DELT: A Simple Diversity-driven `EarlyLate` Training for Dataset Distillation

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Recent advances in dataset distillation have led to solutions in two main directions. The conventional *batch-to-batch* matching mechanism is ideal for small-scale datasets and includes bi-level optimization methods on models and syntheses, such as FRePo, RCIG, and RaT-BPTT, as well as other methods like distribution matching, gradient matching, and weight trajectory matching. Conversely, *batch-to-global* matching typifies decoupled methods, which are particularly advantageous for large-scale datasets. This approach has garnered substantial interest within the community, as seen in SRe^{2}L, G-VBSM, WMDD, and CDA. A primary challenge with the second approach is the lack of diversity among syntheses within each class since samples are optimized independently and the same global supervision signals are reused across different synthetic images. In this study, we propose a new `EarlyLate` training scheme to enhance the diversity of images in *batch-to-global* matching with less computation. Our approach is conceptually simple yet effective, it partitions predefined IPC samples into smaller subtasks and employs local optimizations to distill each subset into distributions from distinct phases, reducing the uniformity induced by the unified optimization process. These distilled images from the subtasks demonstrate effective generalization when applied to the entire task. We conducted extensive experiments on CIFAR, Tiny-ImageNet, ImageNet-1K, and its sub-datasets. Our empirical results demonstrate that the proposed approach significantly improves over previous state-of-the-art methods under various IPCs[1].

1 Introduction

In the era of large models and large datasets, dataset distillation has emerged as a crucial strategy to enhance training efficiency and make AI technologies more accessible and affordable for the general public. Previous approaches [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] primarily employ a *batch-to-batch* matching technique, where information like features, gradients, and trajectories from a local original data batch are used to supervise and train a corresponding batch of generated data. This method's strength lies in its ability to capture fine-grained information from the original data, as each batch's supervision signals vary. However, the downside is the necessity to repeatedly

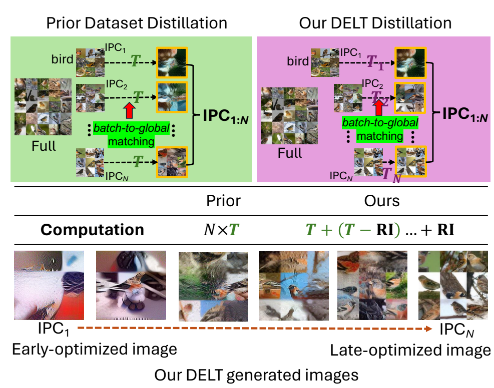

Figure 1: Distill datasets to IPC_N requires $N*T$ iterations in traditional distillation processes (left) but fewer iteration processes (right).

[1] represents n **I**mages **P**er **C**lass for the distilled dataset.

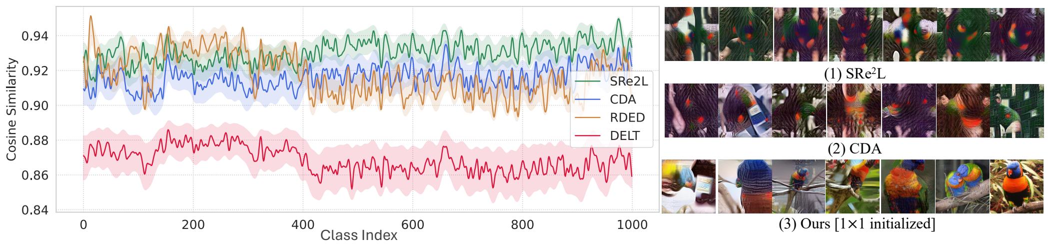

Figure 2: **Left**: Intra-class semantic cosine similarity after a pretrained ResNet-18 model on ImageNet-1K dataset, lower values are better. **Right**: Synthetic images from SRe^{2}L, CDA and our DELT.

input both original and generated data for each training iteration, which significantly increases memory usage and computational costs. Recently, a new decoupled method [11, 12, 13] has been proposed to separate the model training and data synthesis, also it leverages the *batch-to-global* matching to avoid inputting original data during distilled data generation. This solution has demonstrated great advantage on large-scale datasets like ImageNet-1K [11, 14] and ImageNet-21K [12]. However, as shown in Fig. 2 right subfigure, a significant limitation of this method is its strategy of synthesizing each data point individually, where supervision is repetitively applied across various synthetic images. For instance, SRe^{2}L[11] utilizes globally-counted layer-wise running means and variances from the pre-trained model for supervising different intra-class image synthesis. This methodology results in a pronounced lack of diversity within the same category of generated images.

To address this issue, previous studies such as G-VBSM [14] and RDED [15] have been conducted. Specifically, G-VBSM [14] introduces a framework that utilizes a diverse set of *local-match-global* matching signals derived from multiple backbones and statistical metrics, offering more precise and effective matching than the singular model. However, as the diversity of matching models grows, the overall complexity of the framework also increases, thus diminishing its conciseness. RDED [15] crops each original image into multiple patches and ranks these using realism scores generated by an observer model. Then it amalgamates every four chosen patches from previous stage into a single new image, maintaining the resolution of the original images, and produce IPC-numbered distilled images for each class. While RDED is effective for selecting and combining data, it does not enhance or optimize the visual content within the distilled dataset. Thus, the diversity and richness of information it encapsulates largely dependent on the distribution of the original dataset.

Our solution, termed the `EarlyLate` training scheme, is straightforward and also orthogonal to these prior methods: by initializing each image in the same category at a different starting point for optimization, we ensure that the final optimized results vary across images. We also use teacher-ranked real image patches to initialize the synthetic images. This prevents some images from being short-optimized and ensures they provide sufficient information. As shown in Fig. 1 of the computation comparison, our approach not only enhances intra-class diversity but also significantly reduces the computational load of the training process. Specifically, while conventional training requires T optimization iterations per image or batch, in our `EarlyLate` scheme, the first image undergoes T_1 iterations (where $T_1 = T$). Subsequent batches are processed with progressively fewer iterations, such as T_2 ($T_2 = T_1 - \text{RI}$[2]) for the next set, and so forth. The iterations for the final batch are reduced to RI which is $1/j$ of the standard count (where typically $j = 4$ or 8), meaning the total number of optimization iterations required is just about 2/3 of prior *batch-to-global* matching methods, such as SRe^{2}L and CDA. We further visualize the average cosine similarity between each sample of 50 IPCs with the associated cluster centroid within the same class on ImageNet-1K, as shown in Fig. 2 left subfigure, DELT shows significantly better diversity than other counterpart methods across all classes.

