Distill Gold from Massive Ores: Bi-level Data Pruning towards Efficient Dataset Distillation

Yue Xu[1], Yong-Lu Li[1*], Kaitong Cui[1], Ziyu Wang[1],
Cewu Lu[1], Yu-Wing Tai[2], and Chi-Keung Tang[3]

[1] Shanghai Jiao Tong University
{silicxuyue,yonglu_li,ckt0704,wangxiaoyi2021,lucewu}@sjtu.edu.cn,
ckt0704@gmail.com
[2] Dartmouth College
yuwing@gmail.com
[3] Hong Kong University of Science and Technology
cktang@cs.ust.hk

Abstract. Data-efficient learning has garnered significant attention, especially given the current trend of large multi-modal models. Recently, dataset distillation has become an effective approach by synthesizing data samples that are essential for network training. However, it remains to be explored which samples are essential for the dataset distillation process itself. In this work, we study the data efficiency and selection for the dataset distillation task. By re-formulating the dynamics of distillation, we provide insight into the inherent redundancy in the real dataset, both theoretically and empirically. We propose to use the empirical loss value as a static data pruning criterion. To further compensate for the variation of the data value in training, we find the most contributing samples based on their causal effects on the distillation. The proposed selection strategy can efficiently exploit the training dataset, outperform the previous SOTA distillation algorithms, and consistently enhance the distillation algorithms, even on much larger-scale and more heterogeneous datasets, *e.g.*, full ImageNet-1K and Kinetics-400. We believe this paradigm will open up new avenues in the dynamics of distillation and pave the way for efficient dataset distillation. **Our code is available on** https://github.com/silicx/GoldFromOres-BiLP.

Keywords: Dataset Distillation · Data Pruning

1 Introduction

Data is the core of deep learning. In the era of large data and models [22,36,42] as their size and complexity continue to grow, *data-efficient* learning is increasingly crucial for achieving high performance with a limited computing budget. Techniques such as pruning, quantization, and knowledge distillation have been developed to reduce model size without sacrificing performance. Recently, *dataset*
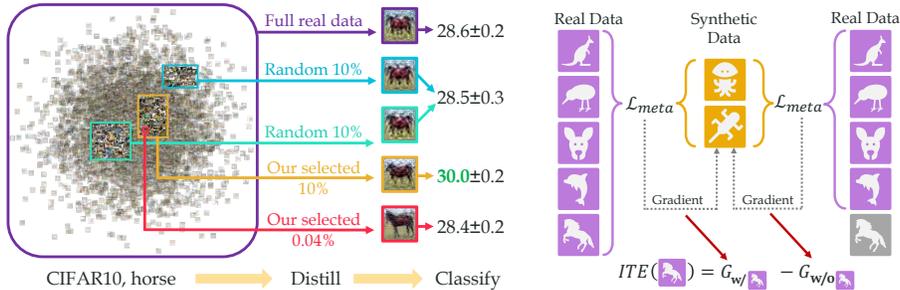
---
* Corresponding author.

Fig. 1: (1) **Left**: *Severe data redundancy* in the dataset distillation process. Taking CIFAR10 and DC [53] as an example, pruning 90% of the real data does not reduce the performance. With our proposed empirical loss criterion, only 0.04% real samples are sufficient for distillation and 10% samples outperform the full real dataset. (2) **Right**: Building upon the empirical loss, we propose to dynamically select samples by their ITE value, which is the difference of meta gradient with/without the sample.

distillation [44] has emerged as a promising approach toward data-efficient AI, where a small and condensed dataset (namely *synthetic* dataset) is learned from the whole large dataset (namely *real* dataset), to maximize the performance of models trained on distilled synthetic data.

Currently, the existing distillation algorithms focus on the evolution of the matching strategies for real and synthetic data [3, 35, 39, 52, 53] or refining and accelerating the bi-level optimization [26, 27, 32, 44]. However, preliminary experiments show that the real dataset could be so redundant during the dataset distillation itself that pruning part of the dataset will not reduce the distillation performance (as shown in Fig. 1). Therefore, an appropriate approach to finding the most valuable real data would help the distillation algorithms to learn the most key data patterns and knowledge. Currently, it is still under-explored how to analyze the samples' value on the dataset distillation task and efficiently select the real dataset. Recently, YOCO [10] proposes an interesting task of pruning the synthetic dataset. DREAM [24] prunes the real data within each batch from the aspect of feature distribution. In comparison, we would principally analyze the intrinsic worth of each real data sample and its relevance to the distillation optimization process from a more holistic perspective.

In this work, we revisit the selection and utilization of real datasets for dataset distillation. We first describe the dataset distillation as the matching of training dynamics on real and synthetic data, where the dynamics are formulated with the neural tangent kernel (NTK) [12]. Further analysis shows that the small scale of synthetic data motivates pruning those real data samples with large empirical loss values. The stability of NTK for wide neural networks also allows for dropping these real samples at the beginning of dataset distillation, which indicates **redundancy** in real data and offers efficiency. To support the theoretical analysis, we conduct comprehensive experiments of pruning real data before distillation. In most scenarios, pruning some real data does not reduce

the distillation performance. For instance, with CIFAR10 for DC [53] and single instance-per-class (IPC=1), removing over **99.9**% of the real data does not affect its performance. We argue that leveraging redundancy in dataset distillation is more significant than regular machine learning applications since synthetic datasets usually possess low capacity, as shown in Fig. 1.

However, pruning real data before distillation does not consistently enhance the distillation performance, which implies that the value or utility of the real samples could *vary* during the training process. To identify the importance of real samples, we measure their contribution to the learning of the synthetic data from the perspective of **causal effect**. We compare the synthetic gradient with or without each real sample, which is essentially the individual treatment effect (ITE) of the real samples for the dataset distillation task. Based on the principles, we propose a new and concise real data sampling algorithm, which can be implemented as a plug-and-play for most dataset distillation methods. In comprehensive experiments, our approach can thus efficiently use the key samples and consistently enhance the base algorithm.

Overall, our contributions are: 1) In-depth insight on data redundancy for dataset distillation. 2) Two core principles for real sample selection, supported both theoretically and empirically. 3) A plug-and-play selection strategy for practical dataset distillation to enhance SOTA methods.

