Not All Samples Should Be Utilized Equally: Towards Understanding and Improving Dataset Distillation

Shaobo Wang[1,2] Yantai Yang[2] Qilong Wang[2] Kaixin Li[3]
Linfeng Zhang[1,2] Junchi Yan[1*]
[1]School of Artificial Intelligence, Shanghai Jiao Tong University
[2]EPIC Lab, Shanghai Jiao Tong University
[3]National University of Singapore

{shaobowang1009,yanjunchi}@sjtu.edu.cn

Abstract

Dataset Distillation (DD) aims to synthesize a small dataset capable of performing comparably to the original dataset. Despite the success of numerous DD methods, theoretical exploration of this area remains unaddressed. In this paper, we take an initial step towards understanding various matching-based DD methods from the perspective of sample difficulty. We begin by empirically examining sample difficulty, measured by gradient norm, and observe that different matching-based methods roughly correspond to specific difficulty tendencies. We then extend the neural scaling laws of data pruning to DD to theoretically explain these matching-based methods. Our findings suggest that prioritizing the synthesis of easier samples from the original dataset can enhance the quality of distilled datasets, especially in low IPC (image-per-class) settings. Based on our empirical observations and theoretical analysis, we introduce the Sample Difficulty Correction (SDC) approach, designed to predominantly generate easier samples to achieve higher dataset quality. Our SDC can be seamlessly integrated into existing methods as a plugin with minimal code adjustments. Experimental results demonstrate that adding SDC generates higher-quality distilled datasets across 7 distillation methods and 6 datasets.

1. Introduction

In an era of data-centric AI, scaling laws [17] have shifted the focus to data quality. Under this scenario, dataset distillation (DD) [36, 43–45] has emerged as a solution for creating high-quality data summaries. Unlike data pruning methods [1, 7, 13, 41, 42, 49] that directly select data points from original datasets, DD methods are designed to generate novel data points through learning. The utility of DD methods has been witnessed in fields such as privacy protection [4, 6, 11, 26], continual learning [14, 29, 35, 50], and neural architecture search [2, 31, 39]

Among the various DD techniques, matching-based methods, particularly gradient matching (GM) [18, 23, 51, 52] and trajectory matching (TM) [3, 8, 12, 15], have demonstrated outstanding performance. However, a gap remains between their theoretical understanding and empirical success. To offer a unified explanation of these methods, we aim to explore the following question:

Question 1: *Is there a unified theory to explain existing matching-based DD methods?*

To address Question 1, we first empirically examine the differences between matching-based distillation methods. It is widely acknowledged that sample difficulty (Definition 1) is a crucial metric in data-centric AI that significantly affects model performance, as seen in dataset pruning [27, 28, 38, 40], and large language model prediction [9, 24, 25]. To track the differences between current distillation methods, we follow [34] and analyze sample difficulty using the GraDN metric (Definition 2). Surprisingly, we discover that the GraDN score is increased in GM-based methods (Figure 1(a)), while TM-based methods may reduce this metric (Figure 1(c)). These distinct trends indicate that the difficulty of samples utilized in GM-based methods is elevated (Figure 2(a)), whereas in TM-based methods, it is reduced (Figure 2(b)) during the distillation process.

Motivated by these observations, we develop a theoretical explanation for current DD methods from the perspective of sample difficulty. Specifically, we draw upon the *neural scaling law* in the data pruning theory [38] to connect sample difficulty with performance. As shown in Figure 4(c), our theory indicates that in matching-based DD

*Corresponding Author.

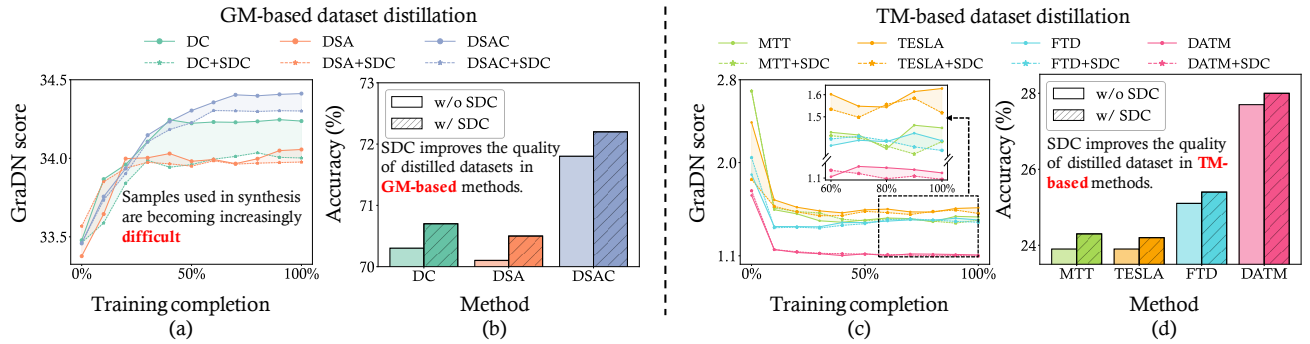

Figure 1. We conducted experiments of GM-based methods on the FashionMNIST dataset and TM-based methods on the CIFAR-100 dataset. (a) Average gradient norms of network parameters for different GM-based methods are enhanced during distillation. The shade represents the gap before and after adding our SDC. (b) Test performance w/ and w/o our proposed SDC on different GM-based methods. Our SDC are incorporated to improve all matching-based methods. (c) Average gradient norms of network parameters for different TM-based methods are alleviated during distillation. (d) Test performance w/ and w/o our proposed SDC on different TM-based methods. Note that the average gradient norms are smoothed using the exponential moving average. Best viewed in color.

methods, when the synthetic dataset is small—specifically, when the images-per-class (IPC) is low—the optimal strategy is to primarily focus on easier samples rather than harder ones to enhance performance. Based on our theory, we further explain why TM-based methods usually outperform GM-based methods in real scenarios.

Beyond developing a theoretical framework, we take steps to explore solutions for improving current approaches. This raises another key research question:

Question 2: *Is it empirically feasible to identify a loss function that surpasses the performance of the matching loss by controlling the difficulty of learned patterns during distillation?*

To address Question 2, based on our empirical observations and theoretical analysis, we propose the novel ***S**ample **D**ifficulty **C**orrection* (SDC) method to improve the synthetic dataset quality in current matching-based distillation methods. We do this by guiding the distillation method to focus more on `easy` samples than `hard` samples, adding an implicit gradient norm regularizer to enhance quality.

Our contributions are listed as follows:

- We empirically investigate *sample difficulty* from the perspective of gradient norm in distillation methods, linking it to synthetic dataset quality. We propose that GM-based methods focus on difficult samples during optimization, while TM-based methods show no dominant preference for difficulty. This may explain the poorer performance of GM-based methods than TM-based methods.
- We theoretically elucidate the mechanism of matching-based DD methods from the perspective of sample difficulty. Adapting the neural scaling law theory from data pruning [38] to distillation settings, we provide insights into how matching strategies evolve with the size of the synthetic dataset. Consequently, we propose that focusing on matching `easy` samples is a better strategy when the synthetic dataset is small.
- We introduce *Sample Difficulty Correction* (SDC) to improve the quality of synthetic datasets in current matching-based DD methods. Our method demonstrates superior generalization performance across 7 distillation methods (DC [52], DSA [51], DSAC [23], MTT [3], FTD [12], TESLA [8], DATM [15]) and 6 datasets (MNIST [10], FashionMNIST [47], SVHN [33], CIFAR-10/100 [19], and Tiny-ImageNet [21]).

2. Preliminaries and Related Work

Dataset distillation involves synthesizing a small, condensed dataset $\mathcal{D}_{\text{syn}}$ that efficiently encapsulates the informational essence of a larger, authentic dataset $\mathcal{D}_{\text{real}}$.

Gradient Matching (GM) based methods are pivotal in achieving distillation by ensuring the alignment of training gradients between surrogate models trained on both the original dataset $\mathcal{D}_{\text{real}}$ and the synthesized dataset $\mathcal{D}_{\text{syn}}$. This method is first introduced by DC [52]. Let θ_t represent the network parameters sampled from distribution P_θ at step t, and C symbolizes the categories within $\mathcal{D}_{\text{real}}$. The cross-entropy loss $\mathcal{L}$, is employed to assess the matching loss by comparing the gradient alignment over a time horizon of T steps. The formal optimization objective of DC is:

$$\arg\min_{\mathcal{D}_{\text{syn}}} \mathop{\mathbb{E}}_{\theta_0 \sim P_\theta, c \sim C} \left[\sum_{t=0}^{T} \mathbf{D}\left(\nabla_\theta \mathcal{L}_{\mathcal{D}^c_{\text{real}}}(\theta_t), \nabla_\theta \mathcal{L}_{\mathcal{D}^c_{\text{syn}}}(\theta_t) \right) \right], \tag{1}$$

where $\mathbf{D}$ measures the cumulative distances (*e.g.*, cosine/L_2 distance in DC) between the gradients of weights corresponding to each category output. The parameter updates for θ are executed in an inner loop via gradient descent, with a specified learning rate η:

$$\theta_{t+1} \leftarrow \theta_t - \eta \cdot \nabla_\theta \mathcal{L}_{\mathcal{D}_{\text{syn}}}(\theta_t). \tag{2}$$

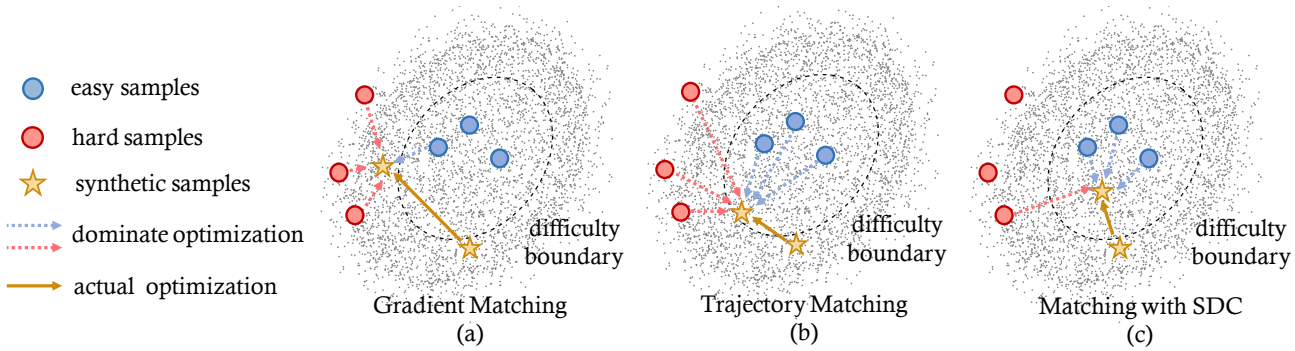

Figure 2. Comparison of matching-based dataset distillation methods from a sample difficulty perspective. (a) Gradient matching-based methods mainly utilize `hard` samples during synthesizing. (b) Trajectory matching-based methods do not explicitly take the difficulty of samples into consideration. (c) Our SDC encourages matching-based methods to prioritize the synthesis of `easy` samples.

Building upon this, DSA [51] enhances DC by implementing consistent image augmentations on both $\mathcal{D}_{\text{real}}$ and $\mathcal{D}_{\text{syn}}$ throughout the optimization process. Moreover, DCC [23] refines the gradient matching objective by incorporating class contrastive signals at each gradient matching step, which results in enhanced stability and performance. Combining DSA and DCC, DSAC [23] further introduces improvements by synergizing these techniques. The revised optimization objective for DCC and DSAC is formulated as:

$$\arg\min_{\mathcal{D}_{\text{syn}}} \mathbb{E}_{\theta_0 \sim P_\theta} \left[\sum_{t=0}^{T} \mathbf{D} \left(\mathbb{E}_{c \in C} \left[\nabla_\theta \mathcal{L}_{\mathcal{D}^c_{\text{real}}}(\theta_t) \right], \mathbb{E}_{c \in C} \left[\nabla_\theta \mathcal{L}_{\mathcal{D}^c_{\text{syn}}}(\theta_t) \right] \right) \right]. \tag{3}$$

Trajectory matching (TM) based approaches aim to match the training trajectories of surrogate models by optimizing over both the real dataset $\mathcal{D}_{\text{real}}$ and the synthesized dataset $\mathcal{D}_{\text{syn}}$. TM-based methods were initially proposed in MTT [3]. Let term $\tau^{\mathcal{D}_{\text{real}}}$ denote the expert training trajectories, represented as a sequential array of parameters $\{\theta_t^{\mathcal{D}_{\text{real}}}\}_{t=0}^{T}$, obtained from training a network on the real dataset $\mathcal{D}_{\text{real}}$. In parallel, $\theta_t^{\mathcal{D}_{\text{syn}}}$ refers to the parameter set of the network trained on $\mathcal{D}_{\text{syn}}$ at step t. In each iteration, parameters $\theta_t^{\mathcal{D}_{\text{real}}}$ and $\theta_{t+M}^{\mathcal{D}_{\text{real}}}$ are randomly selected from the expert trajectory pool $\{\tau^{\mathcal{D}_{\text{real}}}\}$, serving as the initial and target parameters for trajectory alignment, where M is a predetermined hyperparameter. TM-based methods enhance the synthetic dataset $\mathcal{D}_{\text{syn}}$ by minimizing the loss defined as:

$$\arg\min_{\mathcal{D}_{\text{syn}}} \mathbb{E}_{\theta_0 \sim P_\theta} \left[\sum_{t=0}^{T-M} \frac{\mathbf{D}\left(\theta_{t+M}^{\mathcal{D}_{\text{real}}}, \theta_{t+N}^{\mathcal{D}_{\text{syn}}}\right)}{\mathbf{D}\left(\theta_{t+M}^{\mathcal{D}_{\text{real}}}, \theta_{t}^{\mathcal{D}_{\text{real}}}\right)} \right], \tag{4}$$

where $\mathbf{D}$ is a distance metric (*e.g.*, L_2 distance in MTT) and $N << M$ is a predefined hyperparameter. $\theta_{t+N}^{\mathcal{D}_{\text{syn}}}$ is derived through an inner optimization using the cross-entropy loss $\mathcal{L}$ with the learning rate η:

$$\theta_{t+i+1}^{\mathcal{D}_{\text{syn}}} \leftarrow \theta_{t+i}^{\mathcal{D}_{\text{syn}}} - \eta \nabla_\theta \mathcal{L}_{\mathcal{D}_{\text{syn}}}(\theta_{t+i}^{\mathcal{D}_{\text{syn}}}), \text{ where } \theta_t^{\mathcal{D}_{\text{syn}}} := \theta_t^{\mathcal{D}_{\text{real}}}. \tag{5}$$

Similarly, TESLA [8] utilizes linear algebraic manipulations and soft labels to increase compression efficiency, FTD [12] aims to seek a flat trajectory to avoid accumulated trajectory error, and DATM [15] considers matching only necessary parts of trajectory with difficulty alignment.

