DynaDistill: Leveraging Real-Time Feedback for Effective Dataset Distillation

Zongxiong Chen[1] Derui Zhu[2] Jiahui Geng[4] Sonja Schimmler[1,3] Manfred Hauswirth[1,3]

[1]Fraunhofer FOKUS [2]Technical University of Munich [3]Technical University of Berlin
[4]Mohamed bin Zayed University of Artificial Intelligence

{zongxiong.chen, sonja.schimmler, manfred.hauswirth}@fokus.fraunhofer.de
derui.zhu@tum.de jiahui.geng@mbzuai.ac.ae

Abstract

Dataset Distillation (DD) aims to compress the knowledge contained in a large-scale dataset into a substantially smaller synthetic dataset. While this synthetic dataset is meticulously crafted to mirror the performance of the original, it poses significant challenges in training efficiency and data utility. This singular focus, especially the selection of a sole expert trajectory in `MTT` or a single model in `IDC`, inadvertently undermines the potential performance of the distilled synthetic dataset at certain intervals within the whole distillation process. This inefficiency necessitates a protracted series of training iterations to culminate in an improved performance outcome. In this paper, we hypothesize that there exists an optimized training routine across the entire optimization phase, specifically for synthetic dataset training through gradient or trajectory matching. To address these challenges, this paper introduces a novel methodology, namely `DynaDistill`, which is designed to expedite the distillation process by dramatically decreasing the required number of distillation steps in current state-of-the-art methods without compromising their performance. Our empirical results demonstrate that our method achieves comparable performance on par with state-of-the-art methods. Moreover, the design of our method allows it to integrate as a plug-and-play module into existing distillation techniques seamlessly.

1. Introduction

The evolution of deep learning has achieved significant success in diverse domains, attributable to the recent advancements in technology and the proliferation of extensive real-world data [8, 15]. Despite these achievements, the reliance on massive volumes of data for training state-of-the-art models like GPT-2 and GPT-3, which require 45GB and 45TB training data respectively [15], introduces substantial challenges. To overcome these limitations, recent research has introduced a concept named *dataset distillation* [3, 14] or *dataset condensation*. The primary objective of dataset distillation is to minimize training costs by creating a small, synthetic dataset that encapsulates the informative knowledge of a large-scale, original dataset. Importantly, this synthetic dataset aims to yield similar performance on corresponding tasks as the larger, original dataset. Dataset distillation offers a promising solution to the challenge of training on memory and computation resource-constrained devices by efficiently condensing the knowledge required for model training.

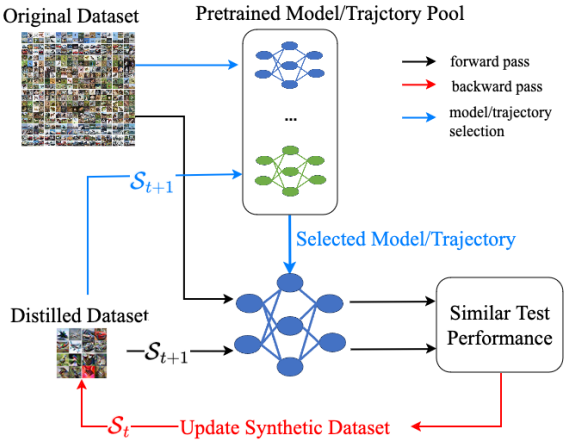


Figure 1. Illustration of (`DynaDistill`). Initially, we prepare a collection of pretrained models or expert trajectories. At each distillation step t, we select a pretrained model/trajectory from the built model pool based on their evaluation performance (i.e. loss value). The synthetic dataset S is given in an autoregressive process, i.e. $\mathcal{S}_{t+1} = \text{Alg}(\mathcal{S}_t, \mathcal{T}; \theta)$, where θ is the selected network parameterized by θ. Model/Trajectory that exhibit superior performance metrics are accorded a higher likelihood of being selected to participate in the distillation process.

