Distributed Boosting: An Enhancing Method on Dataset Distillation

Abstract

Dataset Distillation (DD) is a technique for synthesizing smaller, compressed datasets from large original datasets while retaining essential information to maintain efficacy. Efficient DD is a current research focus among scholars. Squeeze, Recover and Relabel (SRe^{2}L) and Adversarial Prediction Matching (APM) are two advanced and efficient DD methods, yet their performance is moderate with lower volumes of distilled data. This paper proposes an ingenious improvement method, Distributed Boosting (DB), capable of significantly enhancing the performance of these two algorithms at low distillation volumes, leading to DB-SRe^{2}L and DB-APM. Specifically, DB is divided into three stages: Distribute&Encapsulate, Distill, and Integrate&Mix-relabel. DB-SRe^{2}L, compared to SRe^{2}L, demonstrates performance improvements of 25.2%, 26.9%, and 26.2% on full 224×224 ImageNet-1k at Images Per Class (IPC) 10, CIFAR-10 at IPC 10, and CIFAR-10 at IPC 50, respectively. Meanwhile, DB-APM, in comparison to APM, exhibits performance enhancements of 21.2% and 20.9% on CIFAR-10 at IPC 10, CIFAR-100 at IPC 1, respectively. Additionally, we provide a theoretical proof of convergence for DB. To the best of our knowledge, DB is the first method suitable for distributed parallel computing scenarios.

1 INTRODUCTION

In recent years, Dataset Distillation (DD) Wang et al. [2018] has attracted considerable attention in the realms of computer vision Cazenavette et al. [2022], Loo et al. [2022], Cui et al. [2023], Yin et al. [2023] and natural language processing Sucholutsky and Schonlau [2021], Maekawa et al. [2023]. This endeavor seeks to refine the methodology of condensing voluminous datasets into more compact yet epitomizing subsets, maintaining quintessential attributes and features, thereby facilitating models to assimilate knowledge as efficaciously from the distilled datasets as they would from the expansive original datasets. The significance of data condensation in both scholarly and practical contexts is paramount, given its vital contribution to the proficient management and manipulation of copious data across a spectrum of disciplines. The profoundly condensed distilled datasets, encompassing substantial informative value, exhibit the potential for expeditious model training. Consequently, these datasets have emerged as a popular selection for diverse downstream applications, including Federated Learning Hu et al. [2022], Sangermano et al. [2022], Continual Learning Masarczyk and Tautkute [2020], Zhao and Bilen [2021] and Neural Architecture Search Such et al. [2020].

Through the implementation of intricate algorithms, such as meta-model matching Wang et al. [2018], Zhou et al. [2022b], gradient matching Kim et al. [2022], Zhao et al. [2020], Lee et al. [2022], distribution matching Wang et al. [2022], Zhao and Bilen [2023], and trajectory matching Cazenavette et al. [2022], Cui et al. [2023], significant strides have been made in data condensation. These solutions primarily aim to extract smaller datasets and have exhibited commendable performance in instances such as CIFAR, small images Le and Yang [2015], downscaled low-resolution images Chrabaszcz et al. [2017], or a subset of ImageNet Kim et al. [2022]. Nevertheless, all these excellent methods still incur substantial training overhead burdens on the entire 224×224 ImageNet-1k dataset Shao et al. [2023].

In an effort to mitigate issues pertaining to inefficiency, a cohort of algorithms demonstrating enhanced efficiency has been introduced recently Yin et al. [2023], Chen et al. [2023], Shao et al. [2023]. The first instance of DD on the complete 224×224 ImageNet-1k dataset, Squeeze, Recover and Relabel (SRe^{2}L) Yin et al. [2023], was proposed, wherein a Top-1 validation accuracy of 21.3% was realized using a ResNet18 at Image Per Class (IPC) 10. IPC refers to the number of images per class obtained by DD in the context of image classification tasks. By decoupling bi-level optimization, SRe^{2}L boasts a 16× speed enhancement and a 13.6% performance improvement over the state-of-the-art approach, TESLA Cui et al. [2023], which operates on a low-resolution version of ImageNet-1k. Meanwhile, Adversarial Prediction Matching (APM) represents another efficacious DD algorithm. It circumvents the nested optimization challenges prevalent in various popular methodologies Du et al. [2023], Loo et al. [2022], Cui et al. [2023] by employing an adversarial framework that mines critical point samples within the distribution of real data.

However, there exists room for improvement in SRe^{2}L Yin et al. [2023] and APM Chen et al. [2023] when it comes to the distillation of low data volumes (e.g. SRe^{2}L only achieves 2.0% at IPC 1 on CIFAR-100 and APM achieves 10.9% at IPC 1 on CIFAR-100 as Tabel 2 shows). Enhanced methods following SRe^{2}L like G-VBSM Shao et al. [2023] and WMDD Liu et al. [2023] are proposed then. G-VBSM implements a series of intricate strategies, including augmenting intra-class diversity in the distilled datasets, utilization of teacher models with varied architectures, and a greater number of statistical measures compared to SRe^{2}L (convolutional statistics at both patch and channel levels). In contrast to the complex strategies of G-VBSM, we aspire to enhance the performance of SRe^{2}L and APM through a more simplified and universally applicable method.

To attain this objective, we introduce a streamlined method for enhancing the performance of SRe^{2}L and APM, Distributed Boosting (DB) as Fig.3. DB firstly partitions the original dataset for several clients, wherein the teacher model is trained on each distinct client and subsequently generates corresponding data. Nevertheless, partitioning the original dataset for distributed training leads to a decline in the quality of data synthesized by each client, and directly training with these inferior datasets does not fulfill our objective of better learning the real data distribution through multiple clients. To address this issue, since the soft labels obtained through the integration of multiple teachers effectively enhance the generalization capabilities of the student model in knowledge distillation You et al. [2017], we utilized a method where all clients' trained models randomly participate to soft-label the aggregated data by FKD Shen and Xing [2022], thereby encapsulating more of the real dataset's effective information within the distillation data. This process culminates in the realization of two higher-performing algorithms: DB-SRe^{2}L and DB-APM. Specifically, DB-SRe^{2}L, compared to SRe^{2}L, demonstrates performance improvements of 25.2%, 26.9%, and 26.2% on ImageNet at Images Per Class (IPC) 10, CIFAR-10 at IPC 10, and CIFAR-10 at IPC 50, respectively. Meanwhile, DB-APM, in comparison to APM, exhibits performance enhancements of 21.2% and 20.9% on CIFAR-10 at IPC 10, CIFAR-100 at IPC 1, respectively. We also provide a theoretical proof of convergence for the method.