3. Method

3.1. A Closer Look at Sample Difficulty

In this subsection, we aim to intuitively understand dataset distillation through the concept of sample difficulty (Definition 1), which is pivotal in data-centric AI [5, 9, 24, 27, 28, 40, 48]. We begin by empirically observing the evolution of sample difficulty during the distillation process. Firstly, we introduce the commonly used definition of sample difficulty, namely the GraDN score (Definition 2), and validate the reliability of this metric. Furthermore, we track the GraDN score across current dataset distillation methods to delve deeper into their underlying mechanisms.

Definition 1 (Sample Difficulty [30])**.** *Given a training pair (x, y) and a series of pretrained models at training time t, the sample difficulty, denoted $\chi(x, y; \Theta_t)$, is defined as the expected probability of (x, y) being misclassified by an ensemble of models $\theta_t \in \Theta_t$. Formally, it is presented as:*

$$\chi(x, y; \Theta_t) = \mathbb{E}_{\theta_t \in \Theta_t} \left[\mathbb{1}\left(y \neq \theta_t(x)\right) \right], \tag{6}$$

where $\mathbb{1}(z)$ is an indicator function that equals 1 if the boolean input z is true, and 0 otherwise. In this case, the indicator function equals to 1 if the sample (x, y) is misclassified by the model with parameters θ_t, and 0 otherwise.

Definition 2 (GraDN Score [34])**.** *Consider a training pair (x, y), with $\mathcal{L}$ representing the loss function. At time t, the GraDN score for (x, y) is calculated as the average gradient norm of the loss $\mathcal{L}$ across a diverse ensemble of models*

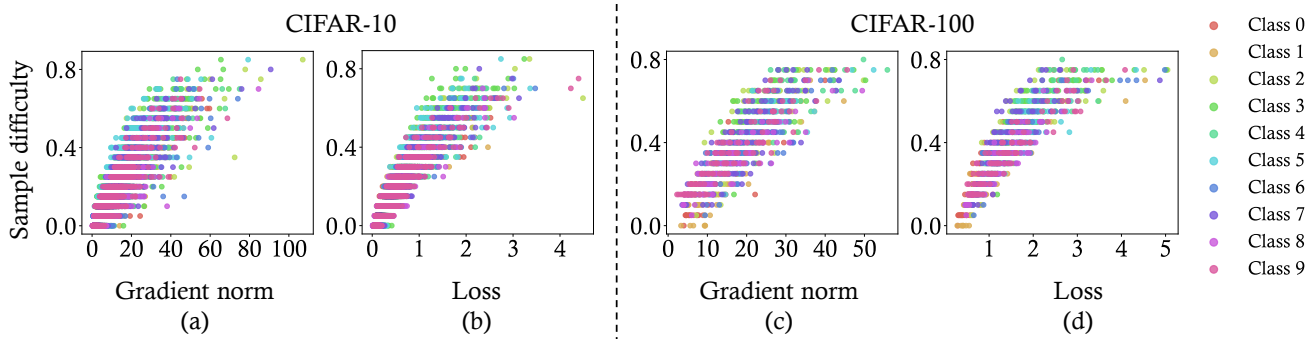

Figure 3. The statistical relationship between sample difficulty $\chi(x, y; \Theta_t)$, gradient norm $\text{GraDN}(x, y; \Theta_t)$, and average validation loss for each sample (x, y) on a series of models with $\theta_t \in \Theta_t$. We observe a significant positive correlation between sample difficulty and both the gradient norm and the loss. Experiments were conducted using ResNet-18 on CIFAR-10 and ResNet-34 on CIFAR-100. Each score was evaluated across 20 pretrained models. For CIFAR-100, 10 categories were randomly selected for visualization. The relationships depicted are: (a) sample difficulty vs. gradient norm on CIFAR-10, (b) sample difficulty vs. loss on CIFAR-10, (c) sample difficulty vs. gradient norm on CIFAR-100, and (d) sample difficulty vs. loss on CIFAR-100.

with parameters $\theta_t \in \Theta_t$:

$$\text{GraDN}(x, y; \Theta_t) = \mathbb{E}_{\theta_t \in \Theta_t} \left[\|\nabla_\theta \mathcal{L}(x, y; \theta_t)\|_2 \right], \quad (7)$$

where $\nabla_\theta \mathcal{L}(x, y; \theta_t)$ denotes the gradient of loss $\mathcal{L}$ on sample (x, y) w.r.t. *the model parameters θ_t, and $\|\cdot\|_2$ denotes L_2 norm.*

According to [30], the difficulty of each sample can be assessed by the misclassification ratio across a series of pretrained models (Definition 1). Additionally, from an optimization perspective, it can be represented by the gradient norm of the loss on a series of pretrained models for this sample (Definition 2). In our study, we adopt Definition 2 to evaluate sample difficulty as interpreted by various matching methods.

Empirical verification of the relationship between sample difficulty and gradient norm. We conducted experiments to verify the reliability of the GraDN score in classifying the CIFAR-10 and CIFAR-100 datasets by training a set of models. As depicted in Figure 3, the GraDN score shows a clear positive correlation with the sample difficulty. For easier samples, GraDN scores are generally lower, exerting minimal impact on the network's gradient flow. Conversely, for harder samples, higher GraDN scores indicate a significant impact on the optimization directions of the models. We show detailed results of the relationships between these metrics in Appendix 1.

Exploring sample difficulty across different distillation methods. Beyond sample difficulty under classification scenarios, we now extend our observations to matching-based distillation methods. Specifically, we examined the average gradient norm of the training cross-entropy loss across network parameters during the distillation process. As shown in Figure 1(a)(c), we found that the average gradient norm (corresponding to the GraDN score) tends to increase in GM-based methods (signifying harder samples), whereas it decreases in TM-based methods (indicating easier samples). This unexpected phenomenon motivates us to further theoretically explore matching-based distillation methods from the perspective of sample difficulty.

3.2. An Analytical Theory for Explaining Matching-based Dataset Distillation

In Section 3.1, we empirically observed distinct trends in *sample difficulty* across various dataset distillation methods. Here, we propose an analytical theory based on the *neural scaling law* to formally analyze sample difficulty in matching-based methods. We extend the theory of data pruning presented by [38] and validate its applicability within the context of DD using an expert-student perceptron model. Unlike data pruning, where the pruned dataset is directly selected from the original dataset, DD involves synthesizing a small, new, unseen dataset.

We start our analysis with tools from statistical mechanics [32]. Let us consider a classification problem in dataset $\mathcal{D}^{\text{real}}$ containing d_{real} samples $\{x_i, y_i\}_{i=1,\ldots,d_{\text{real}}}$, where $x_i \in \mathbb{R}^d \sim \mathcal{N}(0, I_d)$ are *i.i.d.* zero-mean, unit variance Gaussian inputs, and $y_i = \text{sign}(\theta^{\mathcal{D}_{\text{real}}\top} x_i) \in \{-1, +1\}$ are labels generated by an expert perceptron $\theta^{\mathcal{D}_{\text{real}}} \in \mathbb{R}^d$. Our analysis is within the high-dimensional statistics limit, where $d, d_{\text{real}} \to \infty$ while maintaining the ratio of total training samples to parameters $\alpha_{\text{tot}} = d_{\text{real}}/d$ at $O(1)$. The general distillation algorithm proceeds as follows:

1. Train a student perceptron on $\mathcal{D}_{\text{real}}$ for a few epochs to obtain weights θ^{probe}. The gap between can be measured by the angle γ between the probe student θ^{probe} and the expert θ^{real}. If $\theta^{\text{probe}} \approx \theta^{\text{real}}$, we denote the θ^{probe} as a *perfect* probe ($\gamma = 0$). Otherwise, in *imper-*

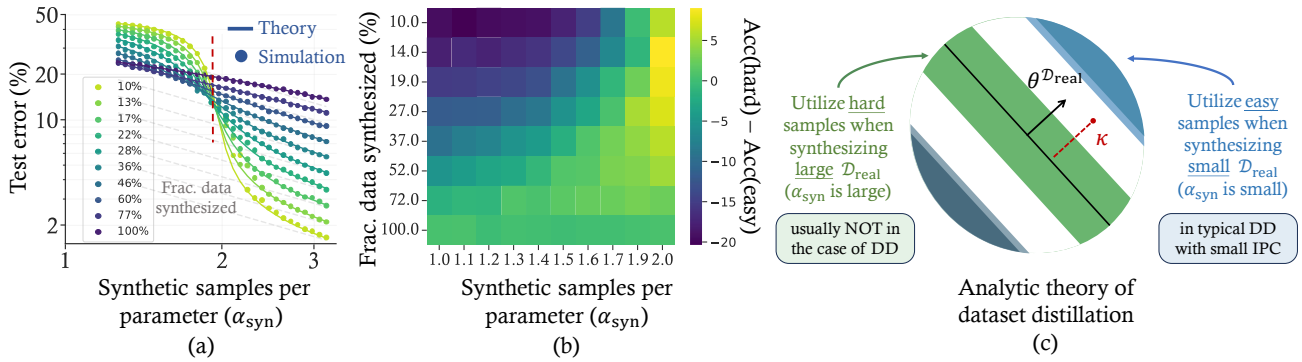

Figure 4. An analytic theory of dataset distillation. (a) Test error ε as a function of the synthetic samples per parameter α_{syn} and fraction of data synthesized f in a perfect expert setting ($\gamma = 0$). (b) We show the difference in test accuracy in learning synthetic dataset by learning from `hard` vs `easy` samples, revealing the change in distillation strategy. (c) Our theory suggests that when we synthesize small dataset (small α_{syn}), the better distillation strategy is to utilize the `easy` samples.

fect probe cases, $\gamma \neq 0$.

2. Compute the margin $m_i = \theta^{\text{probe}\top}(y_i x_i)$ for each training example, categorizing large (small) margins as `easy` (`hard`) samples.
3. Generate a synthetic dataset $\mathcal{D}_{\text{syn}}$ of size $d_{\text{syn}} = f d_{\text{real}}$, by learning from the hardest samples from $\mathcal{D}_{\text{real}}$ for few steps. The ratio of total samples of the synthetic dataset to parameters is $\alpha_{\text{syn}} = d_{\text{syn}}/d$.
4. Train a new perceptron $\theta^{\mathcal{D}_{\text{syn}}}$ on $\mathcal{D}_{\text{syn}}$, aiming to classify training data with the maximal margin $\kappa = \min_i \theta^{\mathcal{D}_{\text{syn}}\top} y_i x_i$.

We analyze the test error ε of this final perceptron $\theta^{\mathcal{D}_{\text{syn}}}$ as a function of α_{syn}, f, and the angle γ between the probe student θ^{probe} and the expert $\theta^{\mathcal{D}_{\text{real}}}$. Similar to [38], we define θ^{probe} as a random Gaussian vector conditioned to have angle γ with the expert. Under this scenario, we can derive an asymptotically exact analytic expression for $\varepsilon(\alpha_{\text{syn}}, f, \gamma)$ (see Appendix 4 for details):

$$\varepsilon = \frac{\cos^{-1}(R)}{\pi}, \quad \text{where } R = \frac{\theta^{\mathcal{D}_{\text{real}}\top}\theta^{\mathcal{D}_{\text{syn}}}}{\|\theta^{\mathcal{D}_{\text{real}}}\|_2 \|\theta^{\mathcal{D}_{\text{syn}}}\|_2} \tag{8}$$

Likewise in [38], we can solve R with saddle point equations in Appendix 4.1, enabling direct predictions of the test error ε according to Eq (8).

Verification of the neural scaling law in dataset distillation. We first evaluated the correctness of our theory in the perfect expert-student setting ($\gamma = 0$). As shown in Figure 4(a), we observed an excellent match between our analytic theory (solid curves) and numerical simulations (dots) of perceptron learning at parameters $d = 200$ in dataset distillation. We also verify our theory in imperfect probe settings when $\gamma \neq 0$, as shown in Appendix 4.2.

The neural scaling law for dataset distillation. We further investigate the relationship between the distillation ratio (α_{syn}), the fraction of data synthesized (f), and the final accuracy of $\theta^{\mathcal{D}_{\text{syn}}}$ under various distillation strategies, such as synthesizing $\mathcal{D}_{\text{syn}}$ from `hard` or `easy` samples. Similar to data pruning, when $f = 1$ (no distillation), the test error follows the classical perceptron learning power-law scaling, $\varepsilon \propto \alpha_{\text{syn}}^{-1}$. In other cases, our findings reveal that for smaller α_{syn} (smaller synthetic datasets), learning from the `hard` samples results in poorer performance than no distillation. Conversely, for larger α_{syn}, focusing on the `hard` samples yields substantially better outcomes than no distillation. We find that in limited data regimes, matching the `easy` samples, which have the largest margins, offers a more effective distillation strategy. This finding highlights that in most cases of DD (where $d_{\text{syn}} << d_{\text{real}}$), it is crucial for the model to first learn from the basic characteristics in $\mathcal{D}_{\text{real}}$; hence, prioritizing `easy` samples facilitates reaching a moderate error level more swiftly.