2   Related Work

Dataset Distillation is a process of condensing a large dataset into a smaller and more representative dataset while maintaining the performance, which has been applied to various domains including images, text [28, 29], videos [45], graph [13, 14], time series [9, 25], medical [20, 21] and multimodal data [46, 47], etc. Existing approaches can be roughly classified into: 1) *Meta-Model Matching* maintains the transferability of the synthetic data, by optimizing the loss on the original dataset of models trained on the synthetic data. In [44] the task of data distillation and the meta-model matching framework was first proposed. In [32] kernel ridge regression was exploited to reduce computational cost and is further extended to infinite wide networks [33]. In [55] the optimization of synthetic data/classifier and feature extractor was separated. In [27] the meta-gradient was computed by exploiting implicit gradients. 2) *Gradient Matching* aligns the gradients of the synthetic and real datasets. It was proposed in [53] and further improved in [51] to perform the same image augmentations on both the real and synthetic data. 3) *Distribution Matching*, where [52] matches feature distributions of the synthetic and real data, which is simple but effective. In [43] layer-wise feature alignment and early exit conditions are used to promote it. In [54] it was further enhanced with regularizers and model pool. 4) *Trajectory Matching:* In [3] the authors proposed MTT to match the training trajectory of the model parameters and in [6] the memory consumption of MTT was reduced and label learning was used. 5) *Factorization* of synthetic data can reduce the storage burden and share knowledge among instances. For example, [16] uses a

strategy of putting multiple images on one synthetic sample. [8] decomposes the synthetic data to the linear network hallucinators and bases, while [23] uses a convolutional network. [19] maintains a smaller base space to further reduce the storage. [38] employs frequency domain factorization. 6) *Bayesian Pseudocoreset* is a family of algorithms that learn the synthetic data with Bayesian inference [15, 30, 40]. Beyond these categories, SRe2L [49] utilizes a 3-stage learning paradigm to decouple the segregation of inner-loop and outer-loop optimization.

Data Selection/Pruning reduces the training data without significantly affecting performance. Classic data selection often calculates a scalar utility score for each sample based on predefined criteria [2, 37, 41] and filters samples based on scores. Some data pruning methods also consider the interaction between samples. For example, [48] examines generalization influence to reduce training data, which aims to identify the smallest subset to satisfy the expected generalization ability. In comparison, data distillation [44] synthesizes new and smaller data, and significantly outperforms data pruning with the same data storage.

3   Preliminaries

3.1   Formulation of Dataset Distillation

Before delving deeper into the real data selection, we would investigate the training dynamics of dataset distillation. Given a real dataset $\mathcal{D}_r = \{x_r^{(i)}, y_r^{(i)}\}_{i=1}^{M_r}$, dataset distillation is to learn a synthetic dataset $\mathcal{D}_s = \{x_s^{(i)}, y_s^{(i)}\}_{i=1}^{M_s}$ that has smaller size ($M_s \ll M_r$) and could train a network to similar performance to full real dataset on specific task. More formally, let $u = f(x; \boldsymbol{\theta})$ be a scalar output network [4] and $\mathcal{L}(\mathcal{D}, \boldsymbol{\theta}) = \sum_{i=1}^{M} \ell(u^{(i)}, y^{(i)})$ be the loss function that measures the model empirical risk on a dataset $\mathcal{D}$. The dynamics of gradient descent optimization are described by differential equations:

$$\dot{\boldsymbol{\theta}}_r = -\frac{\partial \mathcal{L}(\mathcal{D}_r, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\sum_{i=1}^{M_r} \frac{\partial \ell(u_r^{(i)}, y_r^{(i)})}{\partial u_r^{(i)}} \frac{\partial u_r^{(i)}}{\partial \boldsymbol{\theta}},$$
$$\dot{\boldsymbol{\theta}}_s = -\frac{\partial \mathcal{L}(\mathcal{D}_s, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\sum_{i=1}^{M_s} \frac{\partial \ell(u_s^{(i)}, y_s^{(i)})}{\partial u_s^{(i)}} \frac{\partial u_s^{(i)}}{\partial \boldsymbol{\theta}}, \tag{1}$$

where $\boldsymbol{\theta}_r(t)$ and $\boldsymbol{\theta}_s(t)$ are functions of timestamp t which describe the training trajectory of parameters on the real or synthetic dataset. We omit "(t)" for clarity. $\dot{\boldsymbol{\theta}}$ indicates its derivative w.r.t t: $\frac{\partial \boldsymbol{\theta}}{\partial t}, \dot{\boldsymbol{\theta}}_s = \frac{\partial \boldsymbol{\theta}_s}{\partial t}$, and as a common process, the learning rate of gradient descent is absorbed to timestamp t for simplicity. The outputs of both networks on the real dataset are computed for evaluation:

$$u_r^{(i)} = f(x_r^{(i)}; \boldsymbol{\theta}_r), \quad u_s^{(i)} = f(x_s^{(i)}; \boldsymbol{\theta}_s). \tag{2}$$

Note that $u_r^{(i)}$ and $u_s^{(i)}$ are also function of t.

The current dataset distillation can be categorized into two types:

---

[4] The analysis covers scalar functions, but can also generalize to vector functions.

(1) **Empirical Risk Minimization** includes DD [44], KIP [32, 33] and their variants [26, 27], which minimize the empirical risk on the real dataset of the network trained on synthetic data:

$$\mathcal{S} = \arg\min_{\mathcal{S}} \sum_{i=1}^{M_r} \ell(u_s^{(i)}|_{t=+\infty}, y_r^{(i)}). \tag{3}$$

(2) **Training Dynamics Matching** is a simpler agent task for dataset distillation since the converged network $\boldsymbol{\theta}_s^{(i)}|_{t=+\infty}$ is usually intractable for bi-level optimization. Its ultimate target is to match the whole training trajectory on the real and synthetic datasets:

$$\mathcal{S} = \arg\min_{\mathcal{S}} D(u_s^{(i)}(t), u_r^{(i)}(t)), \tag{4}$$

where D is a distance metric. The strategies include gradient trajectory matching [3, 16, 24, 53], feature matching [43, 52, 54], loss matching [39], *etc.*

The generation-based methods [4, 49] are exceptions since they do not involve the network dynamics.

3.2 Redundancy in Dataset Distillation

Currently, the intrinsic value of each real sample under the dataset distillation scenario has not been thoroughly investigated, yet we observe the redundancy within the real dataset which underscores the potential significance of this research area. We randomly drop some portions of the real data to find their maximum pruning ratio that could maintain the distillation accuracy. Comprehensive experiments are conducted on various datasets, networks, distillation algorithms, initialization methods, and synthetic data sizes (represented by instance-per-class, IPC). We use the default parameters of each method, which are detailed in the supplementary material. As shown in Tab. 1, severe data redundancy widely exists in various dataset distillation settings. We will analyze and propose selection criteria dedicated to dataset distillation in the following.