We perform extensive experiments on datasets of CIFAR-10, Tiny-ImageNet, ImageNet-1K and its subsets. On ImageNet-1K, our proposed approach achieves 66.1% under IPC 50 with ResNet-101, outperforming previous state-of-the-art RDED by 4.9%. On small-scale datasets of CIFAR-10, our approach also obtains 2.5% and 19.2% improvement over RDED and SRe^{2}L using ResNet-101.

Our main contributions in this work are as follows:

- We propose a simple yet effective `EarlyLate` training scheme for dataset distillation to enhance the intra-class diversity of synthetic images from *batch-to-global* matching.

[2] RI is the number of round iterations and will be introduced in Sec. 4.3.

- We demonstrate empirically that the proposed method can generate optimized images at different distances from their initializations, to enlarge informativeness among generations.
- We conducted extensive experiments and ablations on various datasets across different scales to prove the effectiveness of the proposed approach[3].

2 Related Work

Dataset Distillation. Dataset distillation or condensation [1] focuses on creating a compact yet representative subset from a large original dataset. This enables more efficient model training while maintaining the ability to evaluate on the original test data distribution and achieve satisfactory performance. Previous works [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] mainly designed how to better match the distribution between original data and generated data in a *batch-to-batch* manner, such as the distribution of features [6], gradients [2], or the model weight trajectories [4, 8]. The primary optimization method used is bi-level optimization [16, 17], which involves optimizing model parameters and updating images simultaneously. For instance, using gradient matching, the process can be formulated as to minimize the gradient distance:

$$\min_{\mathcal{S}\in\mathbb{R}^{N\times d}} D\left(\nabla_\theta \ell(\mathcal{S};\theta), \nabla_\theta \ell(\mathcal{T};\theta)\right) = D(\mathcal{S},\mathcal{T};\theta) \tag{1}$$

where the function $D(\cdot,\cdot)$ is defined as a distance metric such as MSE [18], θ denotes the model parameters, and $\nabla_\theta \ell(\cdot;\theta)$ represents the gradient, utilizing either the original dataset $\mathcal{T}$ or its synthetic version $\mathcal{S}$. N is the number of d-dimensional synthetic data. During distillation, the synthetic dataset $\mathcal{S}$ and model θ are updated alternatively,

$$\mathcal{S} \leftarrow \mathcal{S} - \lambda \nabla_{\mathcal{S}} D(\mathcal{S},\mathcal{T};\theta), \quad \theta \leftarrow \theta - \eta \nabla_\theta \ell(\theta;\mathcal{S}), \tag{2}$$

where λ and η are learning rates designated for $\mathcal{S}$ and θ, respectively.

Batch-to-global matching used in [11, 14, 12, 13] tracks the distribution of BN statistics derived from the original dataset for the local batch synthetic data, the formulation can be:

$$\min_{\mathcal{S}\in\mathbb{R}^{N\times d}} \Big(\sum_l \left\|\mu_l(\mathcal{S}) - \mathbf{BN}_l^{\mathrm{RM}}\right\|_2 + \sum_l \left\|\sigma_l^2(\mathcal{S}) - \mathbf{BN}_l^{\mathrm{RV}}\right\|_2\Big) \tag{3}$$

where l is the index of BN layer, $\mu_l(\mathcal{S})$ and $\sigma_l^2(\mathcal{S})$ are mean and variance. $\mathbf{BN}_l^{\mathrm{RM}}$ and $\mathbf{BN}_l^{\mathrm{RV}}$ are running mean and running variance in the pre-trained model at l-th layer, which are globally counted. Fig. 3 illustrates the difference of *batch-to-batch* and *batch-to-global* matching mechanisms, where b represents a local batch in data $\mathcal{T}$ and $\mathcal{S}$.

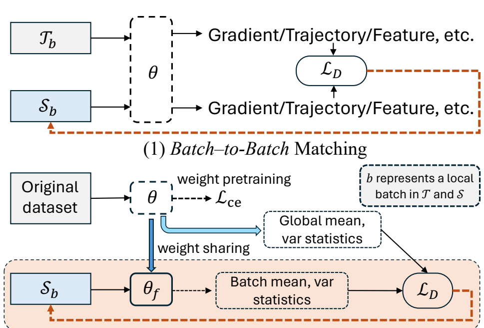

Figure 3: *Batch–to-batch* vs. *batch-to-global* matching in dataset distillation. θ_f indicates weights are pretrained and frozen in this stage.

Moreover, for the recent advances of multi-stage dataset distillation methods, MDC [10] proposes to compress multiple condensation processes into a single one by including an adaptive subset loss on top of the basic condensation loss, so that to obtain datasets with multiple sizes. PDD [9] generates multiple small batches of synthetic images, each batch is conditioned on the accumulated data from previous batches. Unlike PDD, our current synthetic batch is independent with different operation iterations and not relevant to any previous batches. D3 [19] partitions large datasets into smaller subtasks and employs locally trained experts to distill each subset into distributions. These distilled distributions from the subtasks demonstrate effective generalization when applied to the entire task.

Initialization. Weight initialization [20, 21, 22, 23] is pivotal in training neural networks, significantly influencing their optimization process. Proper initialization is essential for ensuring model convergence and mitigating issues such as gradient vanishing. Recently, weight selection [24] introduces a strategy for initializing smaller models by selecting a subset of weights from a pretrained larger model. This

[3]Our synthetic images on ImageNet-1K are available anonymously at link.