Understanding GM-based and TM-based methods with the neural scaling law. As depicted in Figure 1(a), we observe that GM-based methods typically incorporate the `hard` samples within the synthetic dataset. This trend is due to the GM-based matching loss in Eq.(1), which predominantly penalizes samples with large gradients (`hard` samples) (shown in Figure 2(b)). However, in most DD settings, the size of synthetic dataset $\mathcal{D}_{\text{syn}}$ is usually small. Therefore, according to our theory, we should mainly focus on synthesizing the dataset by matching `easy` samples to achieve higher dataset quality. In contrast, the simplicity of the synthetic samples generated by TM-based methods, as shown in Figure 1(a), is not directly concerned through distillation. From Eq.(4), it is evident that TM-based methods prioritize parameter alignment, thus penalizing the matching term without explicitly targeting sample difficulty (shown in Figure 2(c)). This approach results in a synthetic dataset that may be generated by learning samples of randomly vary in difficulty. We can provide a explanation that TM-based methods generalize well in real scenarios than GM-based methods because of they do not explicitly focus on synthesizing by matching `hard` samples.

3.3. Matching with Sample Difficulty Correction

Based on our theoretical analysis of matching-based dataset distillation, we propose a novel method to enhance existing techniques for synthesizing higher-quality distilled datasets. Although TM-based methods have achieved relative success on current benchmark datasets, they do not explicitly consider sample difficulty, which could ensure higher synthetic dataset quality.

A direct approach to impose constraints on sample difficulty is to calculate the gradient norm for each sample as a metric to determine its utility. Let us consider the case of GM-based methods. At step t, a batch of real samples $\mathcal{B}^c_{\text{real}} \sim \mathcal{D}^c_{\text{real}}$ of class $c \in C$ is to be matched with the gradients of a synthetic batch $\mathcal{B}^c_{\text{syn}} \sim \mathcal{D}^c_{\text{syn}}$. To decide whether to utilize each sample in $\mathcal{B}^c_{\text{real}}$, it is natural to compute the gradient norm of each sample and utilize those with a score smaller than a predefined threshold τ. Specifically, a sample (x, y) is utilized if $\|\nabla_\theta \mathcal{L}(x, y; \theta_t)\|_2 \leq \tau$. Consequently, the modified loss for matching only `easy` samples is:

$$\begin{aligned} \mathcal{L}_{\tilde{\mathcal{B}}^c_{\text{real}}} &= \mathbb{E}_{(x,y)\in\tilde{\mathcal{B}}^c_{\text{real}}} \left[\mathcal{L}(x, y; \theta_t)\right], \\ \mathcal{L}_{\tilde{\mathcal{B}}^c_{\text{syn}}} &= \mathbb{E}_{(x,y)\in\tilde{\mathcal{B}}^c_{\text{syn}}} \left[\mathcal{L}(x, y; \theta_t)\right], \end{aligned} \tag{9}$$

where $\tilde{\mathcal{B}}^c_{\text{real}} = \{(x, y) | (x, y) \in \mathcal{B}^c_{\text{real}}, \|\nabla_\theta \mathcal{L}(x, y; \theta_t)\|_2 \leq \tau\}$ denotes the modified batch with only `easy` samples, and $\tilde{\mathcal{B}}^c_{\text{syn}}$ denote a sampled batch from $\mathcal{D}^c_{\text{syn}}$ with the same size as $\tilde{\mathcal{B}}^c_{\text{real}}$. The corresponding matching loss should be:

$$\tilde{L}(\theta_t) = \mathbf{D}\left(\nabla_\theta \mathcal{L}_{\tilde{\mathcal{B}}^c_{\text{real}}}(\theta_t), \nabla_\theta \mathcal{L}_{\tilde{\mathcal{B}}^c_{\text{syn}}}(\theta_t)\right), \tag{10}$$

However, the computational cost of constructing reduced `easy` sample batch $\tilde{\mathcal{B}}^c_{\text{real}}$ from $\mathcal{B}^c_{\text{real}}$ is unrealistic in real-world scenarios because it requires calculating the gradient norm for each sample independently, resulting in a tenfold or greater increase in time. Besides, determining the difficulty threshold τ is also ad-hoc and challenging for each sample. Therefore, we take an alternative approach, *i.e.*, we consider adding the overall sample difficulty of the whole batch $\mathcal{B}^c_{\text{syn}}$ as an implicit regularization term in the matching loss function. Our proposed methods, named *Sample Difficulty Correction* (SDC), can be incorporated into current matching methods with minimal adjustment of code implementation. Specifically, for a single-step GM, we have the following modified loss:

$$\begin{aligned} L_\lambda(\theta_t) = &\underbrace{\mathbf{D}\left(\nabla_\theta \mathcal{L}_{\mathcal{B}^c_{\text{real}}}(\theta_t), \nabla_\theta \mathcal{L}_{\mathcal{B}^c_{\text{syn}}}(\theta_t)\right)}_{\text{Gradient Matching Loss}} \\ &+ \underbrace{\lambda \left\|\nabla_\theta \mathcal{L}_{\mathcal{B}^c_{\text{syn}}}\right\|_2}_{\text{Gradient Norm Regularization}} \end{aligned} \tag{11}$$

For TM-based methods that do not explicitly focus on sample difficulty during distillation, we compute the average gradient norm of the whole dataset $\mathcal{D}_{\text{syn}}$ during the optimization of the student network $\theta_t^{\mathcal{D}_{\text{syn}}}$ *w.r.t.* the training loss as the regularization term. Specifically, we have:

$$\begin{aligned} L_\lambda(\theta_t^{\mathcal{D}_{\text{syn}}}) = &\underbrace{\mathbf{D}\left(\theta_{t+M}^{\mathcal{D}_{\text{real}}}, \theta_{t+N}^{\mathcal{D}_{\text{syn}}}\right) / \mathbf{D}\left(\theta_{t+M}^{\mathcal{D}_{\text{real}}}, \theta_t^{\mathcal{D}_{\text{real}}}\right)}_{\text{Trajectory Matching Loss}} \\ &+ \underbrace{\lambda \left\|\nabla_\theta \mathcal{L}_{\mathcal{D}_{\text{syn}}}\right\|_2}_{\text{Gradient Norm Regularization}} \end{aligned} \tag{12}$$

By adding the gradient norm regularization, we can implicitly enforce current matching-based methods to mainly concentrate on synthesizing `easy` samples to achieve better synthetic data quality. We provide the algorithm pseudocodes for GM- and TM-based methods in Appendix 2.3.

4. Experiments

4.1. Basic Settings

Datasets and baselines. For GM-based methods, we followed previous works to conduct experiments on MNIST [10], FashionMNIST [47], SVHN [33] datasets. We utilized current GM-based methods, including DC [52], DSA [51], and DSAC [23] as baselines. For TM-based methods, we followed the recent papers to use CIFAR-10, CIFAR-100 [19], and Tiny ImageNet [21] datasets. We performed experiments on current baselines including MTT [3], FTD [12], TESLA [8], and DATM [15]. We added our *Sample Difficulty Correction* (SDC) for all these baseline methods. To ensure a fair comparison, we employed identical hyperparameters for GM-based and TM-based methods with and without SDC while keeping all other variables constant, such as model architecture and augmentations. As per convention, for TM-based methods, we used max test accuracy, while for GM-based methods, we utilized the test accuracy from the last iteration. We also compared our methods with classical data pruning algorithms including Random, Herding [46], and Forgetting [41]. All hyperparameters are detailed in Appendix 2.1.

Neural networks for distillation. We used ConvNet as default to conduct experiments. Consistent with other previous methods, we used 3-layer ConvNet for CIFAR-10, CIFAR-100, MNIST, SVHN, and FashionMNIST, and 4-layer ConvNet for Tiny ImageNet.

4.2. Main Results

GM-based methods on MNIST, FashionMNIST, and SVHN. As presented in Table 1, we report the results of three GM-based methods applied to MNIST, FashionMNIST, and SVHN datasets. Each method was evaluated with IPC (images-per-class) values of 1, 10, and 50. Notably, adding SDC improves the test accuracy of baseline methods across all datasets and IPC values, demonstrating

Table 1. Comparison of test accuracy (%) results of GM-based dataset distillation methods w/ and w/o SDC on MNIST, FashionMNIST, and SVHN datasets.

Dataset	MNIST			FashionMNIST			SVHN		
IPC	1	10	50	1	10	50	1	10	50
Ratio (%)	0.02	0.2	1	0.2	2	10	0.2	2	10
Random	64.9±3.5	95.1±0.9	97.9±0.2	51.4±3.8	73.8±0.7	82.5±0.7	14.6±1.6	35.1±4.1	70.9±0.9
Herding	89.2±1.6	93.7±0.3	94.8±0.2	67.0±1.9	71.1±0.7	71.9±0.8	20.9±1.3	50.5±3.3	72.6±0.8
Forgetting	35.5±5.6	68.1±3.3	88.2±1.2	42.0±5.5	53.9±2.0	55.0±1.1	12.1±1.7	16.8±1.2	27.2±1.5
DC	91.8±0.4	97.4±0.2	98.5±0.1	70.3±0.7	82.1±0.3	83.6±0.2	31.1±1.3	75.3±0.6	82.1±0.2
+SDC	92.0±0.4	97.5±0.1	98.9±0.1	70.7±0.5	82.4±0.3	84.7±0.2	31.4±1.2	76.0±0.5	82.3±0.3
DSA	88.9±0.8	97.2±0.1	99.1±0.1	70.1±0.4	84.7±0.2	88.7±0.2	29.4±1.0	79.2±0.4	84.3±0.4
+SDC	89.2±0.4	97.3±0.1	99.2±0.4	70.5±0.5	84.8±0.2	88.9±0.1	30.6±1.0	79.4±0.4	85.3±0.4
DSAC	89.2±0.7	97.7±0.1	98.8±0.1	71.8±0.7	84.9±0.2	88.5±0.2	47.5±1.8	80.1±0.5	87.3±0.2
+SDC	89.7±0.7	97.8±0.1	98.9±0.1	72.2±0.6	85.1±0.1	88.7±0.1	48.1±1.6	80.4±0.3	87.4±0.2
Whole Dataset	99.6±0.0			93.5±0.1			95.4±0.1		

the effectiveness of our approach. Notably, adding SDC to the original method improved the test accuracy of DSA by **1.2%** on the SVHN dataset with IPC = 1, and by **1%** with IPC = 50. For DC on the FashionMNIST dataset with IPC = 50, the test accuracy was increased by **1.1%** with SDC. All hyperparameters are detailed in Table 4.

TM-based methods on CIFAR-10/100 and Tiny ImageNet. As shown in Table 2, we present the results of four TM-based methods trained on CIFAR-10, CIFAR-100 and Tiny ImageNet. By incorporating the average gradient norm as a regularization term during matching with SDC, the resulting test accuracy was generally improved. Notably, employing SDC improved the test accuracy of FTD on CIFAR-10 by **1.2%** with IPC = 10 and **1.1%** with IPC = 50, and enhanced the test accuracy of DATM on Tiny ImageNet by **0.6%**. For FTD, we used EMA (exponential moving average) just as in the original method[12]. All hyperparameters are detailed in Tables 5, 6, 7, and 8.

Generalization performance to other architectures. We evaluated the generalizability of synthetic datasets generated through distillation. We used DSAC and DATM, which are current SOTA methods in GM-based and TM-based distillation, respectively. After distillation, the synthetic datasets were assessed using various neural networks, including ResNet-18 [16], VGG-11 [37], AlexNet [20], LeNet [22] and MLP. As shown in Table 3, even though our synthetic datasets were distilled using ConvNet, it generalizes well across most networks. Notably, for the experiment of DATM on CIFAR-10 with IPC = 1, employing SDC resulted in an accuracy improvement of **4.61%** when using AlexNet. Employing SDC to DSAC led to an accuracy improvement of **0.9%** on SVHN with IPC = 10 when using MLP. Additional results can be found in Appendix 3.1.

4.3. Further Discussions

Discussion of SDC coefficient λ. The selection of the regularization coefficient λ is pivotal for the quality of the distilled dataset. Our theory suggests that a larger λ typically produces better synthetic datasets for smaller IPC values. Ideally, for low IPC settings, it is better to employ a large λ to strongly penalize sample difficulty, whereas, for high IPC settings, the required λ can be small or even close to zero in extreme cases. For simplicity and to maintain consistency across different datasets and baseline methods, we have set $\lambda = 0.002$ as the default value in most of our experiments. As demonstrated in Figure 5, this choice of λ aligns with the IPC values. Results for FTD and TESLA are based on CIFAR-10, results for DSA are based on SVHN, and results for DSAC are based on MNIST. Additionally, we further show that the choice of λ is not sensitive in Appendix 3.3.

Adaptive sample difficulty correction by adaptively increasing λ during distillation. While our SDC seeks simplicity in regularization, DATM [15] claims that the matching difficulty is increased through optimization. Inspired by their observation, we implemented a strategy where λ increases progressively throughout the matching phases. This method is designed to incrementally adjust the focus from easier to more complex patterns. Inspired by their observation, we applied an *Adaptive Sample Difficulty Correction* (ASDC) strategy in our experiments with a TM-based method on the CIFAR-100 with IPC = 1 and with a GM-based method on the FashionMNIST with IPC = 1. The λ of DATM was initialized to 0.02 and logarithmically increased to 0.08 over 10,000 iterations and DSAC was initialized to 0.002 and logarithmically increased to 0.008 over 10,000 steps. For DATM, we use max test accuracy, while for DSAC, we use test accuracy. Experimental results of ASDC validate its potential to significantly enhance learn-

Table 2. Comparison of test accuracy (%) results of TM-based dataset distillation methods w/ and w/o SDC on CIFAR-10, CIFAR-100, and Tiny ImageNet datasets.