4 Methodology

4.1 Empirical Loss Pruning

We study the real data selection of dataset distillation based on the previous formulation. Considering that $\dot{u} = \langle \frac{\partial u}{\partial \boldsymbol{\theta}}, \dot{\boldsymbol{\theta}} \rangle$, the dynamics of outputs can be depicted by differential equations:

$$
\begin{aligned}
\dot{u}_r^{(i)} &= -\sum_{j=1}^{M_r} \frac{\partial \ell(u_r^{(j)}, y_r^{(j)})}{\partial u_r^{(j)}} \langle \frac{\partial u_r^{(i)}}{\partial \boldsymbol{\theta}_r}, \frac{\partial u_r^{(j)}}{\partial \boldsymbol{\theta}_r} \rangle = -\sum_{j=1}^{M_r} \frac{\partial \ell(u_r^{(j)}, y_r^{(j)})}{\partial u_r^{(j)}} \boldsymbol{K}_r^{(ij)}, \\
\dot{u}_s^{(i)} &= -\sum_{j=1}^{M_s} \frac{\partial \ell(u_s^{(j)}, y_s^{(j)})}{\partial u_s^{(j)}} \langle \frac{\partial u_s^{(i)}}{\partial \boldsymbol{\theta}_s}, \frac{\partial u_s^{(j)}}{\partial \boldsymbol{\theta}_s} \rangle = -\sum_{j=1}^{M_s} \frac{\partial \ell(u_s^{(j)}, y_s^{(j)})}{\partial u_s^{(j)}} \boldsymbol{K}_s^{(ij)},
\end{aligned}
\tag{5}
$$

Table 1: Maximum pruning ratio with **random selection** among different datasets, algorithms and synthetic data size.

Dataset	IPC	DC [53]	DSA [51]	DM [52]	MTT [3]	IDC [16]	IDM [54]
CIFAR10 [17]	1	90%	85%	85%	60%	50%	40%
	10	70%	70%	60%	10%	30%	10%
	50	50%	70%	50%	20%	40%	20%
CIFAR100 [17]	1	50%	40%	70%	20%	60%	50%
	10	50%	40%	50%	10%	50%	20%
SVHN [31]	1	60%	60%	85%	90%	60%	30%
	10	80%	70%	95%	80%	30%	5%
TinyImageNet [18]	1	40%	30%	60%	50%	40%	60%
	10	50%	50%	50%	20%	30%	50%

where $\boldsymbol{K}_r$ and $\boldsymbol{K}_s$ are neural tangent kernels [12] (NTK), and matching the derivatives $\dot{u}_r$ and $\dot{u}_s$ is a sufficient and necessary condition of target Eq. (4).

Based on the formulation of dataset distillation dynamics, we notice that $\boldsymbol{K}_r^{(ij)}$ is rank-deficient since synthetic data is much smaller than the real dataset. Since $\dot{u}^i$ is the linear combination of kernel matrix $\boldsymbol{K}$, the rank-deficiency of $\boldsymbol{K}_r^{(ij)}$ leads to the mismatch of complexity of $\dot{u}_s^i$ and $\dot{u}_r^i$, making synthetic dynamics $\dot{u}_s^i$ hard to approximate to $\dot{u}_r^i$. So we argue that **the real dataset has significant redundancy for dataset distillation**. To reduce the optimization difficulty for a more stable distillation process, and also for a more data-efficient distillation algorithm, the dataset distillation needs to select the most valuable real data samples. A more intuitive explanation is that the small synthetic data cannot memorize the larger scale real dataset, so selecting the important but easy samples will help the algorithm.

Furthermore, Eq. (5) also reveals that samples with small gradient value $\|\frac{\partial \ell(u_r, y_r)}{\partial u_r}\|$ are more important for distillation, since pruning the samples with large gradient is equivalent to reduce the kernel rank of real data. Note that for common loss functions like MSE or cross-entropy, the $\|\frac{\partial \ell(u_r, y_r)}{\partial u_r}\|$ has the same monotonicity to $\ell(u_r, y_r)$, so we directly **adopt the empirical loss** $\ell(u_r, y_r)$ **as our real data selection criterion** to avoid additional computation. Please refer to the supplementary for the proofs.

Early Pruning. A principal property of the neural tangent kernels is they tend to be constant for wide neural networks. So in Eq. (5), the NTK for real data $\boldsymbol{K}_r$ is stable during training. So the empirical loss can be regarded as a **static criterion** which does not vary during training. In practice, to maximize the efficiency, we prune the real data in one time **before the beginning** of the distillation algorithm rather than select different samples at each iteration. The empirical loss of each sample is computed by training the network on the real dataset for multiple trials and taking the average loss value.

Table 2: Maximum pruning ratio with **empirical loss selection** among different datasets, algorithms and synthetic data sizes.

Dataset	IPC	DC [53]	DSA [51]	DM [52]	MTT [3]	IDC [16]	RFAD [26]	IDM [54]
CIFAR10 [17]	1	99.5%	99.5%	99.5%	60%	85%	70%	30%
	10	30%	60%	50%	20%	50%	20%	10%
	50	70%	85%	40%	20%	30%	20%	30%
CIFAR100 [17]	1	95%	97%	99.5%	60%	80%	80%	70%
	10	90%	80%	90%	20%	60%	80%	20%
SVHN [31]	1	80%	95%	99%	99%	40%	20%	60%
	10	20%	60%	85%	95%	50%	5%	10%
TinyImageNet [18]	1	97%	99%	99.5%	70%	40%	-	95%
	10	80%	70%	97%	30%	40%	-	60%
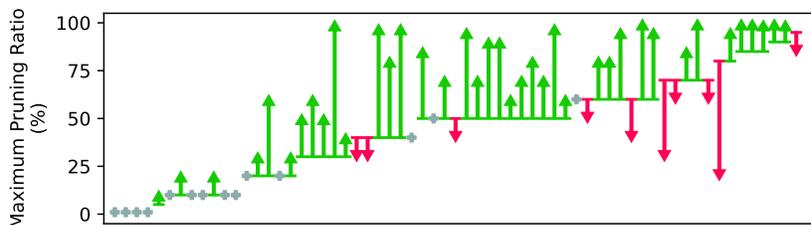
Fig. 2: The comparison of maximum pruning ratio between random selection and empirical loss-based selection: the arrows point from random to the loss-based method. Green: selection by loss is superior; Red: selection by loss is inferior to random.

Empirical Justification Besides the theoretical analysis, we conduct extensive experiments to support our selection strategy. To examine the effectiveness of pruning, we compare the maximum pruning ratio that maintains the distillation performance, and a larger maximum pruning ratio indicates a larger redundancy.

Comprehensive comparison experiments on various datasets, networks, distillation algorithms, initialization methods, and synthetic data sizes (represented by instance-per-class, IPC) are given in Tab. 2. We take the mean and standard deviation of the accuracy of 5 random removal trials. We use the default parameters of each method, which are detailed in the supplementary. The results show that severe data redundancy widely exists in various dataset distillation settings. For many datasets and algorithms, less than 30% samples are sufficient for dataset distillation. We also visualize the comparison between random selection and empirical loss selection in Fig. 2. **Pruning with empirical loss consistently outperforms random selection**. We also find dropping data does not drop the cross-architecture transferability in supplementary.