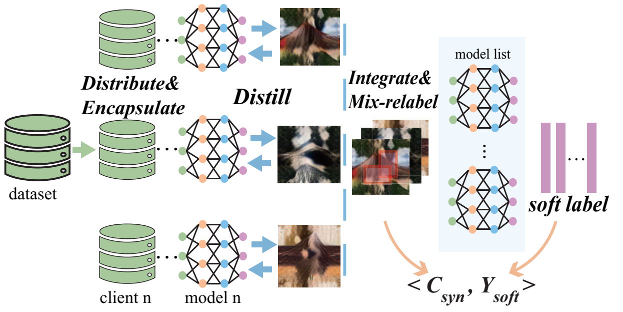

Figure 1: Distributed Boosting

Contributions.

- We introduce DB for SRe^{2}L and APM, leading to DB-SRe^{2}L and DB-APM. To the best of our knowledge, DB is the first DD approach applicable to distributed parallel computing scenarios. Experiments demonstrate that DB-SRe^{2}L and DB-APM exhibit significant performance improvements in low data volumes compared to SRe^{2}L and APM.
- DB-SRe^{2}L and DB-APM have achieved SOTA performance across multiple metrics. In comparison, we achieved an accuracy of 46.5% under ImageNet IPC10 with ResNet18, surpassing the leading G-VBSM 15.1%.
- We analyzed the impact of introducing DB on the efficiency, investigated cross-architecture generalization experiments and studied the impact of varying client numbers on the performance of DB. Furthermore, we provide a theoretical proof of convergence for the DB.

2 BACKGROUND

2.1 DATASET DISTILLATION.

The purpose of dataset condensation is to derive a compact synthetic dataset encapsulating a significant quantum of the information inherent in the original data. Given a voluminous labeled dataset $\mathcal{T} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), ..., (\boldsymbol{x}_{|\mathcal{T}|}, \boldsymbol{y}_{|\mathcal{T}|})\}$ where $\boldsymbol{x}$ is the data and $\boldsymbol{y}$ is the label, the goal is to synthesize a diminutive condensed dataset $\mathcal{S} = \{(\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{y}}_1), ..., (\tilde{\boldsymbol{x}}_{|\mathcal{S}|}, \tilde{\boldsymbol{y}}_{|\mathcal{S}|})\}(|\mathcal{S}| \ll |\mathcal{T}|)$ that retains the pivotal information from the original $\mathcal{T}$. The learning objective with respect to this synthetic condensed data is:

$$\boldsymbol{\theta}_{\mathcal{S}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{S}}(\theta) \tag{1}$$

where $\arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}) = \mathbb{E}_{(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\in\mathcal{S}}[\ell(\phi_{\boldsymbol{\theta}_{\mathcal{S}}}(\tilde{\boldsymbol{x}}), \tilde{\boldsymbol{y}})]$, $\tilde{\boldsymbol{y}}$ is the soft label corresponding to the synthetic data $\tilde{\boldsymbol{x}}$, $\boldsymbol{\theta}$ is the parameters of the model ϕ.

Adhering to the conceptual framework of coresets Bachem et al. [2017] and ϵ-approximate Sachdeva and McAuley

[2023], the objective of the data condensation task can be formulated as:

$$\sup\{|\ell(\phi_{\boldsymbol{\theta}_\mathcal{T}}(\boldsymbol{x}), \boldsymbol{y}) - \ell(\phi_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{x}), \boldsymbol{y})|\}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{T}} \leq \epsilon \tag{2}$$

where ℓ is the loss function, $\phi_{\boldsymbol{\theta}}$ is the model trained on the dataset $\mathcal{T}$ or $\mathcal{S}$, ϵ represents the performance differential for models trained on the synthetic dataset versus the original comprehensive dataset. Consequently, our goal is to optimize the synthetic dataset $\mathcal{S}$ via:

$$\underset{\mathcal{S},|\mathcal{S}|}{\arg\min}(\sup\{|\ell(\phi_{\boldsymbol{\theta}_\mathcal{T}}(\boldsymbol{x}), \boldsymbol{y}) - \ell(\phi_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{x}), \boldsymbol{y})|\}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{T}}) \tag{3}$$

Subsequently, we can acquire $< \text{data}, \text{label} > \in \mathcal{S}$, accompanied by the corresponding quantity of condensed data in each class.

2.2 SQUEEZE, RECOVER AND RELABEL

Squeeze, Recover and Relabel (SRe^{2}L) Yin et al. [2023] is a approach to accomplish the decoupling of the traditional bilevel optimization, comprising three distinct stages: squeeze, recover, and relabel.

In the squeeze stage, the objective is to extract pertinent information from the nascent dataset and amalgamate it into an advanced deep neural network structure. The squeeze stage is tantamount to conventional model training executed on the original dataset:

$$\boldsymbol{\theta}_\mathcal{T} = \underset{\boldsymbol{\theta}}{\arg\min}\, \mathcal{L}_\mathcal{T}(\boldsymbol{\theta}) \tag{4}$$

where $\mathcal{L}_\mathcal{T}(\boldsymbol{\theta})$ conventionally employs a cross-entropy loss function as $\mathcal{L}_\mathcal{T}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}}[\boldsymbol{y}\log(\boldsymbol{p}(\boldsymbol{x}))]$.

In the recover stage, this stage entails aligning the ultimate classification output with intermediary Batch Normalization (BN) statistical data. Additionally, this stage employs multicrop optimization techniques. The learning objective for this phase can be articulated as:

$$\underset{\mathcal{S},|\mathcal{S}|}{\arg\min}\, \ell(\phi_{\boldsymbol{\theta}_\mathcal{T}}(\tilde{\boldsymbol{x}}), \boldsymbol{y}) + \mathcal{R}_{BN}(\tilde{\boldsymbol{x}}) \tag{5}$$

where $\mathcal{R}_{BN}(\tilde{\boldsymbol{x}})$ denotes the BN Trajectory Matching Loss. $\phi_{\boldsymbol{\theta}_\mathcal{T}}$ represents the model pre-trained during the initial stage, which remains frozen in this phase.