Dataset	CIFAR-10			CIFAR-100			Tiny ImageNet		
IPC	1	10	50	1	10	50	1	10	50
Ratio (%)	0.02	0.2	1	0.2	2	10	0.2	2	10
Random	14.4±2.0	26.0±1.2	43.4±1.0	4.2±0.3	14.6±0.5	30.0±0.4	1.4±0.1	5.0±0.2	15.0±0.4
Herding	21.5±1.2	31.6±0.7	40.4±0.6	8.4±0.3	17.3±0.3	33.7±0.5	2.8±0.2	6.3±0.2	16.7±0.3
Forgetting	13.5±1.2	23.3±1.0	23.3±1.1	4.5±0.2	15.1±0.3	30.5±0.3	1.6±0.1	5.1±0.2	15.0±0.3
MTT	45.8±0.3	64.7±0.5	71.5±0.5	23.9±1.0	38.7±0.4	47.3±0.1	8.3±0.4	20.6±0.2	28.0±0.3
+SDC	46.2±0.7	65.3±0.3	71.8±0.5	24.3±0.3	38.8±0.3	47.3±0.2	8.5±0.2	20.7±0.2	28.0±0.2
FTD	46.7±0.7	65.2±0.5	72.2±0.1	25.1±0.4	42.5±0.1	50.3±0.3	10.9±0.1	21.8±0.3	-
+SDC	47.2±0.7	66.4±0.4	73.3±0.4	25.4±0.3	42.6±0.1	50.5±0.3	11.2±0.1	22.2±0.2	-
TESLA	47.4±0.3	65.0±0.7	71.4±0.5	23.9±0.3	35.8±0.7	44.9±0.4	-	-	-
+SDC	47.9±0.7	65.3±0.4	71.8±0.2	24.2±0.2	35.9±0.2	45.0±0.4	-	-	-
DATM	46.1±0.5	66.4±0.6	75.9±0.3	27.7±0.3	47.6±0.2	52.1±0.1	17.1±0.3	30.1±0.3	39.7±0.1
+SDC	46.4±0.4	66.6±0.4	76.1±0.2	28.0±0.2	47.8±0.2	52.5±0.2	17.4±0.2	30.7±0.2	39.9±0.2
Whole Dataset	84.8±0.1			56.2±0.3			37.6±0.4		

Table 3. Cross-architecture evaluation was conducted on the distilled dataset with (a) IPC = 1 for the TM-based method (DATM) and (b) IPC = 10 for the GM-based method (DSAC), both w/ and w/o SDC. Adding SDC improves performance on unseen networks compared to current SOTA methods.

(a)

Dataset	Method	ResNet-18	VGG-11	AlexNet	LeNet	MLP
CIFAR-10	DATM	29.62	25.12	19.38	23.41	**23.08**
	+SDC	**31.29**	**25.99**	**23.99**	**23.65**	22.90
CIFAR-100	DATM	11.52	8.74	1.95	6.71	6.47
	+SDC	**12.10**	**8.78**	**3.73**	**6.84**	**6.51**
Tiny ImageNet	DATM	4.36	5.93	4.33	2.42	2.29
	+SDC	**4.74**	**6.45**	**4.34**	**2.79**	**2.32**

(b)

Dataset	Method	ResNet-18	VGG-11	AlexNet	LeNet	MLP
MNIST	DSAC	97.44	96.88	95.30	95.31	90.62
	+SDC	**97.65**	**97.13**	**95.73**	**95.67**	**90.93**
FashionMNIST	DSAC	82.17	82.59	80.73	79.82	80.09
	+SDC	**82.87**	**82.73**	**81.15**	**79.96**	**80.36**
SVHN	DSAC	70.59	76.44	49.66	55.98	39.11
	+SDC	**71.03**	**76.63**	**49.78**	**56.27**	**40.03**

ing by finetuning regularization according to the complexity of the learned patterns. Figure 6 illustrates that ASDC further improves our method within SOTA matching methods. Additional results are provided in Appendix 3.2.

5. Conclusion

In this study, we empirically examine the matching-based dataset distillation method in relation to *sample difficulty*, observing clear trends as measured by gradient norm. Additionally, we adapt a neural scaling law from data pruning to theoretically explain dataset distillation. Our theoretical analysis suggests that for small synthetic datasets, the optimal approach is to generate data using easier samples from the original dataset rather than harder ones. To facilitate

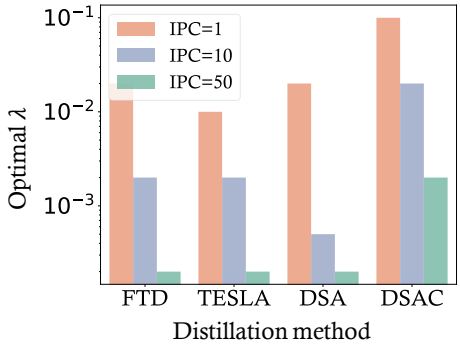

Figure 5. Optimal λ values for various matching-based distillation methods with SDC, performed on datasets with different IPC values.

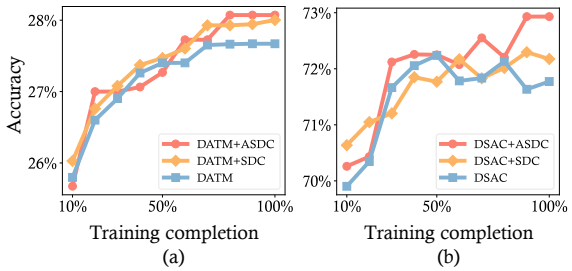

Figure 6. Flexibly adjusting sample difficulty correction with an adaptive increase in λ results in higher accuracy compared to standard SDC and baseline methods. We present the results of (a) DATM and (b) DSAC.

this, we propose a simplicity-centric regularization method, termed *Sample Difficulty Correction* (SDC), aimed at improving synthetic data quality by predominantly utilizing easier samples in the data generation process. This method can be easily incorporated to existing matching-based methods, and can be implemented with a few lines of

code. Experimental results underscore the importance of proper regularization within the optimization process for dataset distillation. We anticipate that this work will deepen the theoretical understanding of dataset distillation.

References

[1] Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023. 1

[2] Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales. Flexible dataset distillation: Learn labels instead of images. *arXiv preprint arXiv:2006.08572*, 2020. 1

[3] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4750–4759, 2022. 1, 2, 3, 6

[4] Dingfan Chen, Raouf Kerkouche, and Mario Fritz. Private set generation with discriminative information. *Advances in Neural Information Processing Systems*, 35:14678–14690, 2022. 1

[5] Jongwon Choi, Kwang Moo Yi, Jihoon Kim, Jinho Choo, Byoungjip Kim, Jinyeop Chang, Youngjune Gwon, and Hyung Jin Chang. Vab-al: Incorporating class imbalance and difficulty with variational bayes for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6749–6758, 2021. 3

[6] Ming-Yu Chung, Sheng-Yen Chou, Chia-Mu Yu, Pin-Yu Chen, Sy-Yen Kuo, and Tsung-Yi Ho. Rethinking backdoor attacks on dataset distillation: A kernel method perspective. *arXiv preprint arXiv:2311.16646*, 2023. 1

[7] Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. *arXiv preprint arXiv:1906.11829*, 2019. 1

[8] Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Scaling up dataset distillation to imagenet-1k with constant memory. In *International Conference on Machine Learning*, pages 6565–6590. PMLR, 2023. 1, 2, 3, 6

[9] Peng Cui, Dan Zhang, Zhijie Deng, Yinpeng Dong, and Jun Zhu. Learning sample difficulty from pre-trained models for reliable prediction. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3

[10] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 2, 6

[11] Tian Dong, Bo Zhao, and Lingjuan Lyu. Privacy for free: How does dataset condensation help privacy? In *International Conference on Machine Learning*, pages 5378–5396. PMLR, 2022. 1

[12] Jiawei Du, Yidi Jiang, Vincent YF Tan, Joey Tianyi Zhou, and Haizhou Li. Minimizing the accumulated trajectory error to improve dataset distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3749–3758, 2023. 1, 2, 3, 6, 7

[13] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pages 2242–2251. PMLR, 2019. 1

[14] Jianyang Gu, Kai Wang, Wei Jiang, and Yang You. Summarizing stream data for memory-restricted online continual learning. *arXiv preprint arXiv:2305.16645*, 2023. 1

[15] Ziyao Guo, Kai Wang, George Cazenavette, Hui Li, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching. *arXiv preprint arXiv:2310.05773*, 2023. 1, 2, 3, 6, 7

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

[17] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1

[18] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. In *International Conference on Machine Learning*, pages 11102–11118. PMLR, 2022. 1

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 6

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 7

[21] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 2, 6

[22] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 7

[23] Saehyung Lee, Sanghyuk Chun, Sangwon Jung, Sangdoo Yun, and Sungroh Yoon. Dataset condensation with contrastive signals. In *International Conference on Machine Learning*, pages 12352–12364. PMLR, 2022. 1, 2, 3, 6

[24] Ming Li, Lichang Chen, Jiuhai Chen, Shwai He, Jiuxiang Gu, and Tianyi Zhou. Selective reflection-tuning: Student-selected data recycling for llm instruction-tuning. *arXiv preprint arXiv:2402.10110*, 2024. 1, 3

[25] Xinyu Lin, Wenjie Wang, Yongqi Li, Shuo Yang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. Data-efficient fine-tuning for llm-based recommendation. *arXiv preprint arXiv:2401.17197*, 2024. 1

[26] Noel Loo, Ramin Hasani, Mathias Lechner, Alexander Amini, and Daniela Rus. Understanding reconstruction attacks with the neural tangent kernel and dataset distillation. *arXiv preprint arXiv:2302.01428*, 2023. 1

[27] Adyasha Maharana, Prateek Yadav, and Mohit Bansal. D2 pruning: Message passing for balancing diversity and dif-

ficulty in data pruning. *arXiv preprint arXiv:2310.07931*, 2023. 1, 3

[28] Javier Maroto and Pascal Frossard. Puma: margin-based data pruning. *arXiv preprint arXiv:2405.06298*, 2024. 1, 3

[29] Wojciech Masarczyk and Ivona Tautkute. Reducing catastrophic forgetting with learning on synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 252–253, 2020. 1

[30] Kristof Meding, Luca M. Schulze Buschoff, Robert Geirhos, and Felix A. Wichmann. Trivial or impossible – dichotomous data difficulty masks model differences (on imagenet and beyond), 2022. 3, 4

[31] Dmitry Medvedev and Alexander D'yakonov. Learning to generate synthetic training data using gradient matching and implicit differentiation. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 138–150. Springer, 2021. 1

[32] Marc Mézard, Giorgio Parisi, and Miguel Angel Virasoro. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*. World Scientific Publishing Company, 1987. 4

[33] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, page 7. Granada, Spain, 2011. 2, 6

[34] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training, 2023. 1, 3

[35] Andrea Rosasco, Antonio Carta, Andrea Cossu, Vincenzo Lomonaco, and Davide Bacciu. Distilled replay: Overcoming forgetting through synthetic samples, 2021. 1

[36] Noveen Sachdeva and Julian McAuley. Data distillation: A survey. *arXiv preprint arXiv:2301.04272*, 2023. 1

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[38] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022. 1, 2, 4, 5, 6, 7

[39] 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. 1

[40] Haoru Tan, Sitong Wu, Fei Du, Yukang Chen, Zhibin Wang, Fan Wang, and Xiaojuan Qi. Data pruning via moving-one-sample-out. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3

[41] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018. 1, 6

[42] Shaobo Wang, Xiangqi Jin, Ziming Wang, Jize Wang, Jiajun Zhang, Kaixin Li, Zichen Wen, Zhong Li, Conghui He, Xuming Hu, and Linfeng Zhang. Data whisperer: Efficient data selection for task-specific llm fine-tuning via few-shot in-context learning. *Annual Meeting of the Association for Computational Linguistics*, 2025. 1

[43] Shaobo Wang, Yicun Yang, Zhiyuan Liu, Chenghao Sun, Xuming Hu, Conghui He, and Linfeng Zhang. Dataset distillation with neural characteristic function: A minmax perspective. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2025. 1

[44] Shaobo Wang, Yantai Yang, Shuaiyu Zhang, Chenghao Sun, Weiya Li, Xuming Hu, and Linfeng Zhang. DRUPI: Dataset reduction using privileged information. In *The Future of Machine Learning Data Practices and Repositories at ICLR 2025*, 2025.

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

[46] Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th annual international conference on machine learning*, pages 1121–1128, 2009. 6

[47] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. 2, 6

[48] Shuai Xie, Zunlei Feng, Ying Chen, Songtao Sun, Chao Ma, and Mingli Song. Deal: Difficulty-aware active learning for semantic segmentation. In *Proceedings of the Asian conference on computer vision*, 2020. 3

[49] Furui Xu*, Shaobo Wang*, Chenghao Sun, Jiajun Zhang, and Linfeng Zhang. Rethink dataset pruning from a generalization perspective. *The Future of Machine Learning Data Practices and Repositories at ICLR 2025*, 2025. 1

[50] Enneng Yang, Li Shen, Zhenyi Wang, Tongliang Liu, and Guibing Guo. An efficient dataset condensation plugin and its application to continual learning. *Advances in Neural Information Processing Systems*, 36, 2023. 1

[51] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021. 1, 2, 3, 6

[52] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*, 2020. 1, 2, 6

Not All Samples Should Be Utilized Equally: Towards Understanding and Improving Dataset Distillation

Supplementary Material

1. More Result on the Relationships between Sample Difficulty, Gardient Norm, and Loss

In this section, we present further findings on the relationships between sample difficulty $\chi(x, y; \Theta_t)$, gradient norm $\text{GraDN}(x, y; \Theta_t)$, and the average validation loss for each sample (x, y) across a range of models characterized by $\theta_t \in \Theta_t$. The experiments were conducted using ResNet-18 on CIFAR-10 and ResNet-34 on CIFAR-100. Each metric was evaluated across 20 pretrained models. We randomly selected 1000 samples for each category in CIFAR-10, and 100 samples for each category in CIFAR-100. For CIFAR-100, 10 categories for visualization were randomly selected for visualization purposes. As shown in Figure 7, Figure 8, Figure 9, Figure 10, Figure 11, Figure 12, it reveals a significant positive correlation between sample difficulty, gradient norm and loss.

2. More Details of Experiments

2.1. Parameter Tables

2.1.1 GM-based Methods

Regarding the GM-based methods, Table 4 provides the corresponding λ values after applying SDC. All results are obtained from a single experiment, and evaluated 20 times. Baseline results are obtained using identical configurations with the original methods' implementations (please refer to DC and DSA[1], and DCC[2]). Experiments with our SDC share consistent hyperparameters with the corresponding baselines.