Although model training on a small synthetic dataset

is fast, generating a rich, informative synthetic dataset is notably resource-intensive. For example, to achieve state-of-the-art results, MTT[1] requires 250,000 iterations (i.e., 5000 iterations for the outer loop and 50 iterations for the inner loop (synthetic steps)) to distill 50,000 CIFAR-10 images into 10 synthetic images. Similarly, the cutting-edge methods DREAM+ [10] and IDC [6] necessitate approximately 200,000 (i.e., 2000 iterations for the outer loop and 100 iterations for the inner loop) iterations to condense the CIFAR-10 dataset into 10 synthetic images. In addition, these methods often have large oscillations in the training loss curve. This is attributed to all models in the distillation step being randomly initialized from an expert trajectory in MTT. Such variability implies that some initialization at specific distillation steps may degrade the performance of synthetic datasets. Existing methods often overlook the significance of all distillation steps, typically relying on the random selection of a single network for distillation. This indiscriminate approach risks leading to a synthetic dataset in a suboptimal state, where an unfavourable initialization can significantly hamper the performance of the synthetic dataset, thereby prolonging the distillation process.

Motivated by these observations and with the premise of the existence of better distillation steps, this work seeks to streamline the optimization steps involved in dataset distillation and simultaneously preserve or even improve the testing performance over state-of-the-art methods. We propose a simple but effective method DynaDistill, utilizing a synthetic dataset $\mathcal{S}$ at each distillation step as feedback to select a better candidate network to reduce the overall training steps. Overall, the contributions of this paper include:

- We introduce DynaDistill, a novel dataset distillation approach focused on enhancing training efficiency by significantly reducing distillation steps.
- Our extensive experimentation across different datasets and distillation algorithms unambiguously demonstrates that DynaDistill effectively reduces the number of distillation steps required while preserving the performance of the distilled dataset.
- DynaDistill is designed to serve as a straightforward plug-and-play module for both existing and future gradient or trajectory matching methods. This compatibility enables these methods to enhance their efficiency by reducing the overall number of distillation steps required.

2. Existing Dataset Distillation Methods

Originally, dataset distillation was proposed as a meta-learning problem involving bilevel optimization [14]. However, this approach entails back-propagation through time, which requires high computational costs and memory overhead, as well as biases in short unroll or issues of gradient exploding or vanishing in long unrolls. Methods based on surrogate objectives have been proposed to enhance computational efficiency, primarily through gradient matching or training trajectory matching for data distillation. Intuitively, models trained on original and distilled data should exhibit similar parameter weights or gradients. We present the principles of three representative methods: MTT [1], IDC [6], and DREAM+ [10]. Our innovative approach can further improve the performance of these methods by integrating the distribution feedback to improve the utility of compressed data during distillation.

Matching Training Trajectories (MTT) first samples θ_t^* at a random time step t from the expert trajectory to initialize the student model. Then it performs N step gradient descent updates on the student model and selects the expert model θ_{t+M}^* as the target model. The objective of MTT is to make the student model approach the expert model and use the L_2 distance of the expert parameters to normalize the matching loss:

$$\mathcal{L}_{\texttt{MTT}} = \frac{\|\hat{\theta}_{t+N}^{\mathcal{S}} - \theta_{t+M}^{\mathcal{T}}\|_2^2}{\|\theta_t^{\mathcal{T}} - \theta_{t+M}^{\mathcal{T}}\|_2^2}, \tag{1}$$

where M, N are the hyperparameters.

Information-Intensive Dataset Condensation (IDC) aligns gradients of classification loss between synthetic and real images $\mathcal{S}$ and $\mathcal{T}$ during distillation, using a random network initialization and gradient matching at each distillation step:

$$\mathcal{L}_{\texttt{IDC}} = D\left(\nabla_\theta \ell(\theta^{\mathcal{T}}; f(\mathcal{S})), \nabla_\theta \ell(\theta^{\mathcal{T}}; \mathcal{T})\right), \tag{2}$$

where f is a multi-formation function, D is a matching metric and ℓ is classification loss.