In the relabel stage, to reflect the true soft labels of the recovered data, a pre-generated soft labeling approach is utilized, akin to FKD Shen and Xing [2022].

$$\tilde{\boldsymbol{y}}_i = \phi_{\boldsymbol{\theta}_\mathcal{T}}(\tilde{\boldsymbol{x}}_{\boldsymbol{R}_i}) \tag{6}$$

where $\tilde{\boldsymbol{x}}_{\boldsymbol{R}_i}$ represents the i-th crop of the synthetic image, and $\tilde{\boldsymbol{y}}_i$ denotes the corresponding soft label. Subsequently, the model $\phi_{\boldsymbol{\theta}_\mathcal{S}}$ can be trained on the synthetic dataset, utilizing the subsequent objective:

$$\mathcal{L}_{syn} = -\sum_i \tilde{\boldsymbol{y}}_i \, log \, \phi_{\boldsymbol{\theta}_\mathcal{S}}(\tilde{\boldsymbol{x}}_{\boldsymbol{R}_i}) \tag{7}$$

2.3 ADVERSARIAL PREDICTIVE MATCHING

Adversarial Predictive Matching (APM) Chen et al. [2023] represents a dataset distillation framework in an adversarial manner, aimed at identifying those hard samples x_i which maximize the divergence in prediction outcomes between models trained on these hard samples $\phi_{\theta_\mathcal{S}}$ and models trained on the original dataset $\phi_{\theta_\mathcal{T}}$. Meanwhile APM minimizes the disparity in logit outputs between $\phi_{\theta_\mathcal{S}}$ and $\phi_{\theta_\mathcal{T}}$ by utilizing soft labels. The specific optimizing strategy employed by APM is as follows:

$$\mathcal{L}_{syn} = \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\sum_{e=0}^{E-1}(-\log[d(\phi_{\theta_\mathcal{T}}(\tilde{x}_i), \phi_{\theta^e_\mathcal{S}}(\tilde{x}_i))] + \alpha H(\tilde{y}_{\tilde{x}_i}, p(\tilde{y}_{\tilde{x}_i}|\tilde{x}_i;\theta_\mathcal{T}))) \tag{8}$$

Where $H(\tilde{y}_{\tilde{x}_i}, p(\tilde{y}_{\tilde{x}_i}|\tilde{x}_i;\theta_\mathcal{T})) = -\log(p(\tilde{y}_{\tilde{x}_i}|\tilde{x}_i;\theta_\mathcal{T}))$ signifies the cross-entropy loss incurred by $\theta_\mathcal{T}$ with respect to $\tilde{x}_i$, concerning the corresponding distilling class $\tilde{y}_{\tilde{x}_i}$, α denotes a hyperparameter facilitating a flexible equilibrium, E represents the total number of training epochs and $d(.,.)$ symbolizes a metric for quantifying distance. From an intuitive standpoint, this loss function is designed to guide the synthetic samples towards evolving into instrumental, hard data points, thereby enhancing the efficacy of the training regimen for $\theta_\mathcal{S}$.

3 APPROACH

3.1 DISTRIBUTED BOOSTING

To alleviate the inherent constraints in SRe^{2}L Yin et al. [2023] and APM Chen et al. [2023], specifically the suboptimal quality generation in low DD data volume scenarios, a distributed boosting methodology has been implemented, leading to DB-SRe^{2}L and DB-APM. DB is consist of three stages as Distribute&Encapsulate, Distill and Integrate&Mix-relabel.

Stage.1 Distribute&Encapsulate.

In this stage, the primordial dataset $\mathcal{T} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), ..., (\boldsymbol{x}_{|\mathcal{T}|}, \boldsymbol{y}_{|\mathcal{T}|})\}$ is evenly allocated across N clients, each affiliated with the predetermined model for encapsulating information purposes. To validate the efficacy and convenience of our method, we employed a random selection allocation approach at this stage. In subsequent experimental analyses, we hypothesize that employing a more rational and refined data allocation method, such as Zhou et al. [2023], could further enhance the performance of DB.

Stage.2 Distill. Subsequent to the encapsulation phase on clients' respective data $\mathcal{T}_n, n \in \{1, 2, ..., N\}, |\mathcal{T}_n| = \frac{|\mathcal{T}|}{N}$, the encapsulated model $\theta_{\mathcal{T}_n}$ of each client engenders distilled data $\mathcal{S}_n$. In this stage, DB-SRe^{2}L executes the Recover stage in SRe^{2}L and DB-APM executes (8) in APM to achieve distilled/synthetic data $\tilde{\boldsymbol{x}}_i$.

Stage.3 Integrate&Mix-relabel. Nonetheless, segmenting the initial data corpus into clients for training precipitates a diminution in the fidelity of the data synthesized by each computational unit. Direct engagement in training with these suboptimal data assemblages fails to achieve the quintessential goal of augmenting comprehension of the inherent data distribution via multiple algorithmic models. To mitigate this quandary, we implemented a stratagem wherein each participant's computational model is harnessed to bestow soft labels upon the distilled data $\tilde{\boldsymbol{x}}_i$ by FKD Shen and Xing [2022]. Specifically, following the regular random-crop resize training strategy, we randomly crop some regions from one image and employ other augmentations like flipping on them, then input these regions into the teachers to generate the corresponding soft label vectors. We store all the region coordinates and augmentation hyper-parameters with the soft label for the following training phase. This technique facilitates the encapsulation of a more substantial portion of the data corpus's efficacious information within the resultant compressed data compilations. In this stage, each client's data and encapsulated models are integrated. For Mix-relabel, subsequent to the aggregation of the distilled data generated at each client, we employ a random selection from the list of all clients' encapsulated models for bestowing soft labels by FKD upon each crop:

$$\tilde{\boldsymbol{y}}_i^n = \phi_{\boldsymbol{\theta}_{\mathcal{T}_k}}(\tilde{\boldsymbol{x}}^n_{\boldsymbol{R}_i}) \tag{9}$$

where $\tilde{\boldsymbol{x}}^n_{\boldsymbol{R}_i}$ represents the i-th crop of the synthetic image from $\mathcal{S}_n$, $\tilde{\boldsymbol{y}}_i^n$ denotes the corresponding soft label and $\theta_{\mathcal{T}_k}$ is randomly selected from $\{\theta_{\mathcal{T}_1}, \theta_{\mathcal{T}_2}, ..., \theta_{\mathcal{T}_n}\}$. Subsequently, the model $\phi_{\boldsymbol{\theta}_S}$ can be trained on the integrated synthetic dataset, utilizing the subsequent objective:

$$\mathcal{L}_{syn} = -\sum_{n}\sum_{i} \tilde{\boldsymbol{y}}_i^n \, log \, \phi_{\boldsymbol{\theta}_S}(\tilde{\boldsymbol{x}}^n_{\boldsymbol{R}_i}) \tag{10}$$