2.1.2 TM-based Methods

The hyperparameters used in our TM-based methods differ slightly from the original methods (see original implementations of MTT[3], DATM[4], TESLA[5], and FTD[6]), particularly in terms of synthesis steps, number of evaluations, and evaluation interval. Our baseline results used the settings in Table 5, Table 6, Table 7 and Table 8. The experiments of applying SDC were conducted in the same setting as in the baselines. In Table 6, and Table 7, we report the optimal hyperparameters using the ConvNetD3 network. All combinations in Table 7 and Table 8 used the ZCA.

2.2. Limitation

Computational Cost: Similar to other methods, we have not yet addressed the large computational cost associated with the dataset distillation. Our experiments were conducted on a mix of RTX 2080 Ti, RTX 3090, RTX 4090, NVIDIA A100, and NVIDIA V100 GPUs. The cost in terms of computational resources and time remains significant for large datasets and high IPC experiments. For example, distilling Tiny ImageNet using DATM with IPC = 1 requires approximately 150GB of GPU memory, and for IPC = 50, a single experiment can take nearly 24 hours to complete.

Hyperparameter Tuning: The selection of the λ requires manual adjustment, which may involve additional costs. The extensive training durations and substantial GPU memory requirements make it challenging to conduct exhaustive experiments with multiple λ values to identify the global optimum, given our computational resource limitations. By exploring a wider range of λ values, it is possible to obtain better results.

2.3. Pseudocodes of adding SDC on Matching-based Distillation Methods

We provide detailed pseudocodes for GM-based methods and TM-based methods. We take DC as the standard GM-based method, and MTT as the standard TM-based method. The detailed pseudocodes are shown in Algorithm 1 for GM-based methods and Algorithm 2 for TM-based methods.

3. Exploring the Effectiveness of SDC in Additional Experiments

3.1. More Results on the Cross-architecture Evaluation

To evaluate the performance of distilled datasets on different network architectures using SDC (marked as +SDC in the tables) and other methods (DATM and DSAC), we conducted cross-architecture evaluation experiments. We compared the effects of DATM and SDC on CIFAR-10, CIFAR-100, and Tiny ImageNet datasets, and the effects of DSAC and SDC on MNIST, FashionMNIST, and SVHN datasets.

[1] https://github.com/VICO-UoE/DatasetCondensation
[2] https://github.com/Saehyung-Lee/DCC
[3] https://github.com/GeorgeCazenavette/mtt-distillation
[4] https://github.com/NUS-HPC-AI-Lab/DATM
[5] https://github.com/justincui03/tesla
[6] https://github.com/AngusDujw/FTD-distillation

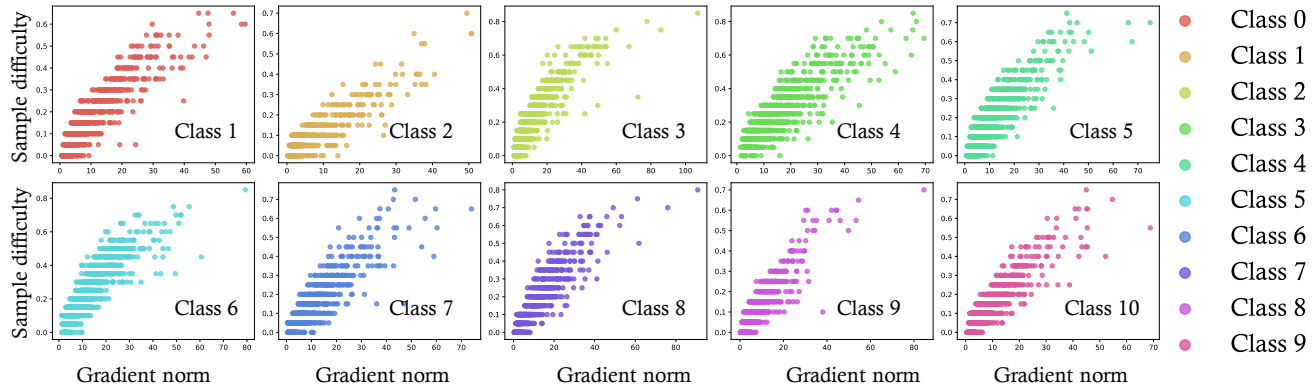

Figure 7. The statistical relationship between sample difficulty $\chi(x, y; \Theta_t)$, gradient norm $\text{GraDN}(x, y; \Theta_t)$ for each sample (x, y) on a series of ResNet-18 models with parameters $\theta_t \in \Theta_t$ on CIFAR-10. 1000 samples were randomly selected for each category.

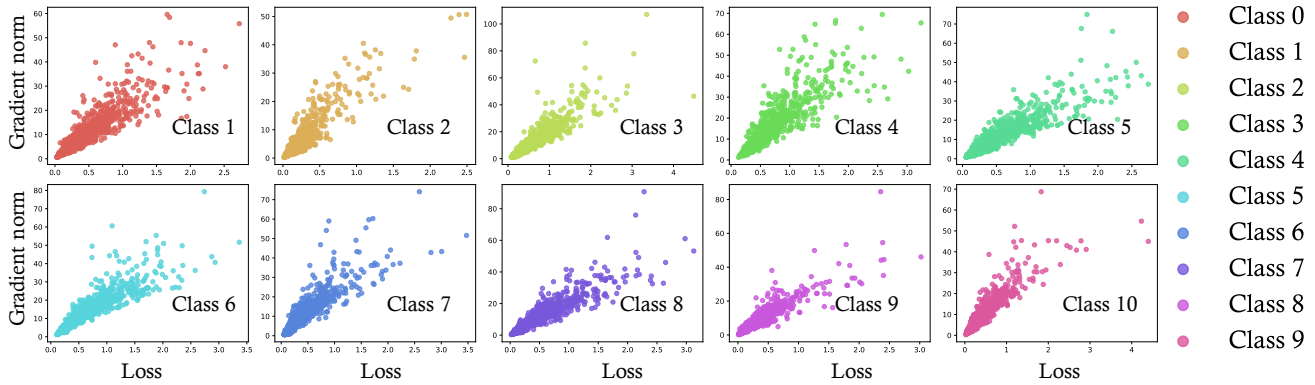

Figure 8. The statistical relationship between gradient norm $\text{GraDN}(x, y; \Theta_t)$ and average validation loss for each sample (x, y) on a series of ResNet-18 models with parameters $\theta_t \in \Theta_t$ on CIFAR-10. 1000 samples were randomly selected for each category.

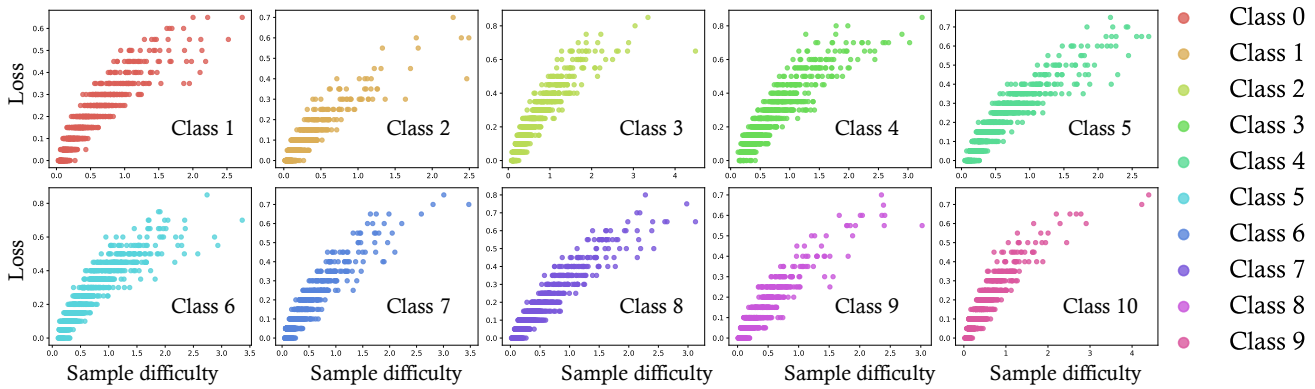

Figure 9. The statistical relationship between sample difficulty $\chi(x, y; \Theta_t)$ and gradient norm $\text{GraDN}(x, y; \Theta_t)$ for each sample (x, y) on a series of ResNet-18 models with parameters $\theta_t \in \Theta_t$ on CIFAR-10. 1000 samples were randomly selected for each category.

Finally, we further evaluated the performance differences between DSAC and SDC methods on MNIST, Fashion-MNIST, and SVHN datasets with IPC = 50. The cross-architecture evaluation experiments for DSAC and DATM, as well as the use of the SDC method on datasets with IPC = 1 of DATM and IPC = 10 of DSAC, can be found in Table 3.

The results of evaluating distilled datasets learned through DATM and SDC methods on CIFAR-10, CIFAR-100, and Tiny ImageNet datasets using ResNet-18, VGG-11, AlexNet, LeNet, and MLP networks are presented in Table 9. For instance, on the CIFAR-100 dataset, the accuracy of the VGG-11 network improved by **1.46%**. It can be

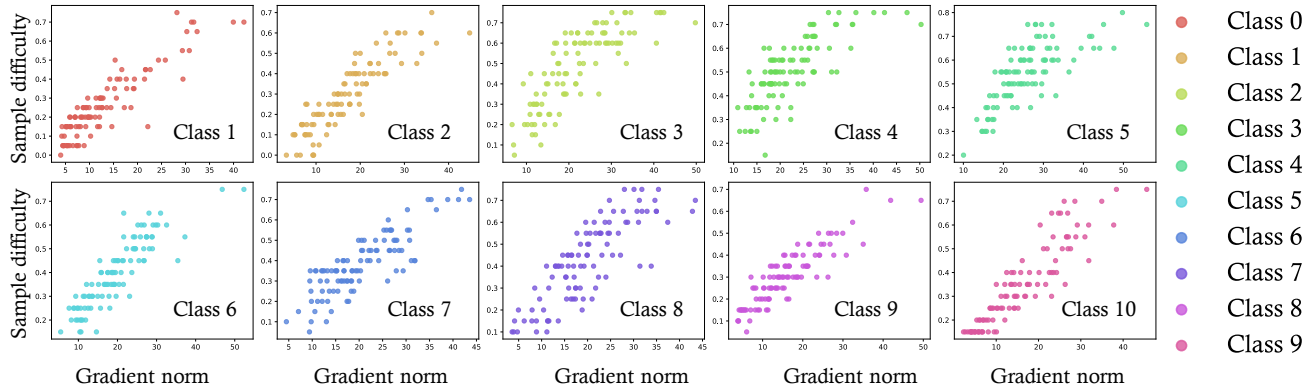

Figure 10. The statistical relationship between sample difficulty $\chi(x, y; \Theta_t)$, gradient norm $\text{GraDN}(x, y; \Theta_t)$ for each sample (x, y) on a series of ResNet-34 models with parameters $\theta_t \in \Theta_t$ on CIFAR-100. 100 samples were randomly selected for each category.

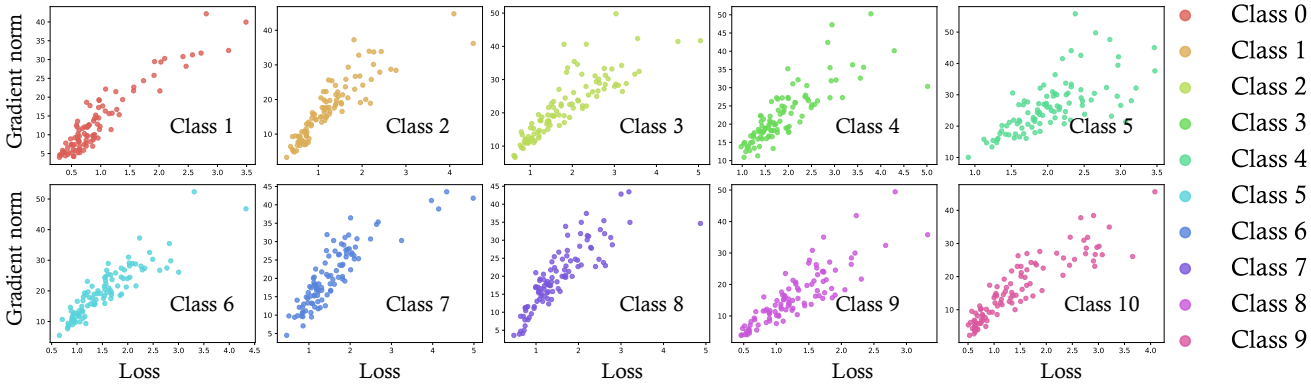

Figure 11. The statistical relationship between gradient norm $\text{GraDN}(x, y; \Theta_t)$ and average validation loss for each sample (x, y) on a series of ResNet-34 models with parameters $\theta_t \in \Theta_t$ on CIFAR-100. 100 samples were randomly selected for each category.

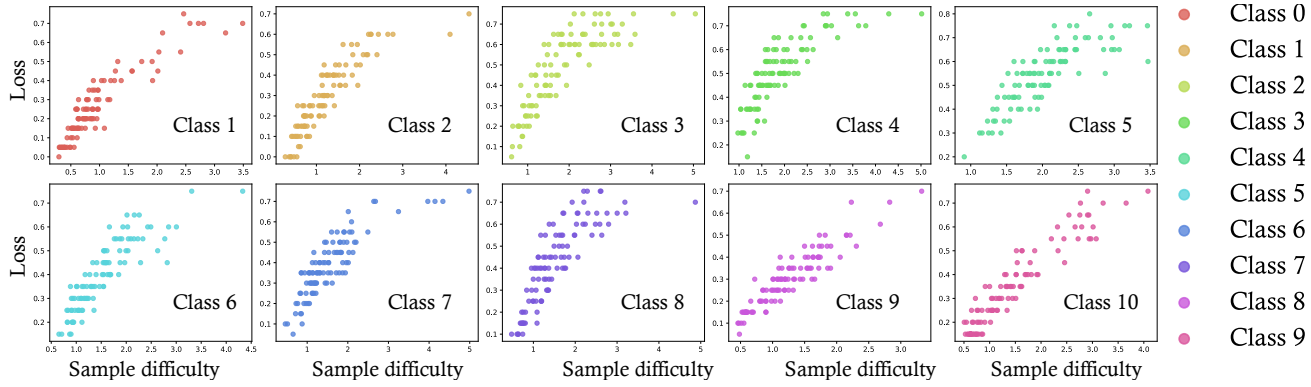

Figure 12. The statistical relationship between sample difficulty $\chi(x, y; \Theta_t)$ and gradient norm $\text{GraDN}(x, y; \Theta_t)$ for each sample (x, y) on a series of ResNet-34 models with parameters $\theta_t \in \Theta_t$ on CIFAR-100. 100 samples were randomly selected for each category.

observed that the performance after applying SDC is generally better than DATM.