Dataset Distillation by Bidirectional REpresentAtive Matching (DREAM+) combines distribution matching and gradient matching techniques to achieve superior performance in dataset distillation. A network is first trained on the full dataset from scratch for one epoch. This trained network is then utilized to select representative samples from the full dataset, employing methods such as KMeans [2] clustering. The loss function of DREAM+ is formulated as follows:

$$\begin{aligned}\mathcal{L}_{\texttt{DREAM+}} =& D_1\left(\nabla_\theta \ell(\theta^{\mathcal{T}}; \mathcal{S}), \nabla_\theta \ell(\theta^{\mathcal{T}}; \mathcal{T})\right) \\ &+ \alpha D_2\left(\phi_\theta(\mathcal{S}), \phi_\theta(\mathcal{T})\right).\end{aligned} \tag{3}$$

where D_1, D_2 are distance metrics and α controls the relative importance.

3. DynaDistill: Feedback-based Distillation Route Selections

The effectiveness of conventional dataset distillation is sensitive to the initialized reference models. The closer the distribution of the reference model used for distilling original data, the smaller the discrepancy between the distilled and original data. To minimize the inconsistent distributions between the reference models and original data at each distillation step, we introduce a robust and efficient approach, DynaDistill, that leverages the distribution discrepancy

between the distilled and original data as feedback to refine the trajectory selection process. Specifically, we first construct a trajectory pool consisting of N expert trajectories, denoted as $[\{\theta_t^{(0)}\}_0^T, \{\theta_t^{(1)}\}_0^T, \ldots, \{\theta_t^{(N)}\}_0^T]$. Then, we select M trajectories randomly from the trajectory pool. We further calculate the model loss $\mathcal{L}$, w.r.t. the distilled data for each selected trajectory, i.e., $[\mathcal{L}_1, \ldots, \mathcal{L}_M]$. To enable a diverse trajectory selection, we employ the temperature sampling with each distilled model loss, $\mathcal{L}_i$, as distribution, to sample a candidate trajectory for distilling data. After each distilling step, we update the trajectory pool with the latest models. In other words, we apply the model loss as a real-time feedback indicator to select the optimized trajectory to distill data. Generally, models/trajectories that exhibit superior performance metrics are accorded a higher likelihood of being selected to participate in the distillation process. Once we find a good network candidate θ_i according to the feedback synthetic dataset $\mathcal{S}_t$ at distillation step t, we use the network to distill a new synthetic dataset $\mathcal{S}_{t+1}$ recursively. Overall, we parameterize the small synthetic dataset at distillation $t+1$ by:

$$\mathcal{S}_{t+1} = \texttt{DynaDistill}(\mathcal{S}_t, \mathcal{T}, \theta_i(\mathcal{S}_t, \mathcal{T})) \tag{4}$$

where $\mathcal{S}_t$ is the synthetic dataset at the previous distillation step t, $\mathcal{T}$ is the original dataset, and θ_i is the selected network parameterized with the feedback knowledge of the synthetic dataset at distillation step t. Please see Figure 1 and Algorithm 1 for details.

Algorithm 1 `DynaDistill`

Require: $\mathcal{L}$: Distillation algorithm's corresponding loss function (`MTT`, `IDC` or `DREAM+`).
Require: $\mathcal{T}$: Real training set.
Require: P_z: Distribution of synthetic image initializations.
$\mathcal{S}_0 \sim P_{\mathcal{S}}$ ▷ Initialize distilled dataset
Randomly initialize N candidate networks $\{\theta_1, \theta_2, \ldots, \theta_N\}$
for each candidate network θ_i **do**
  Update network θ_i on real dataset $\mathcal{T}$
end for
for each distillation step t in $0 \ldots T-1$ **do**
  Randomly select M networks from candidate networks
  for each selected candidate network θ_i **do**
    $\mathcal{L}_i = \mathcal{L}(\mathcal{S}_t, \mathcal{T}; \theta_i)$
  end for
  $\mathcal{L}_i \overset{\Pr}{\sim} \{\mathcal{L}_j | j = 0, \ldots\}, \Pr(\mathcal{L}_i) \propto \frac{\exp(\mathcal{L}_i)}{\sum_j \exp(\mathcal{L}_j)}$
  ▷ Sample interested loss $\mathcal{L}_i$ with regards to probability $\Pr(\mathcal{L}_i)$
  $\mathcal{S}_{t+1} \leftarrow \text{SGD}(\mathcal{S}_t; \mathcal{L}_i)$ ▷ Update $\mathcal{S}_{t+1}$ with respect to $\mathcal{L}_i$
end for
Output: Distilled dataset $\mathcal{S}_T$