Surprisingly, we also observe that **dropping real data can sometimes improve performance**. We show the best performance during data pruning in Tab. 3. In almost all cases, the data selection can notably promote distillation accuracy. This implies that some samples may be "detrimental" to the distillation and the empirical loss is a proper criterion to cancel these negative impacts. This

Table 3: Best performance with data dropping. The performance difference between the *full real dataset* and *pruned dataset* are shown in parentheses (†: compare to our reproduced accuracy).

Dataset	IPC	DC [53]	DSA [51]	DM [52]	MTT [3]	RFAD [26]	IDM [54]
CIFAR10 [17]	1	30.0±0.2 (+**1.7**)	30.9±0.1 (+**2.6**)	29.7±0.3 (+**3.7**)	46.3±0.8 (+0.2)	53.7±0.9 (+0.1)	46.3±0.4 (+**0.7**)
	10	44.9±0.4 (+0.0)	52.4±0.2 (+0.2)	50.0±0.2 (+**1.1**)	65.7±0.3 (+**0.4**)	66.7±0.2 (+**0.4**)	58.9±0.3 (+**0.3**)
	50	54.9±0.5 (+**1.0**)	61.5±0.7 (+**0.9**)	63.4±0.2 (+**0.4**)	72.0±0.4 (+**0.4**)	71.9±0.2 (+**0.8**)	67.8±0.2 (+**0.3**)
CIFAR100 [17]	1	14.1±0.1 (+**1.3**)	15.6±0.1 (+**1.7**)	14.9±0.5 (+**3.5**)	24.6±0.4 (+**0.3**)	30.4±0.6 (+**4.1**)	25.9±0.3 (+**5.8**)
	10	26.5±0.3 (+**1.3**)	32.5±0.4 (+0.2)	32.4±0.3 (+**2.7**)	40.1±0.5 (+0.0)	38.4±0.2 (+**6.4**)	45.9±0.1 (+**0.8**)
SVHN [31]	1	32.2±0.5 (+**1.0**)	28.9±1.3 (+0.1)	29.8±0.5 (+**6.3**†)	43.0±1.1 (+**3.2**†)	53.0±0.2 (+**0.8**†)	71.9±1.0 (+**1.6**)
	10	76.2±0.6 (+0.1)	80.0±0.8 (+**0.8**)	74.6±0.3 (+**0.9**†)	78.1±0.5 (+**0.9**†)	74.2±0.2 (+0.1†)	81.9±0.5 (+0.1)
TinyImage-Net [18]	1	4.9±0.1 (+**0.2**†)	4.3±0.0 (+**0.6**†)	4.8±0.1 (+**0.9**)	9.0±0.4 (+0.2)	-	11.4±0.2 (+**1.3**)
	10	12.8±0.0 (+**0.2**†)	14.8±0.4 (+**2.3**†)	17.5±0.1 (+**4.6**)	23.8±0.3 (+**0.6**)	-	22.9±0.4 (+**1.0**)

observation inspires new approaches that leverage data utility and exploit all data samples in different quality, enabling future analysis of network dynamics and dataset distillation.

4.2   Causal Effect on Synthetic Data

However, the pruning lacks availability in some scenarios when IPC is large on certain algorithms, indicating that sample importance may vary at different stages of the training process; some initially pruned samples can be beneficial in the latter stage. So to take a step further, we investigate the real samples' contribution to the synthetic data during training as compensation for the static empirical loss criterion.

We first generalize the optimization targets of various dataset distillations into a meta-loss function $\mathcal{L}_{\mathrm{meta}}(\mathcal{D}_r, \mathcal{D}_s)$. *E.g.*, the meta-loss of DC [53] is the cosine distance between the gradients of real and synthetic; the meta-loss of DM [52] is the MMD between real and synthetic features. The synthetic data $\mathcal{D}_s$ is learned by gradient descent on the meta-loss function. To examine the causal effects of real samples on the updating and learning of synthetic data, we could observe the consequence on the meta gradient after applying **causal intervention** on the real dataset, *i.e.*, removing a certain real sample. More specifically, we regard the presence of real samples $x_r \in \mathcal{D}_r$ as some binary "treatment" (T), and the meta gradient of synthetic data $\frac{\partial L_{\mathrm{meta}}}{\partial \mathcal{D}_s}$ as some "effect" (Y). Thus for a certain individual $\mathcal{D}_s$, the causal effect of each real sample to the distillation can be quantized by the individual treatment effect (ITE) [34]:

$$ITE(x_r) = Y_{T=1} - Y_{T=0} = \frac{\partial L_{meta}(\mathcal{D}_r, \mathcal{D}_s)}{\partial \mathcal{D}_s} - \frac{\partial L_{meta}(\mathcal{D}_r \setminus \{x_r\}, \mathcal{D}_s)}{\partial \mathcal{D}_s}, \quad (6)$$

which is the difference between the gradient on the synthetic dataset with or without the real sample x_r, and can be obtained on any dataset distillation algorithm by computing two meta gradients for each real sample. We use the L2-norm of ITE gradient $\|ITE(x_r)\|_2$ as the causal criterion of a real sample x_r. We remove the samples with both the *smallest* and the *largest* ITE criterion values

since the small ITE value samples contribute little to the synthetic learning, while the large ITE value samples could be outliers for the distillation, which enhances the variance and instability during training. The ablation analysis in Sec. 5.4 also shows the samples with intermediate-level ITE are more important to the distillation task.

Due to the heavy computation of the meta gradient, we apply three optimization techniques for applicable efficiency.