In our evaluation of distilled datasets learned through DSAC and SDC methods on MNIST, FashionMNIST, and SVHN datasets using the same network architectures, as detailed in Table 10, the results show that performance after applying SDC is superior to the DSAC method across datasets and network architectures. For example, on the MNIST dataset, the accuracy of the VGG-11 network improved by **1.37%**, and on the SVHN dataset, the accuracy of the ResNet-18 improved by **0.61%**.

Additionally, similar evaluation results on MNIST, FashionMNIST, and SVHN datasets with an IPC value of 50 are summarized in Table 11. For example, on the SVHN dataset, the accuracy of the LeNet network improved by **1.0%**. It can be seen that with an increase in IPC value,

Table 4. The λ used for GM-based methods

Dataset	MNIST			FashionMNIST			SVHN		
IPC	1	10	50	1	10	50	1	10	50
DC	0.001	0.0005	0.001	0.0002	0.001	0.01	0.001	0.0005	0.0002
DSA	0.001	0.00002	0.001	0.0002	0.002	0.001	0.005	0.0005	0.01
DSAC	0.001	0.02	0.02	0.002	0.02	0.02	0.005	0.02	0.02

Table 5. Optimal hyperparameters for MTT. A synthesis batch size of "-" means that we used the full support set at each synthesis step.

Dataset	Model	IPC	ZCA	Synthetic Steps (N)	Expert Epochs ($M^{\dagger}$)	Max Start Epoch (T^{+})	Synthetic Batch Size	Learning Rate (Pixels)	Learning Rate (Step Size)	Starting Synthetic Step Size	Num Eval	Eval Iteration	λ
CIFAR-10	ConvNetD3	1	Y	50	2	2	-	10^2	10^{-7}	10^{-2}	5	100	0.0005
		10	Y	30	2	20	-	10^2	10^{-4}	10^{-2}	5	100	0.02
		50	N	30	2	40	-	10^3	10^{-5}	10^{-3}	5	100	0.0002
CIFAR-100	ConvNetD3	1	Y	20	3	20	-	10^3	10^{-5}	10^{-2}	5	100	0.001
		10	N	20	2	20	-	10^3	10^{-5}	10^{-2}	5	100	0.02
		50	Y	80	2	40	125	10^3	10^{-5}	10^{-2}	5	100	0.002
Tiny ImageNet	ConvNetD4	1	N	10	2	10	-	10^4	10^{-4}	10^{-2}	5	100	0.005
		10	N	20	2	40	200	10^4	10^{-4}	10^{-2}	3	200	0.02
		50	N	20	2	40	300	10^4	10^{-4}	10^{-2}	3	200	0.02

Algorithm 1 Gradient Matching with Sample Difficulty Correction

Input: Training set $\mathcal{D}_{\text{real}}$, category set C, classification cross-entropy loss $\mathcal{L}$, probability distribution for weights P_θ, distance metric $\mathbf{D}$, regularization coefficient λ, number of steps T, learning rate η for network parameters.

Initialize distilled data $\mathcal{D}_{\text{syn}} \sim \mathcal{D}_{\text{real}}$.
for each distillation step... **do**
  ▷ Initialize network $\theta_0 \sim P_\theta$
  for $t = 0 \rightarrow T$ **do**
    for $c = 0 \rightarrow C-1$ **do**
      ▷ Sample a mini-batch of distilled images: $\mathcal{B}^c_{\text{real}} \sim \mathcal{D}^c_{\text{real}}$
      ▷ Sample a mini-batch of original images: $\mathcal{B}^c_{\text{syn}} \sim \mathcal{D}^c_{\text{syn}}$
      ▷ Compute $\mathcal{L}_{\mathcal{B}^c_{\text{syn}}} = \mathbb{E}_{(x,y)\in\tilde{\mathcal{B}}^c_{\text{real}}}\left[\mathcal{L}(x,y;\theta_t)\right]$, $\mathcal{L}_{\mathcal{B}^c_{\text{real}}} = \mathbb{E}_{(x,y)\in\tilde{\mathcal{B}}^c_{\text{real}}}\left[\mathcal{L}(x,y;\theta_t)\right]$
      ▷ Compute gradient matching loss $L = \mathbf{D}\left(\nabla_\theta \mathcal{L}_{\mathcal{B}^c_{\text{real}}}, \nabla_\theta \mathcal{L}_{\mathcal{B}^c_{\text{syn}}}\right) + \lambda \left\|\nabla_\theta \mathcal{L}_{\mathcal{B}_{\text{syn}}}\right\|_2^2$
      ▷ Update $\mathcal{D}_{\text{syn}}$ *w.r.t.* L
    end for
    ▷ Update network *w.r.t.* classification loss: $\theta_{t+1} = \theta_t - \eta\nabla\mathcal{L}_{\mathcal{D}_{\text{syn}}}(\theta_t)$
  end for
end for

Output: distilled data $\mathcal{D}_{\text{syn}}$

Algorithm 2 Trajectory Matching with Sample Difficulty Correction

Input: Set of expert parameter trajectories trained on $\mathcal{D}_{\text{real}}$ $\{\tau_i^*\}$, the number of updates between starting and target expert params M, the number of updates to student network per distillation step N, differentiable augmentation function $\mathcal{A}$, maximum start epoch $T^+ < T$, learning rate η for network parameters, regularization coefficient λ, classification cross-entropy loss $\mathcal{L}$.

Initialize distilled data $\mathcal{D}_{\text{syn}} \sim \mathcal{D}_{\text{real}}$.
for each distillation step... **do**
  ▷ Sample expert trajectory: $\tau^* \sim \{\tau_i^*\}$ with $\tau^* = \{\theta_t^{\mathcal{D}_{\text{real}}}\}_0^T$
  ▷ Choose random start epoch, $t \leq T^+$
  ▷ Initialize student network with expert params: $\theta_t^{\mathcal{D}_{\text{syn}}} := \theta_t^{\mathcal{D}_{\text{real}}}$
  for $n = 0 \rightarrow N-1$ **do**
    ▷ Sample a mini-batch of distilled images: $\mathcal{B}_{\text{syn}} \sim \mathcal{D}_{\text{syn}}$
    ▷ Update student network *w.r.t.* classification loss: $\theta_{t+n+1}^{\mathcal{D}_{\text{syn}}} = \theta_{t+n}^{\mathcal{D}_{\text{syn}}} - \eta\nabla\mathcal{L}_{\mathcal{A}(\mathcal{B}_{\text{syn}})}(\theta_{t+n}^{\mathcal{D}_{\text{syn}}})$
  end for
  ▷ Compute loss between ending student and expert params: $L = \frac{\|\theta_{t+N}^{\mathcal{D}_{\text{syn}}} - \theta_{t+M}^{\mathcal{D}_{\text{real}}}\|_2^2}{\|\theta_{t+M}^{\mathcal{D}_{\text{real}}} - \theta_t^{\mathcal{D}_{\text{real}}}\|_2^2} + \lambda\left\|\nabla_\theta\mathcal{L}_{\mathcal{D}_{\text{syn}}}\right\|_2^2$
  ▷ Update $\mathcal{D}_{\text{syn}}$ *w.r.t.* L
end for

Output: distilled data $\mathcal{D}_{\text{syn}}$

the performance after applying SDC remains better in most cases, further demonstrating the superiority of the SDC method in dataset distillation.

3.2. More Results on the Adaptive Sample Difficulty Correction

The dynamic adjustment of SDC, when applied to both DSA and FTD, consistently outperforms both the baseline

Table 6. Optimal hyperparameters for TESLA. A synthesis batch size of "-" means that we used the full support set at each synthesis step.

Dataset	IPC	Matching Steps	Teacher Epochs	Max Start Epoch	Synthetic Batch Size	Learning Rate (Pixels)	Learning Rate (Step Size)	Starting Synthetic Step Size	ZCA	λ
CIFAR-10	1	50	2	3	-	10^2	10^{-7}	10^{-2}	Y	0.01
	10	30	2	20	-	10^2	10^{-4}	10^{-2}	Y	0.002
	50	26	3	40	-	10^3	10^{-5}	10^{-3}	N	0.02
CIFAR-100	1	20	3	20	-	10^3	10^{-5}	10^{-2}	Y	0.001
	10	13	3	30	-	10^3	10^{-5}	10^{-2}	N	0.002
	50	50	2	40	100	10^3	10^{-5}	10^{-2}	Y	0.0002

Table 7. Optimal hyperparameters for FTD. A synthesis batch size of '-' means that we used the full support set at each synthesis step.

Dataset	IPC	Synthetic Step	Expert Epoch	Max Start Epoch	Synthetic Batch Size	Learning Rate (Pixels)	Learning Rate (Step Size)	Learning Rate (Teacher)	Balance coefficient	EMA Decay	λ
CIFAR-10	1	50	2	2	-	100	10^{-7}	0.01	0.3	0.9999	0.002
	10	30	2	20	-	100	10^{-5}	0.001	0.3	0.9995	0.002
	50	30	2	40	-	1000	10^{-5}	0.001	1	0.999	0.0002
CIFAR-100	1	40	3	20	-	1000	10^{-5}	0.01	1	0.9995	0.002
	10	20	2	40	-	1000	10^{-5}	0.01	1	0.9995	0.0002
	50	80	2	40	1000	1000	10^{-5}	0.01	1	0.999	0.002

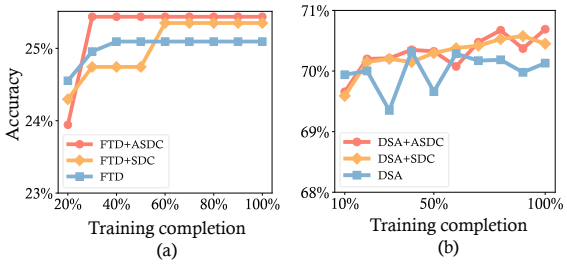

Figure 13. The application of ASDC on (a) FTD and (b) DSA. Gradually increasing λ gives better results than the baseline and the method after applying SDC.

methods and the baseline methods with SDC applied. As shown in Figure 13, we logarithmically increased the λ coefficient for DSA from 0.0002 to 0.002 over 1000 steps and for FTD from 0.002 to 0.008 over 10,000 iterations. The results clearly demonstrate that ASDC yields superior performance. Flexibly adjusting the sample difficulty correction by adaptively increasing λ yields higher accuracy compared to the standard SDC and baseline methods.

3.3. Sensitivity Analysis of SDC coefficient .

In this section, we conducted extensive experiments to study the sensitivity of the hyperparameter λ. Specifically, we conducted experiments of DSA on SVHN dataset with IPC = 1, and DC on SVHN dataset with IPC = 10. As shown in Figure 14, the choice of λ is not sensitive among different matching-based dataset distillation methods.

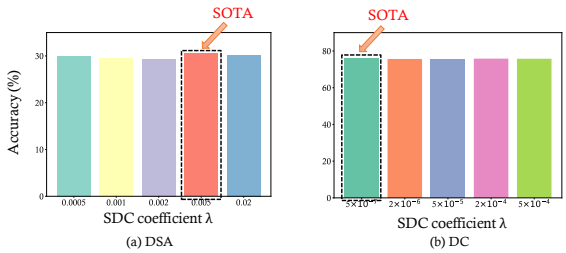

Figure 14. Sensitivity analysis of SDC coefficient λ on different distillation methods. We evaluated the sensitivity across different λs, and showed that the choice of λ did not severely affect the final test performance. (a) DSA on SVHN dataset with IPC = 1 (b) DC on SVHN dataset with IPC = 10.

4. Analytical Theory for Dataset Distillation

In this section, we introduce a theory, adapted from data pruning [38], to the context of dataset distillation within an expert-student perceptron framework, utilizing the tools of statistical mechanics. We investigate the challenge of classifying a dataset $\mathcal{D}_{\text{real}}$ consisting of d_{real} samples $\{x_i, y_i\}_{i=1,\ldots,d_{\text{real}}}$, where the inputs $x_i \sim \mathcal{N}(0, I_d)$ are i.i.d. zero-mean, unit-variance random Gaussian variables, and the labels $y_i = \text{sign}(\theta^{\mathcal{D}_{\text{real}}\top} x_i)$ are generated by an expert perceptron $\theta^{\mathcal{D}_{\text{real}}} \in \mathbb{R}^d$. We assume that the expert perceptron $\theta^{\mathcal{D}_{\text{real}}}$ is randomly drawn from a uniform distribution

Table 8. Optimal hyperparameters for DATM.

Dataset	Model	IPC	Synthetic Step	Expert Epoch	Min Start Epoch	Current Max Start Epoch	Max Start Epoch	Synthetic Batch Size	Learning Rate (Label)	Learning Rate (Pixels)	Num Eval	Eval Iteration	λ
CIFAR-10	ConvNetD3	1	80	2	0	4	4	10	5	100	5	500	0.0002
		10	80	2	0	10	20	100	2	100	5	500	0.0005
		50	80	2	0	20	40	500	2	1000	5	500	0.002
CIFAR-100	ConvNetD3	1	40	3	0	10	20	100	10	1000	5	500	0.02
Tiny ImageNet	ConvNetD4	1	60	2	0	15	20	200	10	10000	5	500	0.002
		10	60	2	10	50	50	250	10	100	3	500	0.002
		50	80	2	40	70	70	250	10	100	3	500	0.002

Table 9. Cross-architecture evaluation. We evaluated distilled datasets with IPC = 10 learned through DATM w/ and w/o SDC on different networks.