Impact of DB on the efficiency of SRe^{2}L and APM. SRe^{2}L and APM both implement a strategy of using trained teacher models to apply soft labels to generated data during execution. Specifically, SRe^{2}L employs the model from the squeeze stage for soft labeling by FKD, while APM uses multiple teacher models with different initializations trained on the original complete dataset to apply soft labels to the generated data. Therefore, the Integrate & Mix-relabel stage in our DB does not affect the efficiency of the original algorithms. Our impact on computational efficiency primarily originates from the Distribute & Encapsulate stage, which we will analyze further. For a single GPU, we assume the encapsulation process for each client is completed sequentially. Since we have partitioned the original dataset, the total data volume through all clients does not increase, making the efficiency impact of our Distribute & Encapsulate stage very minimal compared to the original algorithms. In multi-GPU distributed computing scenarios, the simplest form of distributed generation involves equipping each client with a complete dataset to independently perform dataset compression algorithms and then aggregate the results. Our DB offers higher efficiency compared to simply having each client independently perform dataset compression algorithms because the data volume for each client in our method is 1/N of that in the simple independent distribution approach. In summary, the impact of our DB on the efficiency of the original algorithms is negligible on a single GPU, while in multi-GPU distributed computing scenarios, DB can adapt to this context and enhance the computational efficiency of the original algorithms to the greatest extent.

3.2 CONVERGENCE PROOF

The purpose of this convergence proof is to prove the convergence of the relabel stage of this scheme, that is, the loss labeled in the relabel stage converges to the bound and has better convergence than local dist-squeeze.

We make the following assumptions:

Assumption 1: Non-convex loss function: Assume that $\mathcal{L}_{\mathcal{T},c}(\theta)$ is non-convex for all n, but satisfies certain smoothness conditions.

Assumption 2: Bounded gradient: Keeping the null hypothesis unchanged, there is a constant G such that for all n and θ, there is $\|\nabla\mathcal{L}_{\mathcal{T},c}(\theta)\| \leq G$.

Assumption 3: Appropriate learning rate: The setting of the learning rate η_t remains unchanged and satisfies the Robbins-Monro condition.

Theorem 1: For DB, given a constant B, the total number of iterations T, the bound of the update on the distilled dataset is:

$$\min_{t\in\{1,...,T\}} \mathbb{E}\left[\|\nabla\mathcal{L}_{\mathcal{T}}(\theta^{(t)})\|^2\right] \leq \frac{B}{\sqrt{T}} \tag{11}$$

More details of the proof are presented in Appendix A.2

4 EXPERIMENTS

Datasets. We conducted comparative experiments on a large-scale dataset, encompassing the ImageNet-1k Russakovsky et al. [2015] and smaller-scale datasets, which include and CIFAR10 and CIFAR 100 Krizhevsky et al. [2009]. In addressing ImageNet, for efficiency considerations, DataDAM Sajedi et al. [2023] and TESLA Cui et al. [2023] employed a downsampled version of ImageNet at 64×64 resolution, MTT Cazenavette et al. [2022] utilized ImageNette, which comprises merely 10 categories, whereas SRe^{2}L Yin et al. [2023], G-VBSM Shao et al. [2023], and DB-SRe^{2}L employed the full-scale 224×224 ImageNet dataset.

Setup and Baseline for DB-SRe^{2}L & DB-APM. Owing to our approach's emphasis on demonstrating the performance

Table 1: Comparison in ImageNet-1k

Dataset	IPC	CW128	ResNet18				ResNet50		
		DataDAM	TESLA	SRe^{2}L	G-VBSM	**DB-SRe^{2}L**	SRe^{2}L	G-VBSM	**DB-SRe^{2}L**
ImageNet-1k	10	6.3±0.0	7.7±0.1	21.3±0.6	31.4±0.5	**46.5±0.5**	28.4±0.1	35.4±0.8	**48.3±0.4**
	50	15.5±0.2	-	46.8±0.2	51.8±0.4	**53.8±0.3**	55.6±0.3	58.7±0.3	**60.1±0.3**

[1] DataDAM Sajedi et al. [2023] and TESLA Cui et al. [2023] use the downsampled 64×64 ImageNet-1k.

[2] Except for G-VBSM Shao et al. [2023], 128-width ConvNet(CW128), ResNet18, and ResNet50 denote the model of data synthesis and evaluation. G-VBSM employs a set of models {ResNet18/50, MobileNetV2, EfficientNet-B0, ShuffleNetV2-0.5} for data synthesis.

[3] Experimental results are cited from G-VBSM Shao et al. [2023].

Table 2: Comparison in CIFAR

Dataset	IPC	CW128											ResNet18		
		RCIG	DM	DSA	FRePo	MTT	TESLA	CAFE	FTD	DataDAM	APM	**DB-APM**	SRe^{2}L	G-VBSM	**DB-SRe^{2}L**
CIFAR-10	10	**69.1±0.4**	49.2±0.8	53.2±0.8	65.5±0.6	65.3±0.7	66.4±0.8	46.3±0.6	66.6±0.3	54.2±0.8	41.5±0.4	62.7±0.3	27.2±0.5	53.5±0.6	**54.1±0.2**
	50	73.5±0.3	63.7±0.5	66.8±0.4	71.7±0.2	71.6±0.2	72.6±0.7	55.5±0.6	73.8±0.2	67.0±0.4	73.4±0.2	**74.3±0.2**	47.5±0.6	59.2±0.4	**73.7±0.3**
CIFAR-100	1	**39.3±0.4**	12.2±0.4	16.8±0.2	27.2±0.4	24.3±0.3	24.8±0.4	12.9±0.3	25.2±0.2	14.5±0.5	10.9±0.2	31.8±0.5	2.0±0.2	**25.9±0.5**	18.7±0.4
	10	44.1±0.4	29.7±0.3	32.3±0.3	41.3±0.2	40.6±0.4	41.7±0.3	27.8±0.3	43.4±0.3	34.8±0.5	41.1±0.3	**44.8±0.4**	31.6±0.5	**59.5±0.4**	44.7±0.8