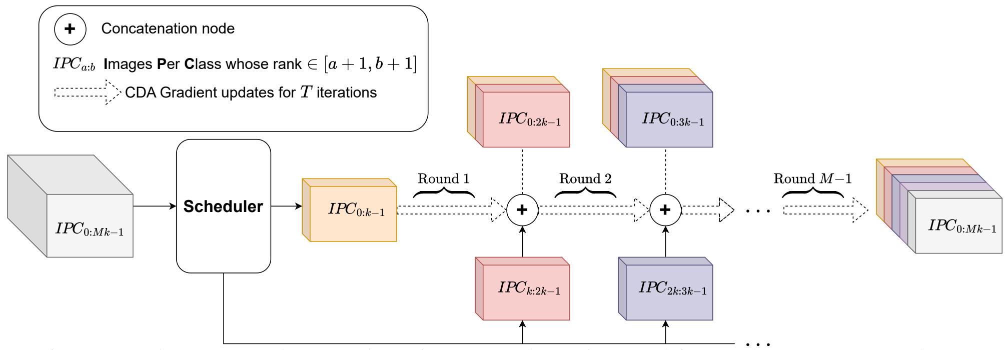

Figure 4: The proposed DELT learning procedure via a multi-round `EarlyLate` scheme.

method facilitates the transfer of learned attributes from the pretrained weights, enhancing the smaller model's performance. Weight subcloning [25] involves manipulating the pretrained model to derive a correspondingly scaled-down version with equivalent initialization. This involves two main steps: initially, it applies a neuron importance ranking to reduce the embedding dimension per layer within the pretrained model. Subsequently, it eliminates blocks from the transformer model to align with the layer count of the scaled-down network.

This work focuses on data initialization for generation processes. Few studies have examined this angle. While, PCA-K [26] appears to be the most relevant. It employs an initialization method that involves drawing samples from a distribution that accurately mirrors and is easily sampled from the training distribution. During training, it is possible to retrieve some details from the original image using the initial noisy sample, which at best provides a blurred representation of the original image.

3 Our Approach

Preliminaries. The objective of a regular dataset distillation task is to generate a compact synthetic dataset $\mathcal{S} = \{(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{y}}_1), \ldots, (\hat{\boldsymbol{x}}_{|\mathcal{S}|}, \hat{\boldsymbol{y}}_{|\mathcal{S}|})\}$ as a *student* dataset that captures a substantial amount of the information from a larger labeled dataset $\mathcal{T} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_{|\mathcal{T}|}, \boldsymbol{y}_{|\mathcal{T}|})\}$, which serves as the *teacher* dataset. Here, $\hat{\boldsymbol{y}}$ represents the soft label for the synthetic sample $\hat{\boldsymbol{x}}$, and the size of $\mathcal{S}$ is much smaller than $\mathcal{T}$, yet it retains the essential information of the original dataset $\mathcal{T}$. The learning goal using this distilled dataset is to train a post-validation model with parameters $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}_{\mathcal{S}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}), \tag{4}$$

$$\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}) = \mathbb{E}_{(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \in \mathcal{S}} \left[\ell(\phi_{\boldsymbol{\theta}_{\mathcal{S}}}(\hat{\boldsymbol{x}}), \hat{\boldsymbol{y}}; \boldsymbol{\theta}) \right], \tag{5}$$

where ℓ is a standard loss function such as soft cross-entropy and $\phi_{\boldsymbol{\theta}_{\mathcal{S}}}$ represents the model.

The primary aim of dataset distillation is to produce synthetic data that ensures minimal performance difference between models trained on the synthetic dataset $\mathcal{S}$ and those trained on the original dataset $\mathcal{T}$ using validation data V. The optimization procedure for generating $\mathcal{S}$ is given by:

$$\underset{\mathcal{S}, |\mathcal{S}|}{\arg\min} \left(\sup \left\{ \left| \ell\left(\phi_{\boldsymbol{\theta}_{\mathcal{T}}}(\boldsymbol{x}_{val}), \boldsymbol{y}_{val}\right) - \ell\left(\phi_{\boldsymbol{\theta}_{\mathcal{S}}}(\boldsymbol{x}_{val}), \boldsymbol{y}_{val}\right) \right| \right\}_{(\boldsymbol{x}_{val}, \boldsymbol{y}_{val}) \sim V} \right). \tag{6}$$

where $(\boldsymbol{x}_{val}, \boldsymbol{y}_{val})$ are the sample and label pairs in the validation set of the real dataset $\mathcal{T}$. The learning task then focuses on the <data, label> pairs within $\mathcal{S}$, maintaining a balanced representation of distilled data across each class.

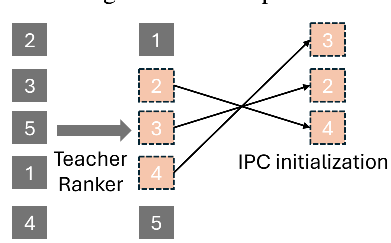

Figure 5: Selection criteria with a teach ranker.

Initialization. Previous dataset distillation methods [11, 14, 12] on large-scale datasets like ImageNet-1K and 21K employ Gaussian noise by default for data initialization in the synthesis phase. However, Gaussian noise is random and lacks any semantic information. Intuitively, using real images provide a more meaningful and structured starting point, and this structured start can lead to quicker convergence during optimization because the initial data already contains useful features and patterns that are closer to the target distribution, which further enhances realism, quality, and generalization of the synthesized images. As shown in Fig. 2 right subfigure, our generated images exhibit both diversity and a high degree of realism in some cases.

Selection Criteria. Here, we introduce how to select real image patches to initialize the synthetic images. In our final syntheses, a significant fraction of our data has been subject to limited optimization iterations, making effective initialization crucial. A proper initialization also dramatically minimizes the overall computational load required for the updating on data. Prior approach [15] has demonstrated that choosing representative data patches from the original dataset without training can yield favorable performance without any additional training. Our observation, however, underscores that applying iterative refinement to original patches can lead to markedly improved results. As illustrated in Fig. 1, our selection criterion is based on a pretrained teacher model as a ranker, we calculate all patches' probabilities and sort them as the initialization pool. Then, we choose lowest, medium, or highest probability patches as the initialization for our optimization.

Diversity-driven IPC Concatenation Training. As shown in Fig. 4, to further emphasize diversity and avoid potential distribution bias from initialization, we optimize the initialized images starting from different points. The motivation behind this design is that different data samples require varying numbers of iterations to converge which is similar to the early stopping idea [27]. Importantly, as images become easier to predict with more updates by class labels, training primarily on easy data points can hinder model generalization. Therefore, our method enhances generalization by generating data samples with varying difficulty levels, acting as a regularizer by limiting the optimization process to a smaller volume of image pixel space. Previous work [28] studies how to perform early stopping training on different layers' weights of the model with progressive retraining to mitigate noisy labels. We are pioneering to study how to leverage early-late training when optimizing data. Moreover, we improve the efficiency of our approach by performing gradient updates in a single scan. Initially, we conduct a single gradient loop, continually introducing new data for distillation by concatenating them at different time stamps. Consequently, the M batch receives the synthetic images of all preceding batches, $IPC_{0:Mk-1}$, as final generations. This process can be simplified as follows:

$$IPC_{0:Mk-1} = [\underbrace{\underbrace{\underbrace{\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_{k-1}}_{IPC_{0:k-1}}, \ldots}_{\ldots}, \hat{\boldsymbol{x}}_{Mk-1}}_{IPC_{0:Mk-1}}] \tag{7}$$

where $[\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_{Mk-1}]$ refers to the concatenation of the generated image. M is the number of batches, k is the number of generated images in each batch. We train these different batches at different starting points, each batch goes through a completed learning phase, but the total number of iterations varies. Then, the multiple IPCs of $\hat{\boldsymbol{x}}$ are concatenated into a simple batch. Because of its early-late training property, we refer to this simple training scheme as `EarlyLate` training.