Dataset	Method	ResNet-18	VGG-11	AlexNet	LeNet	MLP
CIFAR-10	DATM	36.48	37.32	33.19	32.56	27.21
	+SDC	**38.33**	**38.22**	**34.56**	**33.17**	**27.62**
CIFAR-100	DATM	17.87	14.71	15.09	11.76	11.52
	+SDC	**18.97**	**16.17**	**15.73**	**12.44**	**11.87**
Tiny ImageNet	DATM	6.33	8.67	6.18	3.65	3.34
	+SDC	**7.20**	**9.13**	**6.89**	**3.88**	**3.40**

on the sphere $\theta^{\mathcal{D}_{\text{real}}} \sim \text{Unif}(\mathbb{S}^{d-1}(\sqrt{d}))$. Our analysis is situated within the high-dimensional statistics limit where $d, d_{\text{real}} \to \infty$ but the ratio $\alpha_{\text{real}} = d_{\text{real}}/d$ remains $O(1)$.

Specifically, consider synthesizing a dataset by matching only the samples with the smallest margin $|z_i| = |\theta^{\text{probe}\top} x_i|$ along a probe student θ^{probe}. The distilled dataset will then follow a distribution $p(z)$ in the direction of θ^{probe} while remaining isotropic in the null space of θ^{probe}. We assume, without loss of generality, that θ^{probe} has developed some overlap with the expert, quantified by the angle $\gamma = \cos^{-1}\left(\frac{\theta^{\text{probe}\top}\theta^{\mathcal{D}_{\text{real}}}}{\|\theta^{\text{probe}}\|_2 \|\theta^{\mathcal{D}_{\text{real}}}\|_2}\right)$.

Once the dataset has been distilled, we consider training a new student $\theta^{\mathcal{D}_{\text{syn}}}$ from scratch on this distilled dataset. A typical training algorithm aims to find the solution $\theta^{\mathcal{D}_{\text{syn}}}$ which classifies the training data with maximal margin $\kappa = \min_i(\theta^{\mathcal{D}_{\text{syn}}\top} y_i x_i)$. Our goal is to compute the generalization error ε of this student, governed by the overlap between the student and the expert: $\varepsilon = \cos^{-1}(R)/\pi$, where $R = \frac{\theta^{\mathcal{D}_{\text{syn}}\top}\theta^{\mathcal{D}_{\text{real}}}}{\|\theta^{\mathcal{D}_{\text{syn}}}\|_2 \|\theta^{\mathcal{D}_{\text{real}}}\|_2}$.

We provide saddle point equations for the cosine similarity R between the probe $\theta^{\mathcal{D}_{\text{probe}}}$ and the expert $\theta^{\mathcal{D}_{\text{real}}}$, which will be discussed in Section 4.1 and Section 4.2. For our simulations, we set the parameter dimension $d = 200$ for perfect probe settings, and set $d = 50$ for imperfect probe settings. We averaged 100 simulation results to verify the theory.

4.1. Perfect Expert-Teacher Settings

The solution is given by the following saddle point equations for perfect expert-teacher settings, *i.e.,* $\gamma = 0$. For any given α_{syn}, these equations can be solved for the order parameters R, κ. From these parameters, the generalization error can be computed as $\varepsilon = \cos^{-1}(R)/\pi$.

$$R = \frac{2\alpha_{\text{syn}}}{f\sqrt{2\pi}\sqrt{1-R^2}} \int_{-\infty}^{\kappa} Dt \, \exp\left(-\frac{R^2 t^2}{2(1-R^2)}\right) \times \left[1 - \exp\left(-\frac{\gamma(\gamma - 2Rt)}{2(1-R^2)}\right)\right](\kappa - t)$$

$$1 - R^2 = \frac{2\alpha_{\text{syn}}}{f} \int_{-\infty}^{\kappa} Dt \left[H\left(-\frac{Rt}{\sqrt{1-R^2}}\right) - H\left(-\frac{Rt-\gamma}{\sqrt{1-R^2}}\right)\right](\kappa - t)^2$$

Where $H(x) = \frac{1}{2}\left(1 - \frac{2}{\sqrt{\pi}} \int_0^{\left(\frac{x}{\sqrt{2}}\right)} e^{-t^2}\, dt\right)$. This calculation produces the solid theoretical curves shown in Figure 4, which exhibit an excellent match with numerical simulations. Please refer [38] for detailed deductions.

Table 10. Cross-architecture evaluation. We evaluated distilled datasets with IPC = 1 learned through DSAC w/ and w/o SDC on different networks.

Dataset	Method	ResNet-18	VGG-11	AlexNet	LeNet	MLP
MNIST	DSAC	88.58	79.58	83.63	83.46	72.78
	+SDC	**88.70**	**80.95**	**83.92**	**83.66**	**73.51**
FashionMNIST	DSAC	71.60	68.03	66.03	67.09	63.85
	+SDC	**71.70**	**68.82**	**66.47**	**67.19**	**64.93**
SVHN	DSAC	33.04	32.32	14.63	20.89	13.32
	+SDC	**33.65**	**33.84**	**17.18**	**22.40**	**13.86**

Table 11. Cross-architecture evaluation. We evaluated distilled datasets with IPC = 50 learned through DSAC w/ and w/o SDC on different networks.

Dataset	Method	ResNet-18	VGG-11	AlexNet	LeNet	MLP
MNIST	DSAC	**97.97**	98.53	97.95	97.58	94.70
	+SDC	97.95	**98.57**	**97.97**	**97.62**	**94.75**
FashionMNIST	DSAC	86.92	87.03	**85.61**	84.96	83.56
	+SDC	**86.96**	**87.20**	85.58	**85.36**	**83.78**
SVHN	DSAC	86.10	85.62	83.47	77.92	62.68
	+SDC	**86.32**	**85.86**	**83.85**	**78.92**	**63.47**

4.2. Imperfect Expert-Teacher Settings

We have shown the perfect student settings in Section 4.1. When the probe student does not exactly match the expert, an additional parameter θ characterizes the angle between the probe student and the expert. Furthermore, an additional order parameter $\rho = \theta^{\mathcal{D}_{\text{real}}^\top}\theta^{\mathcal{D}_{\text{syn}}}$ represents the typical student-probe overlap, which must be optimized. Consequently, we derive three saddle point equations.

$$
\begin{aligned}
\frac{R-\rho\cos\gamma}{\sin^2\gamma} &= \frac{\alpha_{\text{syn}}}{\pi\Lambda}\left\langle \int_{-\infty}^{\kappa} dt\ \exp\left(-\frac{\Delta(t,z)}{2\Lambda^2}\right) \right. \\
&\quad \left. \times(\kappa-t)\right\rangle_z \\
1-\frac{\rho^2+R^2-2\rho R\cos\gamma}{\sin^2\gamma} &= 2\alpha_{\text{syn}}\left\langle \int_{-\infty}^{\kappa} dt\ \frac{e^{-\frac{(t-\rho z)^2}{2(1-\rho^2)}}}{\sqrt{2\pi}\sqrt{1-\rho^2}} \right. \\
&\quad \left. \times H\left(\frac{\Gamma(t,z)}{\sqrt{1-\rho^2}\Lambda}\right)(\kappa-t)^2\right\rangle_z \\
\frac{\rho-R\cos\gamma}{\sin^2\gamma} &= 2\alpha_{\text{syn}}\left\langle \int_{-\infty}^{\kappa} dt\ \frac{e^{-\frac{(t-\rho z)^2}{2(1-\rho^2)}}}{\sqrt{2\pi}\sqrt{1-\rho^2}} \right. \\
&\quad \left. \times H\left(\frac{\Gamma(t,z)}{\sqrt{1-\rho^2}\Lambda}\right)\left(\frac{z-\rho t}{1-\rho^2}\right)(\kappa-t)\right\rangle_z \\
&\quad + \frac{1}{2\pi\Lambda}\left\langle \exp\left(-\frac{\Delta(t,z)}{2\Lambda^2}\right) \right. \\
&\quad \left. \times\left(\frac{\rho R-\cos\gamma}{1-\rho^2}\right)(\kappa-t)\right\rangle_z
\end{aligned}
$$

Where,

$$
\begin{aligned}
\Lambda &= \sqrt{\sin^2\gamma - R^2 - \rho^2 + 2\rho R\cos\gamma}, \\
\Gamma(t,z) &= z(\rho R-\cos\gamma) - t(R-\rho\cos\gamma), \\
\Delta(t,z) &= z^2\left(\rho^2+\cos^2\gamma-2\rho R\cos\gamma\right) \\
&\quad + 2tz(R\cos\gamma-\rho) + t^2\sin^2\gamma.
\end{aligned}
$$

The notation $\langle\cdot\rangle_z$ denotes an average over the pruned data distribution $p(z)$ for the probe student. For any given $\alpha_{\text{syn}}, p(z), \gamma$, these equations can be solved for the order parameters R, ρ, κ. From these parameters, the generalization error can be readily obtained as $\varepsilon = \cos^{-1}(R)/\pi$. Our simulation results are shown in Figure 15. Please refer [38] for detailed deductions.

5. Visualization Results

Additionally, we show our visualization of distilled datasets by adding SDC into current matching-based methods, as shown in Figure 16, Figure 17, Figure 18, Figure 19, and Figure 20.

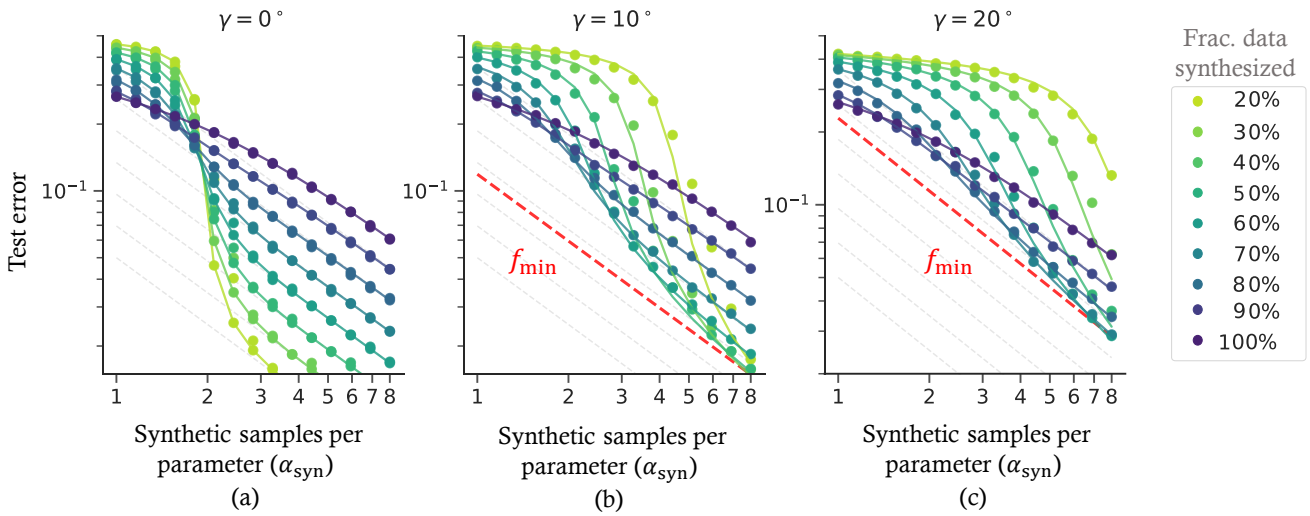

Figure 15. Test error ε as a function of the synthetic samples per parameter α_{syn} and fraction of data synthesized f in (a) the perfect expert setting ($\gamma = 0$) (b) the perfect expert setting ($\gamma = 10°$) (c) the perfect expert setting ($\gamma = 20°$).

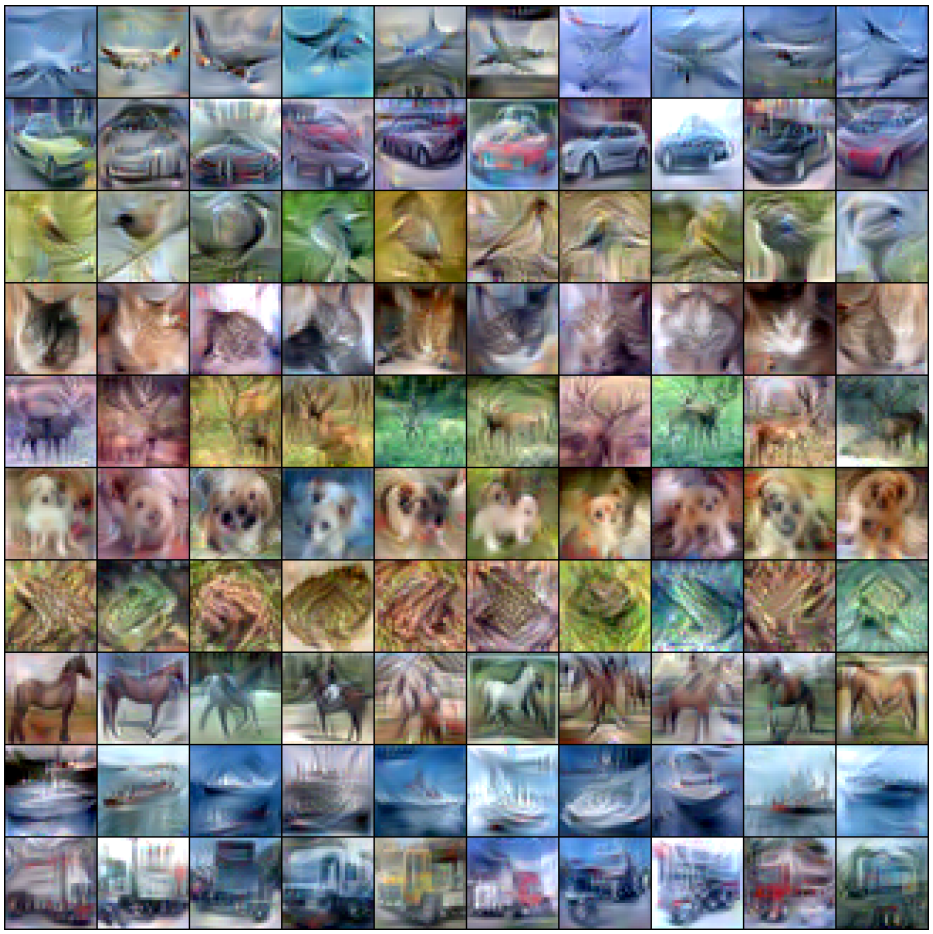

Figure 16. (FTD + SDC, CIFAR-10, IPC = 10) Visualization of distilled images.