[1] Experimental results of DM Zhao and Bilen [2023], DSA Zhao and Bilen [2021], FRePo Zhou et al. [2022a], MTT Cazenavette et al. [2022], TESLA Cui et al. [2023] are cited from TESLA Cui et al. [2023]. Experimental results of CAFE Wang et al. [2022], FTD Du et al. [2023] are cited from FTD Du et al. [2023]. Experimental results of RCIG are cited from Loo et al. [2023]. Experimental results of DataDAM Sajedi et al. [2023], SRe^{2}L Yin et al. [2023] and G-VBSM Shao et al. [2023] are cited from G-VBSM Shao et al. [2023]

[2] Except for G-VBSM Shao et al. [2023], 128-width ConvNet(CW128) and ResNet18 denote the model of data synthesis and evaluation. G-VBSM employs a set of models {CW128, WRN-16-2, ResNet18, ShuffleNetV2-0.5, MobileNetV2-0.5} for data synthesis.

enhancement of DB in relation to the original methodologies, SRe^{2}L Yin et al. [2023] and APM Chen et al. [2023], which represent divergent approaches as verified in their respective foundational studies, we maintained alignment with the original validation settings for DB-SRe^{2}L and DB-APM. For DB-SRe^{2}L, we also conducted comparative analysis with the latest state-of-the-art (SOTA) work, G-VBSM Yin et al. [2023], which follows the SRe^{2}L. In the experiments with CIFAR10 and CIFAR100, we endeavored to include a comprehensive array of SOTA methods, encompassing DM Zhao and Bilen [2023], DSA Zhao and Bilen [2021], FRePo Zhou et al. [2022a], MTT Cazenavette et al. [2022], TESLA Cui et al. [2023], CAFE Wang et al. [2022], FTD Du et al. [2023], DataDAM Sajedi et al. [2023], APM Chen et al. [2023], SRe^{2}L Yin et al. [2023] and G-VBSM Shao et al. [2023].

Implementation Details. In our comparative analysis of DB-SRe^{2}L versus SRe^{2}L and DB-APM versus APM, we endeavored to employ parameters consistent with those recommended by SRe^{2}L Yin et al. [2023] and APM Chen et al. [2023]. For DB-SRe^{2}L, we employ ResNet-{18, 50} He et al. [2016] for its backbone networks. For CIFAR in DB-SRe^{2}L, the first 7×7 Conv layer of ResNet-{18, 50} is replaced by 3×3 Conv layer and the maxpool layer is discarded, following MoCo (CIFAR) He et al. [2020], which is the same setting in SRe^{2}L and G-VBSM. For DB-APM and APM, the teacher number of them is set as 10, the running round is 1800. We adopt the 3-layer ConvNet architecture Gidaris and Komodakis [2018] for distilling CIFAR-10 and CIFAR-100. We also employ ZCA whitening for CIFAR-10 and CIFAR-100 as done by previous works Chen et al. [2023], Zhou et al. [2022b], Loo et al. [2022], Cazenavette et al. [2022], Du et al. [2023], Cui et al. [2023]. The experimental results of our methodology are attained on RTX 4090 GPU.

Evaluation Metrics. We initiated the training of five randomly initialized networks from scratch on the extracted datasets, assessing their performance by reporting the mean and standard deviation of their accuracy on the test sets. In the context of dataset distillation, we employed the term ’IPC’ (an acronym for ’Images Per Class’) to denote the size of the distilled dataset.

4.1 LARGE-SCALE DATASET COMPARISON

In this experiment, DataDAM Sajedi et al. [2023] and TESLA Cui et al. [2023] use the downsampled 64×64 ImageNet-1k. Except for G-VBSM Shao et al. [2023], 128-width ConvNet(CW128), ResNet18, and ResNet50 denote the model of data synthesis and evaluation. G-VBSM employs a set of models {ResNet18/50, MobileNetV2, EfficientNet-B0, ShuffleNetV2-0.5} for data synthesis. Experimental results are cited from G-VBSM Shao et al. [2023].

Table 3: Various Client Number N of DB-SRe^{2}L in CIFAR10

IPC	SRe^{2}L	DB-SRe^{2}L													
		N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10	N=11	N=12	N=13	N=14	N=15
5	23.1	25.1	27.6	27.1	27.9	27.0	30.5	28.3	27.1	32.4	27.7	27.7	30.3	33.5	**39.4**
10	27.2	29.7	36.1	37.9	36.3	42.6	48.1	42.2	45.9	45.8	47.7	46.3	44.8	47.9	**54.1**
20	34.1	43.8	50.9	49.4	52.9	60.3	62.3	62.3	64.6	64.2	58.3	61.2	**65.1**	63.3	64.7
30	40.5	46.9	60.1	56.4	62.4	68.2	69.0	66.5	69.8	68.2	65.0	67.2	**70.0**	69.4	68.2
40	44.3	54.9	62.4	63.2	67.0	70.9	72.4	72.6	**72.9**	70.1	67.5	69.8	72.0	71.0	68.9
50	47.5	55.6	65.7	64.8	72.5	73.3	**73.7**	73.6	73.3	71.5	67.8	70.6	72.9	71.2	70.0
100	59.1	71.3	79.1	79.6	**80.6**	78.1	78.1	78.0	76.3	74.2	75.0	73.6	74.2	73.3	70.7
150	67.2	77.9	80.8	82.2	**82.6**	78.4	77.0	79.6	77.1	74.0	71.0	74.4	75.2	74.1	71.6
200	76.6	81.4	82.1	**83.9**	83.8	78.9	77.3	79.6	78.3	74.2	71.6	75.4	75.5	74.3	71.7
data size	50000	25000	16666	12500	10000	8333	7143	6250	5556	5000	4545	4166	3846	3571	3333
client acc.	94.8	91.6	87.9	88.7	85.7	83.1	82.4	80.4	77.8	76.7	76.3	76.3	76.2	75.3	73.5

Table 4: ImageNet-1k Top-1 Acc. on cross-architecture gener al ization under IPC 50.

Method	Evaluation Model			
	DeiT-Tiny	ResNet50	MobileNetv2	Swin-Tiny
SRe^{2}L	15.41	55.29	36.59	39.23
G-VBSM	29.43	-	48.66	**57.40**
DB-SRe^{2}L	**40.38**	**57.33**	**51.81**	52.02

[1] The synthesis model for SRe^{2}L and DB-SRe^{2}L is ResNet18. G-VBSM employs a set of models {CW128, WRN-16-2, ResNet18, ShuffleNetV2-0.5, MobileNetV2-0.5} for data synthesis

[2] Experimental results are cited from SRe^{2}L Yin et al. [2023] and G-VBSM Shao et al. [2023].