Training Procedure. As illustrated in Fig. 4, our learning procedure is extremely simple using an incremental learning process: We split the total IPCs to be learned into multiple batches. The training begins with the first batch. Following a predefined number of iterations, the second batch commences its iterative training, and this process continues sequentially with subsequent batches. *Batch-to-global* matching algorithm [12] of Eq. 3 has been utilized between each round.

4 Experiments

4.1 Datasets and Results Details

We first run DELT on five standard benchmark tests including CIFAR-10 (10 classes) [29], Tiny-ImageNet (200 classes) [30], ImageNet-1K (1,000 classes) [31] and it variants of ImageNette (10 classes) [32], and ImageNet-100 (100 classes) [33] with performances reported in Table 1 and Table 2. The evaluation protocol is following prior works [15, 11]. We compare DELT to six baseline dataset distillation algorithms including Matching Training Trajectories (MTT) [4], Improved Distribution Matching (IDM) [34], TrajEctory Matching with Constant Memory (TESLA) [8], Squeeze-Recover-Relabel (SRe^{2}L) [11], Difficulty-Aligned Trajectory-Matching (DATM) [35], Realistic-Diverse-Efficient Dataset Distillation (RDED) [15]. Following previous dataset distillation methods [2, 15, 11], we use ConvNet [36], ResNet-18/ResNet-101 [37], EfficientNet-B0 [38], MobileNet-V2 [39], MnasNet1_3 [40], and RegNet-Y-8GF [41], as our backbone for training or post-validation. All our experiments are conducted on 4 NVIDIA RTX 4090 GPUs.

		ResNet-18			ResNet-101			MobileNet-v2
Dataset	IPC	SRe^{2}L [11]	RDED [15]	Ours	SRe^{2}L [11]	RDED [15]	Ours	Ours
CIFAR-10	1	16.6 ± 0.9	22.9 ± 0.4	**24.0 ± 0.8**	13.7 ± 0.2	18.7 ± 0.1	**20.4 ± 1.0**	**20.2 ± 0.4**
	10	29.3 ± 0.5	37.1 ± 0.3	**43.0 ± 0.9**	24.3 ± 0.6	33.7 ± 0.3	**37.4 ± 1.2**	**29.3 ± 0.3**
	50	45.0 ± 0.7	62.1 ± 0.1	**64.9 ± 0.9**	34.9 ± 0.1	51.6 ± 0.4	**54.1 ± 0.8**	**42.9 ± 2.2**
ImageNette	1	19.1 ± 1.1	**35.8 ± 1.0**	24.1 ± 1.8	15.8 ± 0.6	**25.1 ± 2.7**	19.4 ± 1.7	**19.1 ± 1.0**
	10	29.4 ± 3.0	61.4 ± 0.4	**66.0 ± 1.4**	23.4 ± 0.8	54.0 ± 0.4	**55.4 ± 6.2**	**64.7 ± 1.4**
	50	40.9 ± 0.3	80.4 ± 0.4	**88.2 ± 1.2**	36.5 ± 0.7	75.0 ± 1.2	**83.3 ± 1.1**	**85.7 ± 0.4**
Tiny-ImageNet	1	2.62 ± 0.1	**9.7 ± 0.4**	9.3 ± 0.5	1.9 ± 0.1	3.8 ± 0.1	**5.6 ± 1.0**	**3.5 ± 0.5**
	10	16.1 ± 0.2	41.9 ± 0.2	**43.0 ± 0.1**	14.6 ± 1.1	22.9 ± 3.3	**42.8 ± 0.9**	**26.5 ± 0.5**
	50	41.1 ± 0.4	**58.2 ± 0.1**	55.7 ± 0.5	42.5 ± 0.2	41.2 ± 0.4	**58.5 ± 0.3**	**51.3 ± 0.5**
ImageNet-100	10	9.5 ± 0.4	**36.0 ± 0.3**	28.2 ± 1.5	6.4 ± 0.1	**33.9 ± 0.1**	22.4 ± 3.3	**15.8 ± 0.2**
	50	27.0 ± 0.4	61.6 ± 0.1	**67.9 ± 0.6**	25.7 ± 0.3	66.0 ± 0.6	**70.8 ± 2.3**	**55.0 ± 1.8**
	100	-	74.5 ± 0.4	**75.1 ± 0.2**	-	73.5 ± 0.8	**77.6 ± 1.8**	**76.7 ± 0.3**
ImageNet-1K	10	21.3 ± 0.6	42.0 ± 0.1	**45.8 ± 0.1**	30.9 ± 0.1	48.3 ± 1.0	**48.5 ± 1.6**	**35.1 ± 0.5**
	50	46.8 ± 0.2	56.5 ± 0.1	**59.2 ± 0.4**	60.8 ± 0.5	61.2 ± 0.4	**66.1 ± 0.5**	**56.2 ± 0.3**
	100	52.8 ± 0.3	59.8 ± 0.1	**62.4 ± 0.2**	62.8 ± 0.2	-	**67.6 ± 0.3**	**58.9 ± 0.3**

Table 1: Comparison with SOTA dataset distillation methods using relatively large-scale backbones on five benchmarks across different scales. MobileNet-v2 is modified to match the low resolutions of CIFAR-10 and Tiny-ImageNet following [42]. Due to the table space limitation, some other methods that are weaker than RDED are not listed, such as CDA and G-VBSM. Since IPC 1 is not applicable to use `EarlyLate` strategy and the single image in each class is optimized with a constant iteration.