Taylor Approximation of ITE. The causal criterion needs additional meta gradient computations for each of the K real samples (the second term in Eq. (6)), therefore increasing the training time by nearly K times. Thankfully, we can reduce the computation burden on a special family of meta loss functions that are *additive variable separable*: $\mathcal{L}_{meta}(G, \mathcal{D}_s)$, $G = \sum_{x \in \mathcal{D}_r} g(x)$, which covers most of the meta matching algorithms, *e.g.*, $g(x)$ indicates per-sample gradient for DC [53], sample feature vector for DM [52]. In this case, we could leverage the Taylor approximation for ITE value:

$$
\begin{aligned}
ITE(x_r) &= \frac{\partial}{\partial \mathcal{D}_s} \left[\mathcal{L}_{meta}(G, \mathcal{D}_s) - \mathcal{L}_{meta}(G - g(x_r), \mathcal{D}_s) \right] \\
&= \frac{\partial}{\partial \mathcal{D}_s} \left[\left(\frac{\partial L_{meta}(G, \mathcal{D}_s)}{\partial G} \right)^{\top} g(x_r) + h.o.t \right] \\
&\approx \frac{\partial^2 L_{meta}(G, \mathcal{D}_s)}{\partial \mathcal{D}_s \, \partial G} \cdot g(x_r),
\end{aligned}
\tag{7}
$$

where we vectorize G, $g(x)$, and $\mathcal{D}_s$ for clarity. The first term is part of the Hessian matrix of $\mathcal{L}_{meta}$ and not related to specific real sample x_r so it could be computed in one backward pass. The second term has to be computed during the forward pass of meta loss. So we could compute the ITE values for the real batch with one meta gradient computation and a matrix multiplication, which significantly reduces the pruning time. We also analyze the error of the approximation in the supplementary and ignoring high order terms only brings 3% shifting of the sample ranking that is negligible for pruning.

Estimation of Global ITE Distribution. We hope to prune the samples with small ITE values among the full real dataset, but it is computationally expensive to exhaustively evaluate ITE values of the *full* dataset at each iteration, especially with mini-batch optimization methods. Inspired by batch normalization [11], we maintain the global running mean $\hat{\mu}$ and variance $\hat{\sigma}^2$ of ITE values of the real dataset. For each mini-batch, we compute the mean μ_B and variance σ_B^2 within the batch, and update the global statistics by:

$$
\hat{\mu} \leftarrow (1 - \eta)\hat{\mu} + \eta\mu_B, \quad \hat{\sigma}^2 \leftarrow (1 - \eta)\hat{\sigma}^2 + \eta\sigma_B^2.
\tag{8}
$$

Then, given a pruning ratio β, we find the two quantile points $z_{\frac{\beta}{2}}$ and $z_{1-\frac{\beta}{2}}$ of the normal distribution $z \sim \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$, and select the samples with ITE value within $(z_{\frac{\beta}{2}}, z_{1-\frac{\beta}{2}})$, so that the overall selection rate is $1 - \beta$.

Lazy Selection. We prune and update the real data batch every several iterations to reduce the overhead of selection.

Algorithm 1 **Bi**-**L**evel Data **P**runing (**BiLP**)

Input: Real dataset $\mathcal{D}_r$, preemptive pruning rate α, adaptive pruning rate β, meta loss function $\mathcal{L}_{meta}$
Output: Synthetic dataset $\mathcal{D}_s$,
 1: Train networks on full dataset $\mathcal{D}_r$ and record the average per-sample empirical loss $\ell(x_r)$, $\forall x_r \in \mathcal{D}_r$ with Eq. (2),
 2: $\mathcal{D}'_r \leftarrow \{x_r \mid \ell(x_r) < \tau, x_r \in \mathcal{D}_r\}$, $s.t.$ $|\mathcal{D}'_r| = (1-\alpha)|\mathcal{D}_r|$ (**preemptive pruning**),
 3: Initialize synthetic dataset $\mathcal{D}_s$,
 4: Initialize running mean $\hat{\mu}$ and variance $\hat{\sigma}^2$ of ITE,
 5: **repeat**
 6:     Sample a mini-batch $\mathcal{B}_r$ from real dataset,
 7:     Compute meta loss $\mathcal{L}_{meta}(\mathcal{B}_r, \mathcal{D}_s)$,
 8:     Compute meta loss after causal intervention $\mathcal{L}_{meta}(\mathcal{B}_r \setminus \{x_r\}, \mathcal{D}_s)$, $\forall x_r \in \mathcal{B}_r$,
 9:     Compute ITE values of real data $ITE(x_r)$, $\forall x_r \in \mathcal{B}_r$,
10:     Update running stats $\hat{\mu}$, $\hat{\sigma}^2$ with Eq. (8),
11:     $\mathcal{B}'_r \leftarrow \{x_r \mid z_{\frac{\beta}{2}} < ITE(x_r) < z_{1-\frac{\beta}{2}}, z \sim \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)\}$ (**adaptive pruning**),
12:     Update synthetic data: $\mathcal{D}_s \leftarrow \mathcal{D}_s - \frac{\partial}{\partial \mathcal{D}_s}\mathcal{L}_{meta}(\mathcal{B}'_r, \mathcal{D}_s)$.
13: **until** convergence

The three optimizations could reduce the ITE computation to $1/60{,}000$ of the original time. Overall, our method only increases the training time by 7%.

4.3   Bi-level Data Pruning for Dataset Distillation

In the analysis and discussion above, we propose two levels of data selection: the *preemptive* pruning by sample-wise empirical loss which is applied *before* the distillation process, and *adaptive* pruning by causal effect metric which is used *during* the distillation. We combine the two levels to propose a plug-and-play data pruning algorithm whose pseudo-code is shown in Algorithm 1.

5   Experiments

5.1   Datasets and Metrics

In this work, the experiments are conducted on common datasets for dataset distillation task, including CIFAR10 [17] (60,000 32x32 images in 10 classes), CIFAR100 [17] (60,000 32x32 images in 100 classes), SVHN [31] (over 99,000 32×32 images in 10 classes) and TinyImageNet [18] (100,000 64x64 images in 200 classes). We report the top-1 classification accuracy.

5.2   Implementation Details

We apply BiLP on gradient-matching based SOTAs DC [53] and IDC [16], so $g(x)$ in Eq. (7) indicates the per-sample gradient of the classification task. The multi-formation factor is 2 by default to fairly compare to the IDC baseline, and we also

Table 4: Dataset distillation performance of state-of-the-art and the proposed BiLP.