Full 224×224 ImageNet-1k. As depicted in Table 1, DB-SRe^{2}L consistently outperforms both SRe^{2}L and G-VBSM at IPC {10, 50}. Notably, in scenarios of extremely low data compression volumes, DB-SRe^{2}L surpasses SRe^{2}L at IPC10 by 25.2% and 19.9%, respectively, and also exceeds the large-data SOTA method G-VBSM by 15.1% and 12.9%. It is important to note that G-VBSM employs multiple distinct models for encapsulating knowledge and generating soft labels, followed by validation on a single model. In contrast, our method utilizes only a single model throughout the DD and validation processes, thereby demonstrating the efficiency of our approach. In terms of client count N, DB-SRe^{2}L employs N=2 and N=3 at IPC10 and 50, respectively. The experiments validate our method's exceptional performance on large-scale datasets.

4.2 CROSS-ARCHITECTURE GENERALIZATION.

A set of models with real-world applicability is utilized in Cross-Architecture experiments, comprising ResNet50 He et al. [2016], MobileNetV2 Sandler et al. [2018], DeiT-Tiny Touvron et al. [2021], and Swin-Tiny Liu et al. [2021]. The experimental results are illustrated in Table 4. The synthesis model for SRe^{2}L and DB-SRe^{2}L is ResNet18. G-VBSM employs a set of models {CW128, WRN-16-2, ResNet18, ShuffleNetV2-0.5, MobileNetV2-0.5} for data synthesis. Experimental results are cited from SRe^{2}L Yin et al. [2023] and G-VBSM Shao et al. [2023].

Compared to SRe^{2}L, DB-SRe^{2}L demonstrates performance enhancements of 24.97%, 7.07%, 2.04%, 15.22%, and 12.79% in DeiT-Tiny, ResNet18, ResNet50, MobileNetV2, DeiT-Tiny and Swin-Tiny respectively, and achieves better performance than G-VBSM in all cases except for Swin-Tiny. Notably, G-VBSM, which utilizes more complex strategy, employs a set of models including MobileNetV2 for data synthesis. However, G-VBSM's performance on MobileNetV2 was 3.15% lower than that of DB-SRe^{2}L. This further validates the strong generalization capabilities of our DB-SRe^{2}L.

4.3 SMALL-SCALE DATASET COMPARISON

In CIFAR (Small-Scale Dataset) experiments, experimental results of DM Zhao and Bilen [2023], DSA Zhao and Bilen [2021], FRePo Zhou et al. [2022a], MTT Cazenavette et al. [2022], TESLA Cui et al. [2023] are cited from TESLA Cui et al. [2023]. Experimental results of CAFE Wang et al. [2022], FTD Du et al. [2023] are cited from FTD Du et al. [2023]. Experimental results of RCIG are cited from Loo et al. [2023]. Experimental results of DataDAM

Table 5: Various Client Number N of DB-SRe^{2}L in CIFAR100

IPC	SRe^{2}L	DB-SRe^{2}L								
		N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
1	2.0	13.3	13.8	14.7	15.1	17.0	14.5	16.5	16.8	**18.7**
5	24.8	29.7	34.3	35.8	37.0	**37.1**	34.3	31.1	33.9	34.7
10	31.6	41.3	43.3	**44.7**	43.5	41.6	39.3	37.3	37.2	38.7
20	47.9	50.5	**52.5**	50.1	49.2	47.3	44.8	41.3	39.8	42.5
50	56.2	**57.6**	57.4	54.0	51.7	49.5	46.8	43.4	40.9	44.2
data size	50000	25000	16666	12500	10000	8333	7143	6250	5556	5000
client acc.	76.5	67.8	64.3	59.6	56.6	53.1	48.6	46.5	45.7	42.2

Sajedi et al. [2023], SRe^{2}L Yin et al. [2023] and G-VBSM Shao et al. [2023] are cited from G-VBSM Shao et al. [2023]. Except for G-VBSM Shao et al. [2023], 128-width ConvNet(CW128) and ResNet18 denote the model of data synthesis and evaluation. G-VBSM employs a set of models {CW128, WRN-16-2, ResNet18, ShuffleNetV2-0.5, MobileNetV2-0.5} for data synthesis.

CIFAR-10. In Table 2, compared to APM, DB-APM exhibits an average precision increase of 21.2% and 0.9% at IPC10 and IPC50, respectively, where the synthesis and validation employed a 128-width ConvNet (CW128). Although DB-APM did not achieve the highest precision at IPC, it was only 3.7% below the best result of FTD. In contrast, DB-SRe^{2}L showed a significant improvement over SRe^{2}L, with average precision increases of 26.9% and 26.2% at IPC10 and IPC50, respectively, utilizing a ResNet18 model for synthesis and validation. Despite the more refined strategies and different synthesis model architectures of G-VBSM, DB-SRe^{2}L still demonstrated considerable advantages, particularly at IPC50, where it held a 14.5% lead. DB-APM used N=4 and N=2 at IPC10 and IPC50, respectively, while DB-SRe^{2}L used N=15 and N=7 at IPC10 and IPC50.

CIFAR-100. In Table 2, DB-APM surpasses APM with an average precision increment of 20.9% and 3.7% at IPC10 and IPC50, respectively, employing a CW128 model for both synthesis and validation. Concurrently, DB-APM achieves the optimal results compared to other SOTA algorithms at both IPC10 and IPC50. In a similar vein, DB-SRe^{2}L demonstrates an enhanced average precision of 16.7% and 13.1% over SRe^{2}L at IPC10 and IPC50, respectively, utilizing a ResNet18 model for synthesis and validation. Although DB-SRe^{2}L exhibits a slightly lower reported precision compared to G-VBSM, it is noteworthy that G-VBSM employed multiple models with diverse architectures during the DD process, whereas our approach relied solely on a single model architecture. DB-APM utilized N=8 and N=2 at IPC10 and IPC50, respectively, while DB-SRe^{2}L employed N=10 and N=4 at IPC10 and IPC50, respectively.