		ConvNet					
Dataset	IPC	MTT [4]	IDM [34]	TESLA [8]	DATM [35]	RDED [15]	Ours
ImageNette	1	**47.7 ± 0.9**	-	-	-	33.8 ± 0.8	29.8 ± 1.4
	10	63.0 ± 1.3	-	-	-	**63.2 ± 0.7**	51.7 ± 1.2
	50	-	-	-	-	83.8 ± 0.2	**84.5 ± 0.4**
Tiny-ImageNet	1	8.8 ± 0.3	10.1 ± 0.2	-	**17.1 ± 0.3**	12.0 ± 0.1	12.4 ± 0.8
	10	23.2 ± 0.2	21.9 ± 0.3	-	31.1 ± 0.3	39.6 ± 0.1	**40.0 ± 0.4**
	50	28.0 ± 0.3	27.7 ± 0.3	-	39.7 ± 0.3	47.6 ± 0.2	**48.6 ± 0.2**
ImageNet-100	10	-	17.1 ± 0.6	-	-	**29.6 ± 0.1**	24.7 ± 1.5
	50	-	26.3 ± 0.4	-	-	50.2 ± 0.2	**51.9 ± 1.1**
	100	-	-	-	-	58.6 ± 0.4	**61.5 ± 0.5**
ImageNet-1K	1	-	-	7.7 ± 0.2	-	6.4 ± 0.1	**8.8 ± 0.5**
	10	-	-	17.8 ± 1.3	-	20.4 ± 0.1	**31.3 ± 0.8**
	50	-	-	27.9 ± 1.2	-	38.4 ± 0.2	**41.7 ± 0.1**

Table 2: Comparison with SOTA dataset distillation methods using small-scale backbone architecture on four benchmark datasets. Following [4, 34, 15], Conv-3 is used for CIFAR-10, Conv-4 for Tiny-ImageNet and ImageNet-1K, Conv-5 for ImageNette, and Conv-6 for ImageNet-100 and ImageNet-1K. Entries marked with "-" are missing due to scalability issue.

As shown in Table 1, our approach establishes the new state-of-the-art accuracy in 13 out of 15 of the configurations on five datasets from small-scale CIFAR-10 to large-scale ImageNet-1K using relatively large backbone architecture of ResNet-101, in many cases with significant margins of improvement. The results using small-scale architecture ConvNet are shown in Table 2, our approach also achieves the state-of-the-art accuracy in 8 out of 12 of the configurations on four datasets.

4.2 Cross-architecture generalization

An important characteristic of distilled datasets is their effectiveness in generalizing to novel training architectures. In this context, we assess the transferability of DELT's distilled datasets tailored for ImageNet-1K with 10 images per class. Following previous studies [11, 15], we test our models using five distinct architectures: ResNet-18 [37], MobileNet-V2 [39], MnasNet1_3 [40], EfficientNet-B0 [38], and RegNet-Y-8GF [41]. As shown in Table 4, our proposed approach demonstrates significant better performance than other competitive methods on all these architectures.

4.3 Ablation Study

Mosaic splicing pattern. Mosaic stitching method [43] in RDED selects four crops from the train set as the optimal hyper-parameter, and puts the contents of the four crops into a synthetic image that is directly used for post-validation. In this work, considering that we use different difficulty levels of

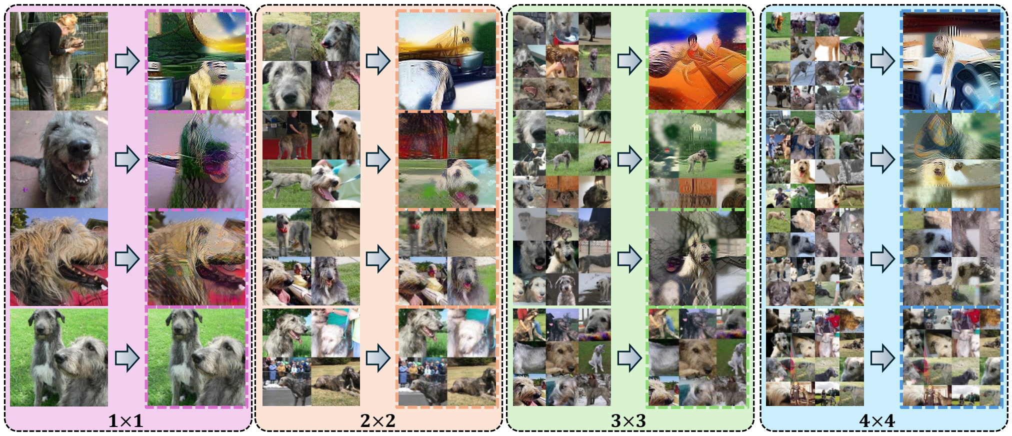

Figure 6: Mosaic splicing patterns on ImageNet-1K using real image patches as the initialization. In each block, the left column is the starting real image initialized samples and right is the final optimized syntheses. From top to bottom are images generated by early training and late training.

selection for initialization, we examine different strategies of the Mosaic splicing patterns, including 1×1, 2×2, 3×3, 4×4, and 5×5 patches, as illustrated in Fig. 11. The ablation results are shown in Table 3, it can be observed that 1×1 achieves the best accuracy.

Initialization. We examine how different initialization strategies affect final performance, including: choosing lowest probability crops, medium probability crops and highest probability crops. Our results are shown in Table 3. Overall, the performance gap between different strategies is not significant, and selecting the medium probability crops as the initialization achieves the best accuracy.

Optimization iterations. We examine two types of optimization iterations: maximum iteration (MI) for the earliest batch training and round iteration (RI). MI presents the number of optimization iterations that the earliest batch goes through. RI represents the number of iterations used for each round in Fig. 4. It essentially indicates the iteration gap between the optimization of two adjacent batches. As shown in Table 3, we test MI values of 1K, 2K, and 4K, using 500 and 1K iterations for each RI. Note that when MI is set to 1K, it is not feasible to use 1K as RI. The results show that 4K (same as [11, 12]) MI and 500 RI achieves the best accuracy.

Early-only vs. EarlyLate. Early-only is equivalent to using constant MI to optimize each image. The method will transform to baseline *batch-to-global* matching of CDA [12] + real image initialization. Our results in Table 3 clearly show that the `EarlyLate` training bring a significant improvement on final performance. More importantly, this strategy is the key factor in enhancing generation diversity.