Method	CIFAR10 [17]			CIFAR100 [17]		SVHN [31]		
IPC	1	10	50	1	10	1	10	50
Full Dataset	84.8±0.1			56.2±0.3		95.4±0.1		
Random	14.4±2.0	26.0±1.2	43.4±1.0	4.2±0.3	14.6±0.5	14.6±1.6	35.1±4.1	70.9±0.9
Herding	21.5±1.2	31.6±0.7	40.4±0.6	8.4±0.3	17.3±0.3	20.9±1.3	50.5±3.3	72.6±0.8
DC [53]	28.3±0.5	44.9±0.5	53.9±0.5	12.8±0.3	25.2±0.3	31.2±1.4	76.1±0.6	82.3±0.3
KIP [32]	49.9±0.2	62.7±0.3	68.6±0.2	15.7±0.2	28.3±0.1	57.3±0.1	75.0±0.1	80.5±0.1
MTT [3]	46.3±0.8	65.6±0.7	71.6±0.2	24.3±0.3	40.1±0.4	-	-	-
FRePo [55]	46.8±0.7	65.5±0.4	71.7±0.2	28.7±0.1	42.5±0.2	-	-	-
HaBa [23]	48.3±0.8	69.9±0.4	74.0±0.2	33.4±0.4	40.2±0.2	69.8±1.3	83.2±0.4	88.3±0.1
IDC [16]	50.6±0.4	67.5±0.5	74.5±0.1	-	45.1±0.4	68.5±0.9	87.5±0.3	90.1±0.1
RFAD-NN [26]	53.6±1.2	66.3±0.5	71.1±0.4	26.3±1.1	33.0±0.3	52.2±2.2	74.9±0.4	80.9±0.3
IDM [54]	45.6±0.7	58.6±0.1	67.5±0.1	20.1±0.3	45.1±0.1	-	-	-
Zhang *et al.* [50]	49.2	67.1	73.8	29.8	45.6	-	-	-
DREAM [24]	51.1±0.3	69.4±0.4	74.8±0.1	29.5±0.3	46.8±0.7	69.8±0.8	87.9±0.4	90.5±0.1
PDD [5]	-	67.9±0.2	76.5±0.4	-	45.8±0.5	-	-	-
BiLP+DC	30.5±0.3	45.2±0.4	54.9±0.3	13.7±0.7	26.0±0.5	32.2±0.3	76.4±0.5	82.8±0.6
BiLP+IDC	51.5±0.3	69.4±0.5	75.4±0.2	30.1±0.4	47.2±0.6	70.3±0.6	88.3±0.1	90.8±0.4
BiLP+IDC (x3)	**55.9±0.5**	**69.8±1.1**	**76.9±0.9**	**34.0±0.7**	**48.0±1.0**	**77.2±0.6**	**88.7±0.4**	**91.0±0.7**

show the results with factor 3 (BiLP+IDC(x3) in Tab. 4). All the experiments including the efficiency analysis are conducted on one single RTX4090. For more details please refer to the supplementary.

5.3   Results

We compare our BiLP to various baselines in Tab. 4. Our selection strategy could notably enhance the current distillation algorithms. On average, BiLP consistently enhances DC and IDC by 0.8% and 1.2% (BiLP+DC and BiLP+IDC), especially on more diversified dataset CIFAR100. Moreover, with a larger multi-formation factor (BiLP+IDC x3), our method could surpass most of the state-of-the-art. Note that our method is also efficient due to the preemptive pruning, *e.g.*, BiLP on CIFAR100 only required 50% samples of the real dataset. This experiment shows the feasibility of embedding the data selection mechanism into the current distillation paradigm to boost performance and enhance efficiency, especially the preemptive pruning before the distillation.

5.4   Ablation Study

Pruning Criteria We analyze the impact of different pruning criteria in Table 5, involving the preemptive pruning by empirical loss and adaptive pruning with causal effects in two directions (prune the samples with large ITE or small ITE). The results indicate that across all datasets and IPC settings, the full BiLP consistently achieves the highest accuracy. In addition, both preemptive

Table 5: Ablation study on the pruning criteria on CIFAR10 and CIFAR100.

Preemptive Pruning	✗	✓	✓	✓	✓
Prune Small ITE	✗	✗	✗	✓	✓
Prune Large ITE	✗	✗	✓	✗	✓
CIFAR10, IPC=1	$50.6_{\pm0.4}$	$51.0_{\pm0.3}$	$51.2_{\pm0.3}$	$51.0_{\pm0.3}$	$\mathbf{51.3_{\pm0.3}}$
CIFAR10, IPC=10	$67.5_{\pm0.5}$	$68.5_{\pm0.1}$	$68.9_{\pm0.5}$	$68.7_{\pm0.4}$	$\mathbf{69.2_{\pm0.1}}$
CIFAR100, IPC=1	$28.2_{\pm0.6}$	$29.1_{\pm0.8}$	$29.8_{\pm0.2}$	$29.4_{\pm0.2}$	$\mathbf{30.1_{\pm0.4}}$

Table 6: Comparison of training time with ITE computation methods. We report the training seconds per iteration (s/iter) on CIFAR10 and IPC=10.

Running Stats Estimation	✗	✓	✓	✓
Taylor Approximation	✗	✗	✓	✓
Lazy Selection	✗	✗	✗	✓
ITE Computation (s/iter)	1799.22	23.03	0.27	**0.03**
Total Iteration (s/iter)	1799.53	23.38	0.65	**0.47**

and adaptive pruning brings performance gain. Interestingly, pruning large ITE values demonstrates a slightly better performance than pruning small ITE values, achieving the highest accuracy of 69.2%. This could imply that removing larger ITE may be more beneficial in scenarios where the synthetic data's complexity is limited. But overall, applying pruning of both large and small ITE would outperform any pruning in a single direction.

Time Complexity Since the computation time of ITE would pose a major bottleneck to the algorithm efficiency of BiLP, we offer a comparative analysis of the training time per iteration for different optimization levels in Table 6. The experiment is conducted on the CIFAR10 dataset with IPC=10 on IDC [16]. Without any optimization, the computation of ITE takes 5,800x more time than the rest steps of the distillation due to its computational intensity on the multiple meta gradients. The running estimation of global ITE distribution would help the mini-batch-based optimization. The Taylor approximation could effectively reduce the computation time by 98.8%. With the lazy selection (update the data per 10 iterations like [24]), our full optimization techniques could reduce the training overhead to ignorable 7% of the original algorithm.

Computation of Empirical Loss Criterion For the preemptive pruning, we train the networks on the full dataset and take the per-sample loss after convergence as the empirical loss criterion. However, we find that the loss values in very early classification epochs are also informative enough for data selection, probably as the early dynamics of samples can reflect their training difficulty and importance. Moreover, the number of trials to obtain loss values has little influence on the selection performance.
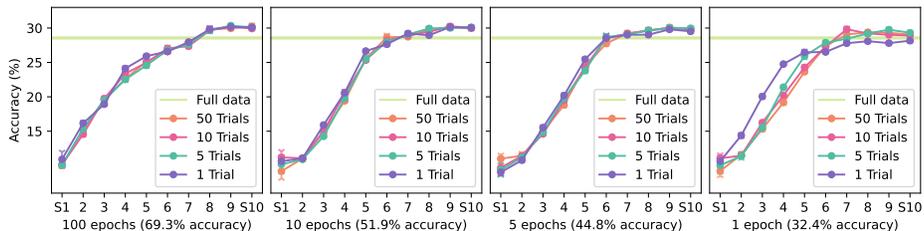
Fig. 3: Stratified distillation performance of different classification trials and epochs numbers. An ideal stratification yields a monotonously increasing performance curve.