Table 6: Ablation Study in DB-APM

Dataset	IPC	APM	D-APM	DB-APM
CIFAR-10	10	41.5±0.4	40.5±0.3	**62.7±0.3**
	50	73.4±0.2	68.4±0.2	**74.3±0.1**
CIFAR-100	1	10.9±0.2	9.6±0.5	**31.8±0.3**
	10	41.1±0.3	39.9±0.4	**44.8±0.2**

4.4 SELECTION OF CLIENT NUMBER.

To elucidate the impact of varying the Client Number (N) on the final distilled data, we present empirical results of DB-SRe^{2}L under different N on CIFAR10 and CIFAR100 datasets, as shown in Table 3 and Table 5. These results facilitate the derivation of empirical patterns. We employed the ResNet18 model for both synthesis and validation processes. In the tables, 'data size' refers to the amount of data distributed on average to each client, while 'client acc.' indicates the average precision of clients' models encapsulating data information. Analysis of the tables reveals that with increasing IPC, the optimal N for DD decreases, and the optimal precision in DD does not surpass the average precision of the teacher models. When the data volume is low, the DB consistently achieves higher validation accuracy compared to the original algorithm, regardless of the number of client selected. However, as the distilled data volume increases, if the average accuracy of the teacher models per client is less than or only marginally better than that of the original algorithm, the results obtained using the DB may be inferior to the original algorithm. Based on this observation, we hypothesize that enhancing the precision of each client's teacher model could further improve the performance of our DB approach. A natural question arises: could repeatedly training several models with the same architecture but different initializations on the original dataset as the teacher model for each client address this issue? In fact, this approach has already been implemented in MTT, TESLA, FTD and APM. In our experiments, we set the number of teacher models in the APM algorithm to 10. Empirical results indicated in Table 2, demonstrate that our DB-APM still outperforms APM, underscoring its efficacy.

Furthermore, when data is allocated equally and randomly to each client for individual training, it essentially represents a random dataset compression approach. As demonstrated by the CIFAR10 experiments in Table 3, the DB-SRe^{2}L algorithm achieves approximately 6.6× and 6.2× higher compression rates compared to the random algorithm at IPC50 and IPC100, respectively. Similarly, as shown by the CIFAR100 experiments in Table 5, the DB-SRe^{2}L algorithm outperforms the random algorithm with approximately 5.5× and 4.2× higher compression rates at IPC10 and IPC20, respectively.

4.5 ABLATION STUDY.

Table 6 delineates the ablation study concerning DB-APM. Herein, D-APM denotes the DB-APM stripped of the Mix-relabel operation. The tabulation reveals that D-APM uniformly manifested inferior accuracy across other conditions. This decrement in performance is attributed to the reduced volume of encapsulated data per client, resulting in diminished quality of data generated by the model. However, the employment of the Mix-relabel strategy at this juncture serves to rectify this shortfall, thereby enabling DB-APM to outperform APM in terms of efficacy.

5 RELATED WORK

In the domain of dataset distillation, the prevalent strategy involves formulating the task as a bi-level optimization problem, with the objective of minimizing the generalization error of a model trained on refined data with respect to the original data Wang et al. [2018], Zhou et al. [2022b], Nguyen et al. [2021], Loo et al. [2022]. However, addressing this problem is not straightforward, especially when optimizing the surrogate model via gradient descent, as it necessitates unrolling a complex computation graph. To confront this challenge, recent studies have proposed using Kernel Ridge Regression (KRR) to approximate the model training process, thereby obtaining a closed-form solution for the optimal weights Zhou et al. [2022b], Nguyen et al. [2021], or by convexifying the model training process with the Neural Tangent Kernel (NTK) and directly solving the problem through Implicit Gradients (IG) Loo et al. [2022]. Guo et al. [2023] adopts the reduced kernel mean embedding to distill a smaller reduced dataset. Despite these approaches, they either require significant computational resources or are subject to the limitations of convex optimization relaxation. Furthermore, there are other methods based on matching techniques. Gradient matching methods involve a one-step distance matching process, aligning networks trained on the original dataset with those trained on synthetic data, incorporating techniques such as Zhao et al. [2020], Zhao and Bilen [2021] and Kim et al. [2022]. Distribution matching methods aim to directly align the distributions of original and synthetic data through a single-stage optimization process, involving techniques such as Zhao and Bilen [2023], Wang et al. [2022] and Zhao and Bilen [2022]. Trajectory matching methods are dedicated to aligning the training trajectories of models trained on both original and synthetic data, including techniques such as Cazenavette et al. [2022] and Cui et al. [2023]. Yin et al. [2023] and Shao et al. [2023] can be considered as distribution matching decoupling solutions on the distribution of statistics. Chen et al. [2023] is a method based on adversarial prediction matching.

6 DISCUSSION

Although our DB-SRe^{2}L and DB-APM methods have achieved substantial performance improvements over SRe^{2}L and APM respectively, and have attained SOTA precision in some tests, the proposed DB method exhibited significant effects primarily in scenarios with lower volumes of distilled data. Moreover, if random selection is considered a simplest form of DD algorithm, our DB results demonstrate the potential of combining two different DD algorithms. A straightforward approach would be the use of a coreset method like Dataset Quantization Zhou et al. [2023], to partition the original dataset more effectively. This would enhance the average precision of the client's teacher models by more rational partitioning of the original dataset, ultimately leading to higher-quality distilled data.

7 CONCLUSION

In this work, we address the issue of limited performance in low compression data volume for two advanced and highly efficient algorithms SRe^{2}L and APM in the domain of dataset compression, by proposing a distributed enhancement method, DB, that is ingeniously suited for parallel computing environments. Our DB method encompasses three steps: Distribute&Encapsulate, Distill and Integrate&Mix-relabel. We have conducted a theoretical convergence proof for DB, analyzed the impact of DB on the operational efficiency of the original algorithms, and executed extensive experiments. Empirical results demonstrate that our improved methodologies, DB-APM and DB-SRe^{2}L, not only significantly outperform the original algorithms across various metrics but also achieve state-of-the-art performance. Additionally, our method demonstrates the potential for combining two distinct dataset distillation techniques and lays the groundwork for distributed dataset distillation. In future research, we aim to further explore more advanced methods of distributed dataset distillation.

References

Olivier Bachem, Mario Lucic, and Andreas Krause. Practical coreset constructions for machine learning. *arXiv preprint arXiv:1703.06476*, 2017.

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.

Mingyang Chen, Bo Huang, Junda Lu, Bing Li, Yi Wang, Minhao Cheng, and Wei Wang. Dataset distillation via adversarial prediction matching. *arXiv preprint arXiv:2312.08912*, 2023.

Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

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.

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

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.

Lan-Zhe Guo, Zhi Zhou, Yu-Feng Li, and Zhi-Hua Zhou. Identifying useful learnwares for heterogeneous label spaces. In *International Conference on Machine Learning*, pages 12122–12131. PMLR, 2023.

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.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

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

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

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

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

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.