4. Experiments

4.1. Experimental Setups

Datasets We evaluate the performance of neural networks trained on condensed datasets generated by `MTT`, `IDC` and `DREAM+` as baselines. Following previous works [1, 6, 9], we conduct experiments on datasets including CIFAR-10 and CIFAR-100 datasets [7]. Specifically, CIFAR-10 contains 10 different categories, each category has 60K (50K training images and 10K test images) images of size 32×32. CIFAR-100 contains 100 different categories, each with 500 training images and 100 test images of the same size.

Network architectures Following prior dataset distillation works [1, 6, 9, 12], unless explicitly specified, we employ 3-layer convolutional networks (ConvNet-3) [11] with 128 filters and instance normalization [13], ReLU non-linearity, and 2×2 average pooling with stride 2.

Training Details We adhere to the hyperparameters predefined in their respective codebases during the synthesis of the distilled dataset which includes hyperparameters such as the number of synthetic steps, batch size, and augmentation strategy. We ensure consistent experimental settings for fair comparison across these methods and present our results accordingly. When enabling `DynaDistill` in `MTT`, `IDC`, and `DREAM+`, we set the temperature parameter to 0.5. Additionally, the number of expert trajectories selected at each distillation step is configured to be 3.

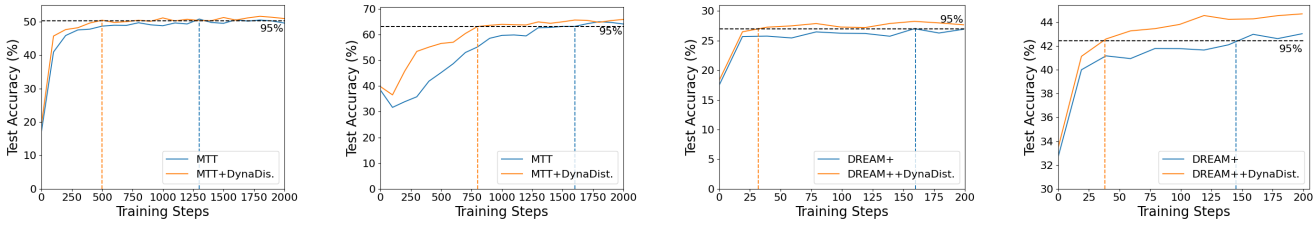
4.2. Results Analysis

The comprehensive results are summarized in Table 1. Notably, we omit the results for 10 and 50 images-per-class (IPC) in `MTT` on CIFAR-100 due to out-of-memory issues. Overall, our methods exhibit slightly superior performance compared to the original `MTT`, `IDC`, and `DREAM+` methods, except for CIFAR-10 with IPC equals 1.

We also present the test accuracy curve plotted against the training steps during the distillation phase (see Figure 2). We annotated the computational steps required for different curves to reach 95% of the peak value for the first time. Our observations indicate that both `MTT` and `DREAM+` consistently achieve better performance within the same distillation epochs. This suggests that the incorporation of `DynaDistill` facilitates enhanced dataset distillation, leading to improved test accuracy over the course of training. The results clearly illustrate that our method requires substantially fewer training steps to achieve comparable performance to the original `MTT` and `DREAM+` methods. For example, in Figure 2a, our method version attains a test accuracy of 50.49% at epoch 500, whereas the original version reaches 50.84% at 1300 epochs, which improve the training efficiency by 2.6 times. This significant reduction in training steps underscores the efficiency and effectiveness of our approach in accelerating the dataset distillation process while

Dataset	IPC	Method						Full Dataset
		MTT	MTT+DynaDis.	IDC	IDC+DynaDis.	DREAM+	DREAM++DynaDis.	
CIFAR10	1	49.63	**51.65**	49.56	**50.47**	**50.69**	49.45	84.8
	10	64.16	**65.94**	65.69	**66.29**	68.7	**69.23**	
	50	72.49	**72.89**	73.24	**73.26**	74.15	**74.45**	
CIFAR100	1	33.13	**33.48**	28.21	**29.46**	27.37	**29.01**	56.2
	10	-	-	44.73	**45.60**	43.32	**44.97**	
	50	-	-	51.9	**52.59**	51.3	**51.8**	