Real image stitching vs. Minimax diffusion vs. Ours. We further compare the performance of our approach with real image stitching [15] and diffusion generation [44]. The results are presented in Table 3d. While the first two methods produce more realistic images, each image contains limited information. In contrast, our method achieves the best final performance.

4.4 Computational Analysis

For image optimization-based methods like SRe^{2}L and CDA, the total computational cost is calculated as $N \times T$, where N is the MI. In our `EarlyLate` scheme, the first batch images undergo T_1 iterations (where $T_1 = T$). Subsequent batches are processed with progressively fewer iterations, such as T_2 ($T_2 = T_1 - \text{RI}$) for the next set, and so forth. The iterations for the final batch are reduced to RI which is $1/j$ of the standard count (where $j = 4$ or 8 in our ablation), the total number of our optimization iterations required is $N \times T - \frac{j(j-1)}{2}\text{RI}$, which is roughly 2/3 of prior *batch-to-global* matching methods. Our real time consumptions for data generation are shown in Table 5, note that the smaller the dataset like CIFAR, the more time is spent on loading and processing the data, rather than training.

4.5 Visualization of DELT

Fig. 7 illustrates a comprehensive visual comparison between randomly selected synthetic images from our distilled dataset and those from the real image patches [15], MinimaxDiffusion [44], MTT [4], IDC [45], SRe^{2}L [11], SCDD [46], CDA [12] and G-VBSM [14] distilled data. It can be observed that

Table 3: **Ablation experiments** on various aspects of our framework with ResNet-18 on ImageNet-1K.

# Patches	Top 1 acc
1×1	**57.57**
2×2	56.92
3×3	56.62
4×4	56.71
5×5	56.51

(a) **Number of patches**. Ablation on initializing different numbers of scoring patches. Results are from ResNet-18 on ImageNet-1K for 500 iterations to synthesize 50 IPCs.

Selection criteria	Top 1 acc
Lowest probability	57.55
Medium probability	**57.67**
Highest probability	57.03

(b) **Selection criteria**. Initializing 1×1 images selected according to teacher model's probability

Iterations	Round Iterations 500	Round Iterations 1K
1K	44.87	n/a
2K	45.61	44.40
4K	**46.42**	44.66

(c) **Round Iterations**. Top-1 acc. of our method for IPC 10 using different round iterations with ResNet-18.

Dataset	CDA [12] + Our init.	Ours
ImageNet-1K	43.5	**45.8**
Tiny-ImageNet	42.2	**43.0**
CIFAR-10	39.4	**43.0**

(d) Ablation on init. and `EarlyLate` under IPC 10.

IPC	RDED [15]	MinimaxDiffusion [44]	Ours
10	42.0	44.3	**45.8**
50	56.5	58.6	**59.2**

(e) Comparison with real and diffusion generated data.

Table 4: **Cross-architecture generalization**. Results are evaluated on IPC 10.

Recover \Validation		ResNet-18	EfficientNet-B0	MobileNet-V2	MnasNet1_3	RegNet-Y-8GF
ResNet-18	SRe^{2}L [11]	41.9	41.9	33.1	39.3	51.5
	CDA [12]	42.2	43.9	34.2	39.7	52.9
	G-VBSM [14]	41.4	42.6	33.5	40.1	52.2
	RDED [15]	42.3	42.8	34.4	40.0	54.8
	Ours	**46.4** (+4.1)	**47.1** (+4.3)	**36.1** (+1.7)	**40.7** (+0.7)	**57.5** (+2.7)

Table 5: **Actual computational consumption and analysis** (hours under IPC 50) in data synthesis with image optimization-based methods on a single NVIDIA 4090 GPU. "RI" represents *round iterations*. A total 4K iterations are used for all methods and datasets to ensure fair comparisons.

Method	Dataset (hours) ImageNet-1K	Tiny-ImageNet	CIFAR-10
G-VBSM [14]	114.1	5.5	0.195
SRe^{2}L [11]	29.0	5.0	0.084
CDA [12]	29.0	5.0	0.084
Ours (RI = 500)	**17.6** (↓39.3%)	**3.4** (↓32.0%)	**0.083** (↓1.1%)
Ours (RI = 1K)	**18.8** (↓35.2%)	**3.6** (↓28.0%)	**0.084** (↓0.0%)

the images generated by each method have their own characteristics. MinimaxDiffusion leverages the diffusion model to synthesize images which is close to the real ones. However, as in our above ablation, both real and diffusion-generated data are inferior to ours. MTT results show noticeable artifacts and distortions, the objects in all images are located in the middle of the generations, the diversity is limited. IDC results also show distorted and less recognizable dog images, but diversity is increased. SRe^{2}L exhibits some dog features but with significant distortions and similar simple background. SCDD shows more recognizable dog features but still the color is simple and monochromatic, the same situation happens in CDA. G-VBSM shows more colorful patterns, possibly due to recovery from multiple different networks, but all generations are in the same pattern and the diversity is not large. Our approach's synthetic images exhibit a higher degree of diversity, including both compressed distorted images from long-optimized initializations and clear, recognizable dog images from short-optimized initializations, a unique capability not present in other methods.

4.6 Application I: Data-free Network Pruning

Our distilled dataset acts as a multifunctional training tool and boosts the adaptability for diverse downstream applications. We validate its utility in the scenario of data-free network pruning [47]. Table 6 shows the applicability of our dataset in this task when pruning 50% weights, where it significantly surpasses previous methods such as SRe^{2}L and RDED under IPC 10 and 50.

Figure 7: Distilled dataset visualization compared with other image optimization-based methods.

Table 6: Accuracy of data-free network pruning using slimming [48] on VGG11-BN [49].

	SRe^{2}L [11]	RDED [15]	Ours
IPC 10	12.5	13.2	**17.9**$_{(+4.7)}$
IPC 50	31.7	42.8	**44.8**$_{(+2.0)}$

4.7 Application II: Continual Learning

We examine the effectiveness of DELT generated images in the continual learning scenario. Following the setup in prior studies [11, 6], we perform 100-step class-incremental experiments on ImageNet-1K, comparing our results with the baselines G-VBSM and SRe^{2}L. As shown in Fig. 8, our DELT distilled dataset significantly outperforms G-VBSM, with an average improvement of about 10% in 100-step class-incremental learning task. This highlights the significant benefits of deploying DELT, particularly in mitigating the challenges of continual learning.

