorates and becomes similar to that of random sampling. Therefore, it is important to explore whether dataset distillation can overcome this limitation and maintain superior performance on larger datasets.

7.3 Weak Labels

Currently, research on dataset distillation primarily focuses on classification tasks. However, its potential for more complex tasks, such as image detection and segmentation, named entity recognition, summarization, and machine translation, remains untapped. Exploring the technique's effectiveness on these tasks could provide deeper insights into data characteristics and the inner workings of AI.

Acknowledgements

This work is partially funded by the European Union's Horizon 2020 Research and Innovation Program through Marie Skłodowska-Curie Grant 860627 (CLoud ARtificial Intelligence For pathologY (CLARIFY) Project). We also sincerely thank Awesome-Dataset-Distillation for its comprehensive and timely DD publication summary.

References

[Abdal *et al.*, 2019] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019.

[Andrychowicz *et al.*, 2016] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*, 29, 2016.

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[Carlini *et al.*, 2023] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. *arXiv preprint arXiv:2301.13188*, 2023.

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

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

[Chen *et al.*, 2022] Dingfan Chen, Raouf Kerkouche, and Mario Fritz. Private set generation with discriminative information. *arXiv preprint arXiv:2211.04446*, 2022.

[Cui *et al.*, 2022] Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Dc-bench: Dataset condensation benchmark. *arXiv preprint arXiv:2207.09639*, 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 *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[Deng, 2022] Olga Deng, Zhiwei; Russakovsky. Remember the past: Distilling datasets into addressable memories for neural networks. *arXiv preprint arXiv:2206.02916*, 2022.

[Dong *et al.*, 2022] Tian Dong, Bo Zhao, and Lingjuan Lyu. Privacy for free: How does dataset condensation help privacy? *arXiv preprint arXiv:2206.00240*, 2022.

[Dwork, 2006] Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33*, pages 1–12. Springer, 2006.

[Fredrikson *et al.*, 2015] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[Geng *et al.*, 2021] Jiahui Geng, Yongli Mou, Feifei Li, Qing Li, Oya Beyan, Stefan Decker, and Chunming Rong. Towards general deep leakage in federated learning. *arXiv preprint arXiv:2110.09074*, 2021.

[Geng *et al.*, 2023] Jiahui Geng, Yongli Mou, Qing Li, Feifei Li, Oya Beyan, Stefan Decker, and Chunming Rong. Improved gradient inversion attacks and defenses in federated learning. *IEEE Transactions on Big Data*, 2023.

[Graves *et al.*, 2017] Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *international conference on machine learning*, pages 1311–1320. PMLR, 2017.

[Hu *et al.*, 2022] Shengyuan Hu, Jack Goetz, Kshitiz Malik, Hongyuan Zhan, Zhe Liu, and Yue Liu. Fedsynth: Gradient compression via synthetic data in federated learning. *arXiv preprint arXiv:2204.01273*, 2022.

[Jacot *et al.*, 2018] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[Jin *et al.*, 2021] Wei Jin, Lingxiao Zhao, Shichang Zhang, Yozen Liu, Jiliang Tang, and Neil Shah. Graph condensation for graph neural networks. *arXiv preprint arXiv:2110.07580*, 2021.

[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 *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 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. *arXiv preprint arXiv:2205.14959*, 2022.

[Koehn, 2005] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand, September 13-15 2005.

[Konyushkova *et al.*, 2017] Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. Learning active learning from

data. *Advances in neural information processing systems*, 30, 2017.

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

[Li *et al.*, 2022] Guang Li, Ren Togo, Takahiro Ogawa, and Miki Haseyama. Dataset distillation for medical dataset sharing. *arXiv preprint arXiv:2209.14603*, 2022.

[Liu *et al.*, 2022a] Mengyang Liu, Shanchuan Li, Xinshi Chen, and Le Song. Graph condensation via receptive field distribution matching. *arXiv preprint arXiv:2206.13697*, 2022.

[Liu *et al.*, 2022b] Songhua Liu, Kai Wang, Xingyi Yang, Jingwen Ye, and Xinchao Wang. Dataset distillation via factorization. *arXiv preprint arXiv:2210.16774*, 2022.

[Liu *et al.*, 2023] Yugeng Liu, Zheng Li, Michael Backes, Yun Shen, and Yang Zhang. Backdoor attacks against dataset distillation. *arXiv preprint arXiv:2301.01197*, 2023.

[Loo *et al.*, 2022] Noel Loo, Ramin Hasani, Alexander Amini, and Daniela Rus. Efficient dataset distillation using random feature approximation. *arXiv preprint arXiv:2210.12067*, 2022.

[Maclaurin *et al.*, 2015] Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pages 2113–2122. PMLR, 2015.

[Nguyen *et al.*, 2020] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. *arXiv preprint arXiv:2011.00050*, 2020.

[Nguyen *et al.*, 2021] Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. *Advances in Neural Information Processing Systems*, 34:5186–5198, 2021.

[Platt and others, 1999] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[Prabhu *et al.*, 2020] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *European conference on computer vision*, pages 524–540. Springer, 2020.

[Schmidhuber, 2015] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[Sener and Savarese, 2017] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[Shokri *et al.*, 2017] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

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

[Sucholutsky and Schonlau, 2021] Ilia Sucholutsky and Matthias Schonlau. Soft-label dataset distillation and text dataset distillation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[Tsilivis *et al.*, 2022] Nikolaos Tsilivis, Jingtong Su, and Julia Kempe. Can we achieve robustness from data alone? *arXiv preprint arXiv:2207.11727*, 2022.

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

[Wang *et al.*, 2022] Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. Cafe: Learning to condense dataset by aligning features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12196–12205, 2022.

[Warden, 2018] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

[Wu *et al.*, 2022] Yihan Wu, Xinda Li, Florian Kerschbaum, Heng Huang, and Hongyang Zhang. Towards robust dataset learning. *arXiv preprint arXiv:2211.10752*, 2022.

[Xiong *et al.*, 2022] Yuanhao Xiong, Ruochen Wang, Minhao Cheng, Felix Yu, and Cho-Jui Hsieh. Feddm: Iterative distribution matching for communication-efficient federated learning. *arXiv preprint arXiv:2207.09653*, 2022.

[Zhao and Bilen, 2021] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. *ArXiv*, abs/2110.04181, 2021.

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

[Zhao *et al.*, 2021] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *ICLR*, 1(2):3, 2021.

[Zhao Bo, 2021] Bilen Hakan Zhao Bo. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021.

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