Table 1. Performance of dataset distillation methods on CIFAR-10 and CIFAR-100 datasets. The distilled datasets are all obtained from ConvNet-3. MTT and MTT+DynaDis. take 2000 distillation iterations. IDC, IDC+DynaDis., DREAM+, and DREAM++DynaDis. take 400 distillation iterations. Best results are marked in bold.



(a) CIFAR10, IPC=1 (b) CIFAR10, IPC=10 (c) CIFAR100, IPC=1 (d) CIFAR100, IPC=10

Figure 2. Performance comparison across varying training steps in MTT on CIFAR-10 and DREAM+ on CIFAR-100. We present the first 2000 distillation epochs in MTT and the first 200 distillation epochs DREAM+ for better visualization.

maintaining competitive performance outcomes.

Method	Evaluation Model		
	ConvNet-3	ResNet-10	DenseNet-121
MTT	33.13	25.38	25.19
MTT+DynaDis.	**33.48**	**26.43**	**26.17**
IDC	28.21	21.73	21.35
IDC+DynaDis.	**29.46**	**21.87**	**21.78**
DREAM+	27.37	21.69	21.43
DREAM++DynaDis.	**29.01**	**22.15**	**22.07**

Table 2. Performance of synthetic data learned on the CIFAR-100 dataset (IPC=1) trained on ConvNet-3 and evaluated on different network architectures.

Cross-Architecture Generalization In Table 2, we present the performance of our baseline models, including ConvNet-3 and ResNet-10 [4], as well as DenseNet-121 [5]. By incorporating DynaDistill, we ensure that the generalization of the synthetic dataset is not compromised, and our method consistently demonstrates slight improvements across different architectures.

5. Conclusion and Limitations

In this paper, we introduce a novel dataset distillation method, namely DynaDistill. By employing an autoregressive approach to select an optimal pretrained model or expert trajectory at each distillation step, DynaDistill reduces the optimization steps and enhances the performance of the distilled dataset. DynaDistill seamlessly integrates with existing surrogate objective-based frameworks, such as trajectory matching, gradient matching and distribution matching algorithms. However, our approach necessitates the simultaneous evaluation of multiple pretrained models from the model pool at each distillation step. This will escalate the demand of GPU resources.

6. Acknowledgements

This work received funding from the Federal Ministry of Education and Research of Germany (BMBF) under grant number 16DII138 (Weizenbaum-Institut) and by the German Research Foundation (DFG) under project number and 460234259 (NFDI4DataScience).

References

[1] 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. 2, 3

[2] Edward W Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, 21:768–769, 1965. 2

[3] Jiahui Geng, Zongxiong Chen, Yuandou Wang, Herbert Woisetschlaeger, Sonja Schimmler, Ruben Mayer, Zhiming Zhao, and Chunming Rong. A survey on dataset distillation: Approaches, applications and future directions. *arXiv preprint arXiv:2305.01975*, 2023. 1

[4] 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. 4

[5] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 4

[6] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. *arXiv preprint arXiv:2205.14959*, 2022. 2, 3

[7] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 3

[8] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 1

[9] Yanqing Liu, Jianyang Gu, Kai Wang, Zheng Zhu, Wei Jiang, and Yang You. Dream: Efficient dataset distillation by representative matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17314–17324, 2023. 3

[10] Yanqing Liu, Jianyang Gu, Kai Wang, Zheng Zhu, Kaipeng Zhang, Wei Jiang, and Yang You. Dream+: Efficient dataset distillation by bidirectional representative matching. *arXiv preprint arXiv:2310.15052*, 2023. 2

[11] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 3

[12] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. *arXiv preprint arXiv:2011.00050*, 2020. 3

[13] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 3

[14] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. 1, 2

[15] Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. A brief overview of chatgpt: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica*, 10(5):1122–1136, 2023. 1