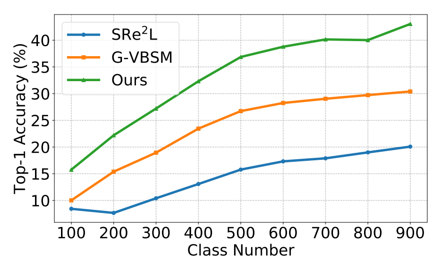

Figure 8: Continual learning results.

5 Conclusion

We have introduced a new training strategy, `EarlyLate`, to improve image diversity in *batch-to-global* matching scenarios for dataset distillation. The proposed approach organizes predefined IPC samples into smaller, manageable subtasks and utilizes local optimizations. This strategy helps in refining each subset into distributions characteristic of different phases, thereby mitigating the homogeneity typically caused by a singular optimization process. The images refined through this method exhibit robust generalization across the entire task. We have extensively evaluated this approach on CIFAR-10 and 100, Tiny-ImageNet, ImageNet-1K, and its variants. Our empirical findings indicate that our approach significantly outperforms prior state-of-the-art methods across various IPC configurations.

Limitations. Our method effectively avoids the issue of insufficient data diversity generated by *batch-to-global* methods and reduces the computational cost of the generation process. However, there is still a performance gap when training the model on our generated data compared to training on the original dataset. Additionally, our short-optimized data exhibits similar semantic information to the original images, which may potentially leak the privacy of the original dataset.

References

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

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

[3] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. *Advances in Neural Information Processing Systems*, 35:9813–9827, 2022.

[4] 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.

[5] 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.

[6] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023*, 2023.

[7] Songhua Liu, Kai Wang, Xingyi Yang, Jingwen Ye, and Xinchao Wang. Dataset distillation via factorization. *Advances in Neural Information Processing Systems*, 35:1100–1113, 2022.

[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.

[9] Xuxi Chen, Yu Yang, Zhangyang Wang, and Baharan Mirzasoleiman. Data distillation can be like vodka: Distilling more times for better quality. In *The Twelfth International Conference on Learning Representations*, 2024.

[10] Yang He, Lingao Xiao, Joey Tianyi Zhou, and Ivor Tsang. Multisize dataset condensation. *ICLR*, 2024.

[11] Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. In *NeurIPS*, 2023.

[12] Zeyuan Yin and Zhiqiang Shen. Dataset distillation in large data era. *arXiv preprint arXiv:2311.18838*, 2023.

[13] Haoyang Liu, Tiancheng Xing, Luwei Li, Vibhu Dalal, Jingrui He, and Haohan Wang. Dataset distillation via the wasserstein metric. *arXiv preprint arXiv:2311.18531*, 2023.

[14] Shitong Shao, Zeyuan Yin, Muxin Zhou, Xindong Zhang, and Zhiqiang Shen. Generalized large-scale data condensation via various backbone and statistical matching. In *CVPR*, 2024.

[15] Peng Sun, Bei Shi, Daiwei Yu, and Tao Lin. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. In *CVPR*, 2024.

[16] Risheng Liu, Jiaxin Gao, Jin Zhang, Deyu Meng, and Zhouchen Lin. Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):10045–10067, 2021.

[17] Yihua Zhang, Prashant Khanduri, Ioannis Tsaknakis, Yuguang Yao, Mingyi Hong, and Sijia Liu. An introduction to bi-level optimization: Foundations and applications in signal processing and machine learning. *arXiv preprint arXiv:2308.00788*, 2023.

[18] Zhou Wang and Alan C Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117, 2009.

[19] Tian Qin, Zhiwei Deng, and David Alvarez-Melis. Distributional dataset distillation with subtask decomposition. *arXiv preprint arXiv:2403.00999*, 2024.

[20] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[22] Dmytro Mishkin and Jiri Matas. All you need is a good init. *arXiv preprint arXiv:1511.06422*, 2015.

[23] Trieu H Trinh, Minh-Thang Luong, and Quoc V Le. Selfie: Self-supervised pretraining for image embedding. *arXiv preprint arXiv:1906.02940*, 2019.

[24] Zhiqiu Xu, Yanjie Chen, Kirill Vishniakov, Yida Yin, Zhiqiang Shen, Trevor Darrell, Lingjie Liu, and Zhuang Liu. Initializing models with larger ones. In *The Twelfth International Conference on Learning Representations*, 2024.

[25] Mohammad Samragh, Mehrdad Farajtabar, Sachin Mehta, Raviteja Vemulapalli, Fartash Faghri, Devang Naik, Oncel Tuzel, and Mohammad Rastegari. Weight subcloning: direct initialization of transformers using larger pretrained ones. *arXiv preprint arXiv:2312.09299*, 2023.

[26] Jeffrey Zhang, Shao-Yu Chang, Kedan Li, and David Forsyth. Preserving image properties through initializations in diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5242–5250, 2024.

[27] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 2002.

[28] Yingbin Bai, Erkun Yang, Bo Han, Yanhua Yang, Jiatong Li, Yinian Mao, Gang Niu, and Tongliang Liu. Understanding and improving early stopping for learning with noisy labels. *Advances in Neural Information Processing Systems*, 34:24392–24403, 2021.

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

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

[31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[32] Fastai. Fastai/imagenette: A smaller subset of 10 easily classified classes from imagenet, and a little more french.

[33] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020.

[34] Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7856–7865, 2023.

[35] Ziyao Guo, Kai Wang, George Cazenavette, Hui Li, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching. In *The Twelfth International Conference on Learning Representations*, 2024.

[36] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4367–4375, 2018.

[37] 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.

[38] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[39] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[40] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2820–2828, 2019.

[41] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10428–10436, 2020.

[42] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11953–11962, 2022.

[43] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[44] Jianyang Gu, Saeed Vahidian, Vyacheslav Kungurtsev, Haonan Wang, Wei Jiang, Yang You, and Yiran Chen. Efficient dataset distillation via minimax diffusion. In *CVPR*, 2024.

[45] 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 *Proceedings of the 39th International Conference on Machine Learning*, 2022.

[46] Muxin Zhou, Zeyuan Yin, Shitong Shao, and Zhiqiang Shen. Self-supervised dataset distillation: A good compression is all you need. *arXiv preprint arXiv:2404.07976*, 2024.

[47] Suraj Srinivas and R Venkatesh Babu. Data-free parameter pruning for deep neural networks. *arXiv preprint arXiv:1507.06149*, 2015.

[48] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pages 2736–2744, 2017.