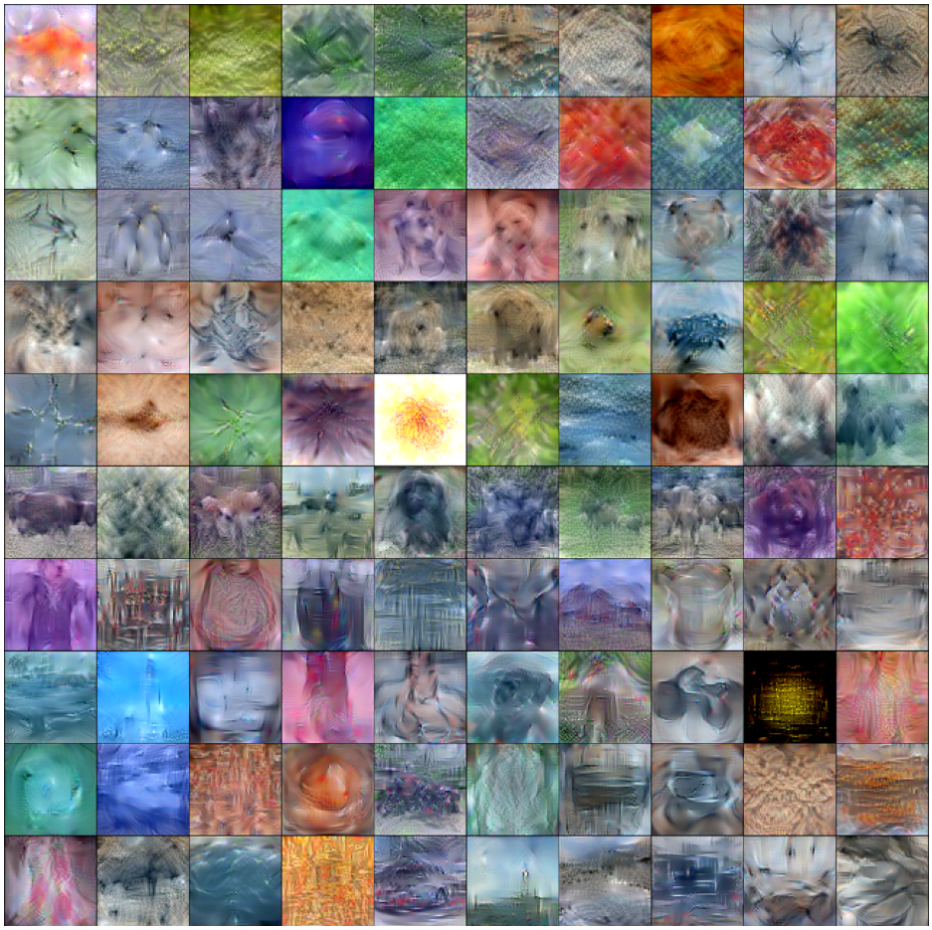

Figure 17. (DATM + SDC, Tiny ImageNet, IPC = 1, 1 / 2) Visualization of distilled images.

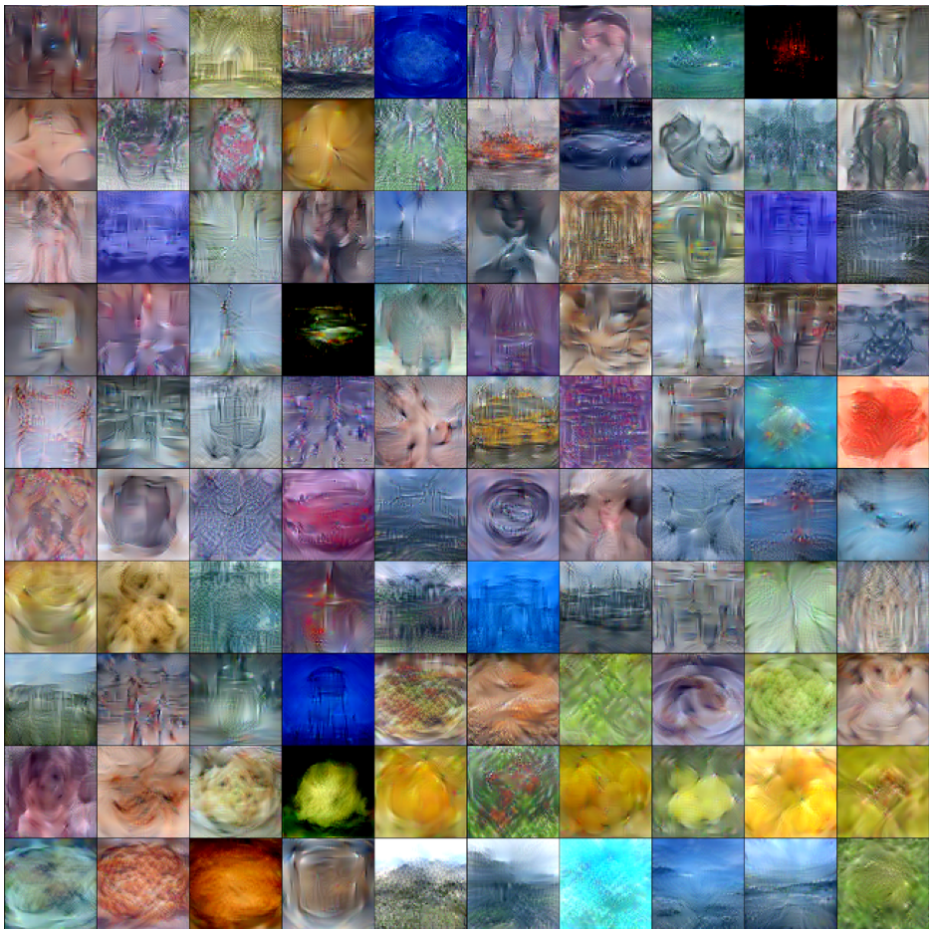

Figure 18. (DATM + SDC, Tiny ImageNet, IPC = 1, 2 / 2) Visualization of distilled images.

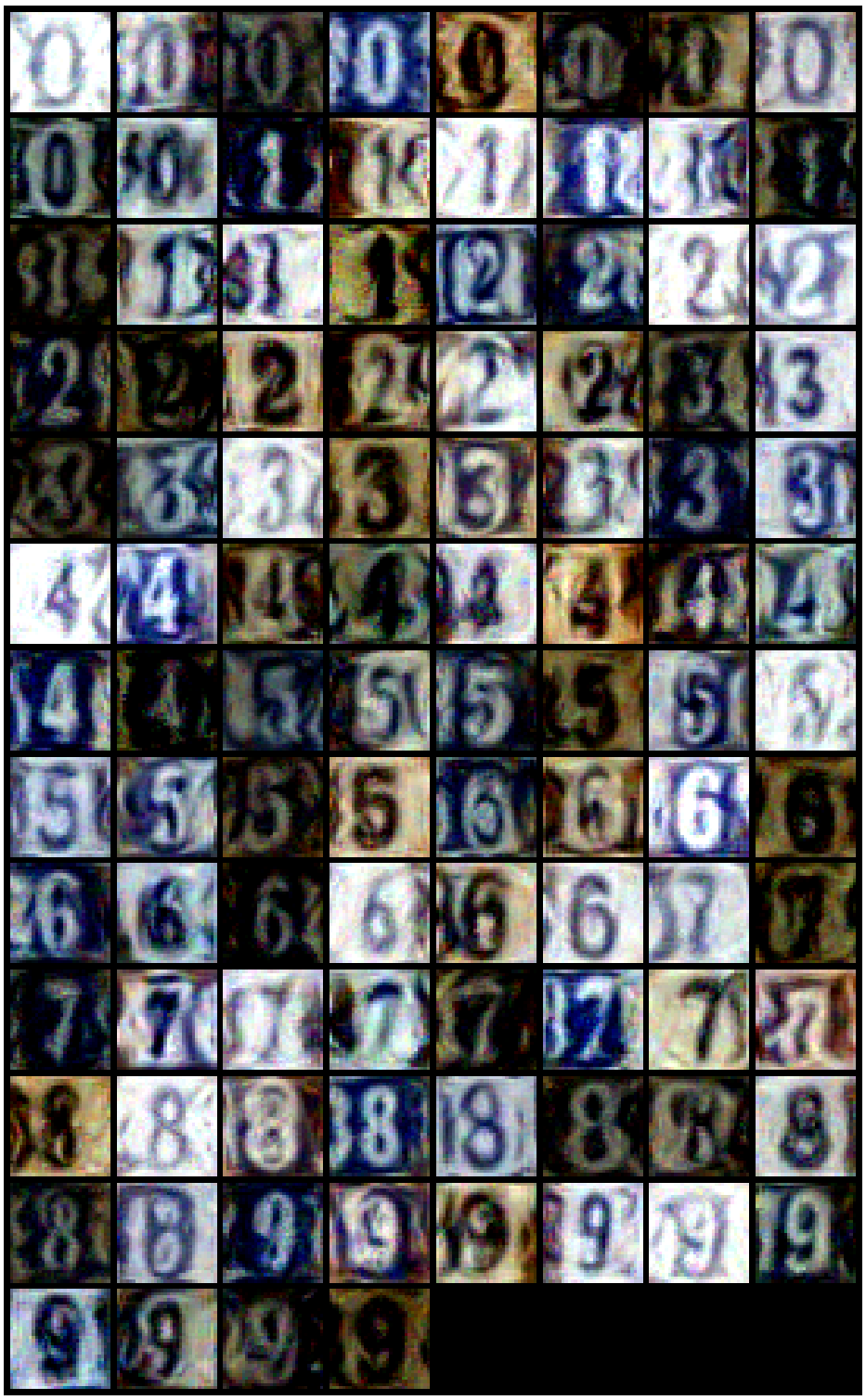

Figure 19. (DSA + SDC, SVHN, IPC = 10) Visualization of distilled images.

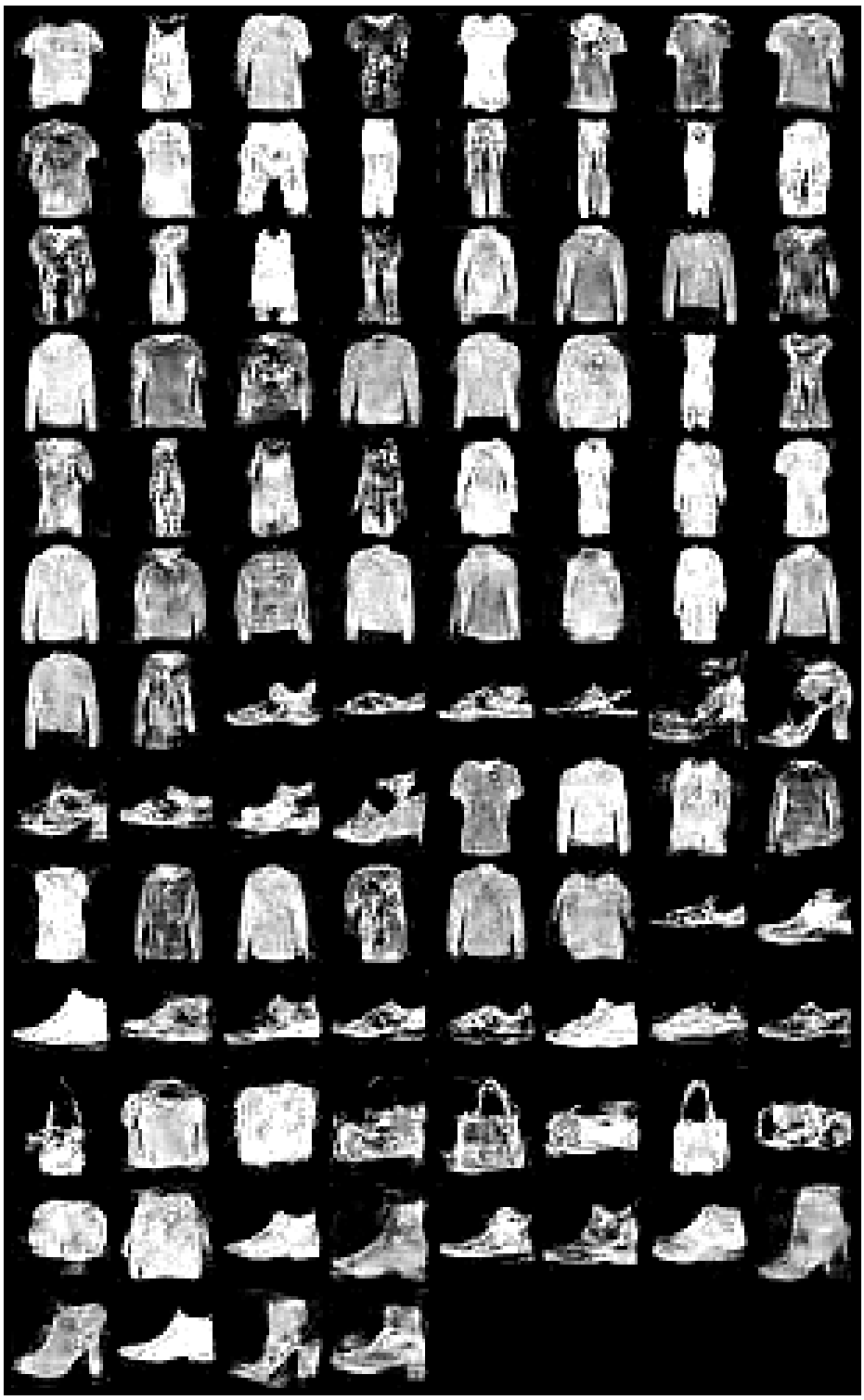

Figure 20. (DC + SDC, FashionMNIST, IPC = 10) Visualization of distilled images.