We use a stratified experiment to examine the two factors: we sort the samples from large to small by the loss value and stratify them into 10 partitions S1~S10, where S1 contains the samples with the largest loss values. We take the average loss curves of 1, 5, 10, and 50 trials of classification, and take the mean loss value for the first 1, 5, 10, and 100 epochs when the training converges at around 100 epochs. As shown in Fig. 3 by the empirical results with DC [53] on CIFAR10, IPC=1, most loss values produce a good stratification and could distinguish the important samples unless only train the network for 1 epoch. Therefore we argue that **early epoch loss in very few trials is also accurate for selection**, which can be utilized with reduced computational burden of generating loss values, and can also be ignored or naturally embedded in the distillation process itself, extending our paradigm to broader applications.

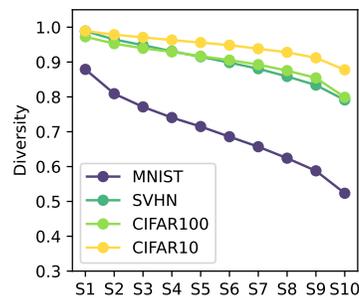### 5.5   Extended Discussion and Limitation

Efficient Distillation of Large-scale Datasets Our data utility paradigm can be efficiently extended to larger-scale and more heterogeneous datasets. We apply the data utility selection to the distillation of ImageNet-1K [7] and scale up to IPC=50, and also the large-scale video dataset Kinetics-400 [1] (detailed in supplementary). The results are listed in Tab. 7. Most methods struggle with high IPC due to demanding GRAM, except DM which allows class-separate training. MTT is extremely expensive for large-scale data due to its expert training. Our BiLP could mitigate training costs by its preemptive pruning, which significantly reduces the training time by at most 60% while maintaining or enhancing the performance. It is especially suitable for large-scale scenarios when the size increases, whose signal-to-noise ratio continues to decrease.

Higher-order Interaction of Data Utility Both our proposed preemptive and adaptive pruning is based on per-sample criteria, *i.e.* we assume each sample independently contributes to the distillation process. However, high-order information and interactions between samples may exist, and they have more complex causal mechanisms to the data synthesizing than the individual ITE in our method. In some extreme scenarios, these interactions are not negligible. For

Table 7: Dataset distillation on large-scale image and video datasets (*est.*: estimated).

Dataset	IPC	Algorithm	Full Data		BiLP		Random Real
			Accuracy	Training Time	Accuracy	Training Time	
ImageNet-1K [7] (1.3M images)	1	DC [53]	1.79±0.04	23.6h	**2.02±0.10**	**17.3h**	0.43±0.02
		DM [52]	1.58±0.11	22.4h	**1.95±0.12**	**9.9h**	
		MTT [3]	-	205.0h (*est.*)	**2.10±0.08**	**31.2h**	
	10	DM [52]	3.86±0.16	24.7h	**5.21±0.11**	**15.3h**	1.57±0.21
	50	DM [52]	8.22±0.86	35.0h	**9.41±0.38**	**20.6h**	5.29±0.70
Kinetics-400 [1] (300K videos)	1	DM [52]	2.78±0.14	37.3h	**2.92±0.15**	**29.6h**	0.90±0.23
		MTT [3]	-	460.8h (*est.*)	**2.77±0.21**	**76.8h**	
	10	DM [52]	9.48±0.15	43.5h	**9.70±0.12**	**32.2h**	3.33±0.43

example, *data diversity* is a higher-order data utility indicator since it is a property of the population rather than an individual. We observe the data selection paradigm always performs poorly on low diversity datasets like MNIST, which is due to the *diversity vanishing* in the small loss samples. We conduct similar stratified experiments as Sec. 5.4 and use the quotient of intraclass variance and interclass variance as the diversity metric (a large value indicates larger diversity). As shown in Fig. 4, only MNIST severely drops the diversity for the subgroups with a small loss. It can be challenging to incorporate sample interactions like diversity into consideration without

Fig. 4: The diversity of each stratified subgroup. MNIST suffers from diversity vanishing.

sacrificing efficiency (*e.g.* use annealing or Monte-Carlo algorithms), so we leave it to future work. In-depth discussion and modeling of high-order data utility involve complex systems which is beyond our scope. It is worth noting that the impact of diversity vanishing can be negligible in most realistic scenarios (*e.g.*, CIFAR), especially for large-scale datasets due to their large overall diversity. We provide the stratified visualization in the supplementary material.

6   Conclusion

This paper introduces a novel bi-level data pruning approach for efficient dataset distillation, which leverages the inherent redundancy in large datasets. We propose preemptive pruning from the dynamics of distillation, which is applied before the distillation, accompanied by further adaptive pruning based on causal effects. Our method would consistently enhance the performance of distillation algorithms while reducing the computational burden. We believe that our findings will inspire future research to explore more sophisticated data utility models and optimization techniques, ultimately leading to more efficient and effective dataset distillation methods.

Acknowledgments

References

1. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017)
2. Castro, F.M., Marin-Jimenez, M.J., Guil, N., Schmid, C., Alahari, K.: End-to-end incremental learning. In: ECCV (2018)
3. Cazenavette, G., Wang, T., Torralba, A., Efros, A.A., Zhu, J.Y.: Dataset distillation by matching training trajectories. In: CVPR (2022)
4. Cazenavette, G., Wang, T., Torralba, A., Efros, A.A., Zhu, J.Y.: Generalizing dataset distillation via deep generative prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3739–3748 (2023)
5. Chen, X., Yang, Y., Wang, Z., Mirzasoleiman, B.: Data distillation can be like vodka: Distilling more times for better quality. arXiv preprint arXiv:2310.06982 (2023)
6. Cui, J., Wang, R., Si, S., Hsieh, C.J.: Scaling up dataset distillation to imagenet-1k with constant memory. arXiv preprint arXiv:2211.10586 (2022)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
8. Deng, Z., Russakovsky, O.: Remember the past: Distilling datasets into addressable memories for neural networks. arXiv preprint arXiv:2206.02916 (2022)
9. Ding, J., Liu, Z., Zheng, G., Jin, H., Kong, L.: CondTSF: One-line plugin of dataset condensation for time series forecasting. arXiv preprint arXiv:2406.02131 (2024)
10. He, Y., Xiao, L., Zhou, J.T.: You only condense once: Two rules for pruning condensed datasets. Advances in Neural Information Processing Systems **36** (2024)
11. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. pmlr (2015)
12. Jacot, A., Gabriel, F., Hongler, C.: Neural tangent kernel: Convergence and generalization in neural networks. Advances in neural information processing systems **31** (2018)
13. Jin, W., Tang, X., Jiang, H., Li, Z., Zhang, D., Tang, J., Ying, B.: Condensing graphs via one-step gradient matching. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) (2022)
14. Jin, W., Zhao, L., Zhang, S., Liu, Y., Tang, J., Shah, N.: Graph condensation for graph neural networks. In: Proceedings of the International Conference on Learning Representations (ICLR) (2022)
15. Kim, B., Choi, J., Lee, S., Lee, Y., Ha, J.W., Lee, J.: On divergence measures for bayesian pseudocoresets. arXiv preprint arXiv:2210.06205 (2022)
16. Kim, J.H., Kim, J., Oh, S.J., Yun, S., Song, H., Jeong, J., Ha, J.W., Song, H.O.: Dataset condensation via efficient synthetic-data parameterization. In: ICML (2022)
17. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)

18. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N **7**(7), 3 (2015)
19. Lee, H.B., Lee, D.B., Hwang, S.J.: Dataset condensation with latent space knowledge factorization and sharing. arXiv preprint arXiv:2208.10494 (2022)
20. Li, G., Togo, R., Ogawa, T., Haseyama, M.: Soft-label anonymous gastric x-ray image distillation. In: Proceedings of the IEEE International Conference on Image Processing (ICIP). pp. 305–309 (2020)
21. Li, G., Togo, R., Ogawa, T., Haseyama, M.: Compressed gastric image generation based on soft-label dataset distillation for medical data sharing. Computer Methods and Programs in Biomedicine **227**, 107189 (2022)
22. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597 (2023)
23. Liu, S., Wang, K., Yang, X., Ye, J., Wang, X.: Dataset distillation via factorization. arXiv preprint arXiv:2210.16774 (2022)
24. Liu, Y., Gu, J., Wang, K., Zhu, Z., Jiang, W., You, Y.: Dream: Efficient dataset distillation by representative matching. arXiv preprint arXiv:2302.14416 (2023)
25. Liu, Z., Hao, K., Zheng, G., Yu, Y.: Dataset condensation for time series classification via dual domain matching. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) (2024)
26. Loo, N., Hasani, R., Amini, A., Rus, D.: Efficient dataset distillation using random feature approximation. arXiv preprint arXiv:2210.12067 (2022)
27. Loo, N., Hasani, R., Lechner, M., Rus, D.: Dataset distillation with convexified implicit gradients. arXiv preprint arXiv:2302.06755 (2023)
28. Maekawa, A., Kobayashi, N., Funakoshi, K., Okumura, M.: Dataset distillation with attention labels for fine-tuning bert. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL). pp. 119–127 (2023)
29. Maekawa, A., Kosugi, S., Funakoshi, K., Okumura, M.: DiLM: Distilling dataset into language model for text-level dataset distillation. In: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) (2024)
30. Manousakas, D., Xu, Z., Mascolo, C., Campbell, T.: Bayesian pseudocoresets. In: NeurIPS (2020)
31. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)
32. Nguyen, T., Chen, Z., Lee, J.: Dataset meta-learning from kernel ridge-regression. arXiv preprint arXiv:2011.00050 (2020)
33. Nguyen, T., Novak, R., Xiao, L., Lee, J.: Dataset distillation with infinitely wide convolutional networks. In: NeurIPS (2021)
34. Rubin, D.B.: Causal inference using potential outcomes: Design, modeling, decisions. Journal of the American Statistical Association **100**(469), 322–331 (2005)
35. Sajedi, A., Khaki, S., Amjadian, E., Liu, L.Z., Lawryshyn, Y.A., Plataniotis, K.N.: Datadam: Efficient dataset distillation with attention matching. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17097–17107 (2023)
36. Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. arXiv preprint arXiv:2111.02114 (2021)
37. Sener, O., Savarese, S.: A geometric approach to active learning for convolutional neural networks. ArXiv **abs/1708.00489** (2017)

38. Shin, D., Shin, S., Moon, I.C.: Frequency domain-based dataset distillation. arXiv preprint arXiv:2311.08819 (2023)
39. Shin, S., Bae, H., Shin, D., Joo, W., Moon, I.C.: Loss-curvature matching for dataset selection and condensation. In: International Conference on Artificial Intelligence and Statistics. pp. 8606–8628. PMLR (2023)
40. Tiwary, P., Shubham, K., Kashyap, V., et al.: Constructing bayesian pseudo-coresets using contrastive divergence. arXiv preprint arXiv:2303.11278 (2023)
41. Toneva, M., Sordoni, A., des Combes, R.T., Trischler, A., Bengio, Y., Gordon, G.J.: An empirical study of example forgetting during deep neural network learning. ArXiv **abs/1812.05159** (2018)
42. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
43. Wang, K., Zhao, B., Peng, X., Zhu, Z., Yang, S., Wang, S., Huang, G., Bilen, H., Wang, X., You, Y.: Cafe: Learning to condense dataset by aligning features. In: CVPR (2022)
44. Wang, T., Zhu, J.Y., Torralba, A., Efros, A.A.: Dataset distillation. arXiv preprint arXiv:1811.10959 (2018)
45. Wang, Z., Xu, Y., Lu, C., Li, Y.L.: Dancing with still images: Video distillation via static-dynamic disentanglement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6296–6304 (2024)
46. Wu, X., Zhang, B., Deng, Z., Russakovsky, O.: Vision-language dataset distillation. arXiv preprint arXiv:2308.07545 (2023)
47. Xu, Y., Lin, Z., Qiu, Y., Lu, C., Li, Y.L.: Low-rank similarity mining for multimodal dataset distillation. In: Proceedings of the International Conference on Machine Learning (ICML) (2024)
48. Yang, S., Xie, Z., Peng, H., Xu, M., Sun, M., Li, P.: Dataset pruning: Reducing training data by examining generalization influence. ArXiv **abs/2205.09329** (2022)
49. Yin, Z., Xing, E., Shen, Z.: Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. arXiv preprint arXiv:2306.13092 (2023)
50. Zhang, L., Zhang, J., Lei, B., Mukherjee, S., Pan, X., Zhao, B., Ding, C., Li, Y., Xu, D.: Accelerating dataset distillation via model augmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11950–11959 (2023)
51. Zhao, B., Bilen, H.: Dataset condensation with differentiable siamese augmentation. In: ICML (2021)
52. Zhao, B., Bilen, H.: Dataset condensation with distribution matching. In: WACV (2023)
53. Zhao, B., Mopuri, K.R., Bilen, H.: Dataset condensation with gradient matching. arXiv preprint arXiv:2006.05929 (2020)
54. Zhao, G., Li, G., Qin, Y., Yu, Y.: Improved distribution matching for dataset condensation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7856–7865 (2023)
55. Zhou, Y., Nezhadarya, E., Ba, J.: Dataset distillation using neural feature regression. arXiv preprint arXiv:2206.00719 (2022)