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

[50] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[51] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

Appendix

A Broader Impacts

Our dataset distillation framework can significantly reduce the computational resources required for training machine learning models. This leads to lower energy consumption and cost, making AI more accessible and sustainable. By generating smaller, more manageable datasets, researchers and developers can iterate and experiment more quickly, accelerating the pace of innovation in various AI applications. However, condensed datasets might inadvertently amplify biases present in the original data. If the distillation process does not adequately address bias, it could lead to unfair or discriminatory AI systems. Also, simplifying datasets may lead to a loss of important nuances and context, potentially degrading the performance of models in real-world applications where such details are crucial. Moreover, the models may overfit to condensed data, indicating that models trained on distilled datasets might perform well on the condensed data but poorly on more diverse real-world data, limiting their generalizability and robustness.

B Training Details

Table 7: Hyper-parameter settings.

(a) Validation settings

config	value
optimizer	AdamW
base learning rate	0.001 (all) 0.0025 (MobileNet-v2)
weight decay	0.01
batch size	100 (IPC50) 50 (IPC10) 10 (IPC1)
learning rate schedule	cosine decay
training epoch	300
augmentation	RandAugment RandomResizedCrop RandomHorizontalFlip

(b) Recovery settings

config	value
α_{BN}	0.01
optimizer	Adam
base learning rate	0.25
momentum	β_1, β_2 = 0.5, 0.9
batch size	100
learning rate schedule	cosine decay
recovery iteration	4,000
round iteration	500 [IPC 10, 50, 100]
initialization	top medium
augmentation	RandomResizedCrop

(c) Dataset-specific settings in recovery

config	CIFAR10	Tiny-ImageNet	ImageNette	ImageNet-100	ImageNet-1K
RandAugment (`m`)	5	4	6	6	6
RandAugment (`n`)	4	3	2	2	2
RandAugment (`mstd`)	1.0	1.0	1.0	1.0	1.0
IPC1 Recovery Iterations	2K (R18) 3K (R101) 2K (MobileNet) -	500 (R18) 500 (R101) 500 (MobileNet) 1K (Conv4)	1K (R18) 1K (R101) 2K (MobileNet) 4K (Conv5)	- - - -	3K (Conv4) - - -

For reproducibility, we provide all our hyper-parameter settings used in our experiments in Table 7, we outline such details below.

Squeezing and Pre-trained models. Following the previous works [11, 12, 15], we use the official PyTorch [50] pre-trained ResNet-18 model for ImageNet-1K, and we use the same official Torchvision [50] code to produce our pre-trained models, ResNet-18 and ConvNet, for the other datasets.

Ranking. A crucial part of our method is initialization, we simply use ResNet-18 pre-trained models to rank and select the top-medium images as initialization for all our datasets, except for ImageNet-100 where we simply extracted the top-medium images based on the rankings of the original ImageNet-1K.

Recovery. For our synthesis, we provide the details of the general hyper-parameters used for different datasets, including ImageNet-1K, ImageNet-100, ImageNette, Tiny-ImageNet, and CIFAR10, in Table 7b. Because synthesizing a single image per class, i.e., IPC 1, is quite special as we cannot use

Figure 9: Synthetic image visualizations on Tiny-ImageNet generated by our DELT.

rounds, we apply different numbers of iterations based on both the dataset scale and the validation teacher model as outlined in Table 7c.

Validation. This includes both the soft-label generation, Relabel in SRe^{2}L, and evaluation, or post-training. We outline such details in Table 7a. We use `timm`'s version of RandAugment [51] with different settings depending on the synthesized dataset being validated as outlined in Table 7c.

C More Visualizations

We provide more visualizations on synthetic Tiny-ImageNet, ImageNette and CIFAR-10 datasets. In each figure, each column represents a different class, with images progressing from long optimization at the top to short optimization at the bottom.

Figure 10: Synthetic image visualizations on ImageNette generated by our DELT.

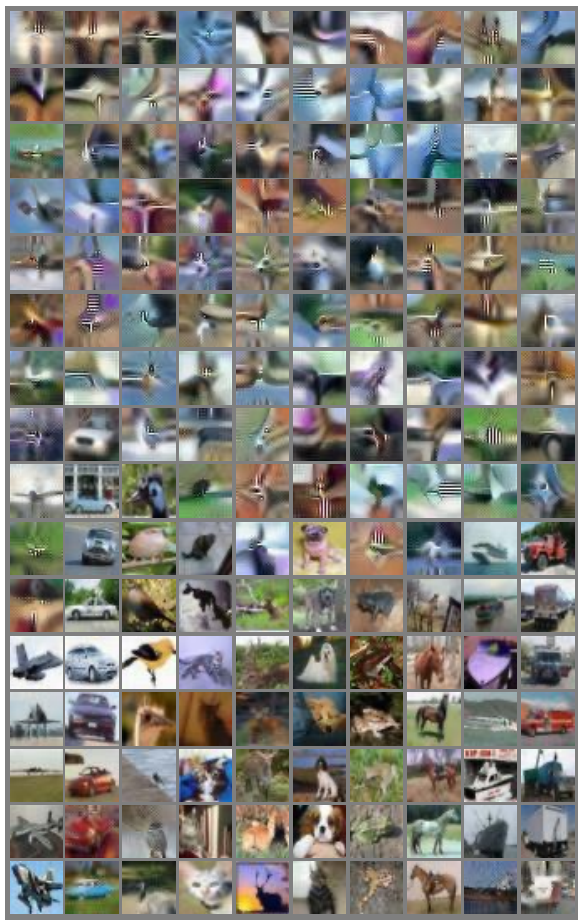

Figure 11: Synthetic image visualizations on CIFAR-10 generated by our DELT.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have clearly stated the contributions and scope of this paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations have been discussed in the conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical assumptions.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided all the experimental details to reproduce the results. Code is also available in the supplemental materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have included the code in the supplemental materials and shared the data link anonymously in the main paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have specified all the training and test details to understand the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We have performed our experiments three times for each to provide the mean and variance accuracy suitably and correctly in our tables.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided the details of computer resources in the experimental section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This research conducted in the paper conforms in every respect with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed both potential positive societal impacts and negative societal impacts in Sec. A.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We believe this paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited all papers and credited all code we utilized in this work.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our code has been included in the supplemental materials and is well documented, we have also shared the synthetic data in the main paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.