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.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

Noel Loo, Ramin Hasani, Alexander Amini, and Daniela Rus. Efficient dataset distillation using random feature approximation. *Advances in Neural Information Processing Systems*, 35:13877–13891, 2022.

Noel Loo, Ramin Hasani, Mathias Lechner, and Daniela Rus. Dataset distillation with convexified implicit gradients. In *International Conference on Machine Learning*, pages 22649–22674. PMLR, 2023.

Aru Maekawa, Naoki Kobayashi, Kotaro Funakoshi, and Manabu Okumura. Dataset distillation with attention labels for fine-tuning bert. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 119–127, 2023.

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

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

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

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

Ahmad Sajedi, Samir Khaki, Ehsan Amjadian, Lucy Z Liu, Yuri A Lawryshyn, and Konstantinos N Plataniotis. Datadam: Efficient dataset distillation with attention matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17097–17107, 2023.

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.

Mattia Sangermano, Antonio Carta, Andrea Cossu, and Davide Bacciu. Sample condensation in online continual learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 01–08. IEEE, 2022.

Shitong Shao, Zeyuan Yin, Muxin Zhou, Xindong Zhang, and Zhiqiang Shen. Generalized large-scale data condensation via various backbone and statistical matching. *arXiv preprint arXiv:2311.17950*, 2023.

Zhiqiang Shen and Eric Xing. A fast knowledge distillation framework for visual recognition. In *European Conference on Computer Vision*, pages 673–690. Springer, 2022.

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

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

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

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

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

Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. *arXiv preprint arXiv:2306.13092*, 2023.

Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1285–1294, 2017.

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

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

Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6514–6523, 2023.

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

Daquan Zhou, Kai Wang, Jianyang Gu, Xiangyu Peng, Dongze Lian, Yifan Zhang, Yang You, and Jiashi Feng. Dataset quantization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17205–17216, 2023.

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

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

Supplementary Material

A CONVERGENCE PROOF

A.1 ASSUMPTION

We make the following assumptions:

Assumption 1: Non-convex loss function: Assume that $\mathcal{L}_{\mathcal{T},c}(\theta)$ is non-convex for all n, but satisfies certain smoothness conditions.

Assumption 2: Bounded gradient: Keeping the null hypothesis unchanged, there is a constant G such that for all n and θ, there is $\|\nabla \mathcal{L}_{\mathcal{T},c}(\theta)\| \leq G$.

Assumption 3: Appropriate learning rate: The setting of the learning rate η_t remains unchanged and satisfies the Robbins-Monro condition.

A.2 PROOF

First, we consider local updates per client. Since the gradient is bounded, we have:

$$\|\theta_n^{(t+1)} - \theta_n^{(t)}\| = \|\eta_t \nabla \mathcal{L}_{\mathcal{T},c}(\theta_n^{(t)})\| \leq \eta_t G \tag{12}$$

The update of global parameters can be written as:

$$\theta^{(t+1)} = \theta^{(t)} - \frac{\eta_t}{N} \sum_{n=1}^{N} \nabla \mathcal{L}_{\mathcal{T},c}(\theta_n^{(t)}) \tag{13}$$

For the non-convex case, we use a common smoothness assumption instead of convexity. Assume that the loss function $\mathcal{L}_{\mathcal{T}}(\theta)$ is L-smooth, that is, for any θ_1, θ_2, there is:

$$\mathcal{L}_{\mathcal{T}}(\theta_2) \leq \mathcal{L}_{\mathcal{T}}(\theta_1) + \nabla \mathcal{L}_{\mathcal{T}}(\theta_1)^T(\theta_2 - \theta_1) + \frac{L}{2}\|\theta_2 - \theta_1\|^2 \tag{14}$$

For L-smooth non-convex functions, we consider the following inequalities:

$$\mathcal{L}_{\mathcal{T}}(\theta^{(t+1)}) - \mathcal{L}_{\mathcal{T}}(\theta^{(t)}) \leq \nabla \mathcal{L}_{\mathcal{T}}(\theta^{(t)})^T(\theta^{(t+1)} - \theta^{(t)}) + \frac{L}{2}\|\theta^{(t+1)} - \theta^{(t)}\|^2 \tag{15}$$

Using the expression of global parameter update, we get:

$$\mathcal{L}_{\mathcal{T}}(\theta^{(t+1)}) - \mathcal{L}_{\mathcal{T}}(\theta^{(t)}) \leq -\eta_t \nabla \mathcal{L}_{\mathcal{T}}(\theta^{(t)})^T \left(\frac{1}{N} \sum_{n=1}^{C} \nabla \mathcal{L}_{\mathcal{T},c}(\theta_n^{(t)}) \right) + \frac{L\eta_t^2 G^2}{2} \tag{16}$$

For non-convex functions, we cannot guarantee convergence to the global minimum. Instead, we focus on proving that the algorithm converges to a local minimum or some bound. In particular, we consider the expectation of a decrease in the value of a function and focus on proving a bound of the form:

$$\min_{t \in \{1,\dots,T\}} \mathbb{E}\left[\|\nabla \mathcal{L}_{\mathcal{T}}(\theta^{(t)})\|^2 \right] \leq \frac{B}{\sqrt{T}} \tag{17}$$

where B is a constant that depends on G, L, N and initial conditions, T is the total number of iterations.

This bound shows that as the number of iterations T increases, the expected minimum squared gradient norm decreases at a rate $\frac{1}{\sqrt{T}}$, pointing to what we can expect from the algorithm Find a point where the gradient is close to zero.

B EFFICIENCY COMPARISON

Since our DB algorithm does not alter the essence of the SRe2L and APM algorithms during the distillation phase, the time consumed to distill each image is consistent with that of SRe2L and APM. The distinction between our approach and SRe2L/APM lies solely in the model training phase. Here, we present the time difference per epoch for training models when two NVIDIA RTX 4090 GPUs are available, as shown in Table 7. SRe2L/APM refers to training a complete dataset with two GPU, whereas DB-SRe2L/DB-APM denotes dividing the dataset into two parts for each GPU and training them simultaneously. The model used here is ResNet-18 with a 256 batch-size. It can be observed that, in a multi-card scenario, training each sub-dataset in a distributed manner is more efficient than training a complete dataset with multiple cards.

Table 7: Comsumed Time (s)/Epoch

Dataset	ImageNet-1k	CIFAR10	CIFAR100
SRe^{2}L/APM	671.8	7.4	7.4
DB-SRe^{2}L/DB-SRe^{2}L	**382.2**	**3.3**	**3.3**

C VISUALIZATION

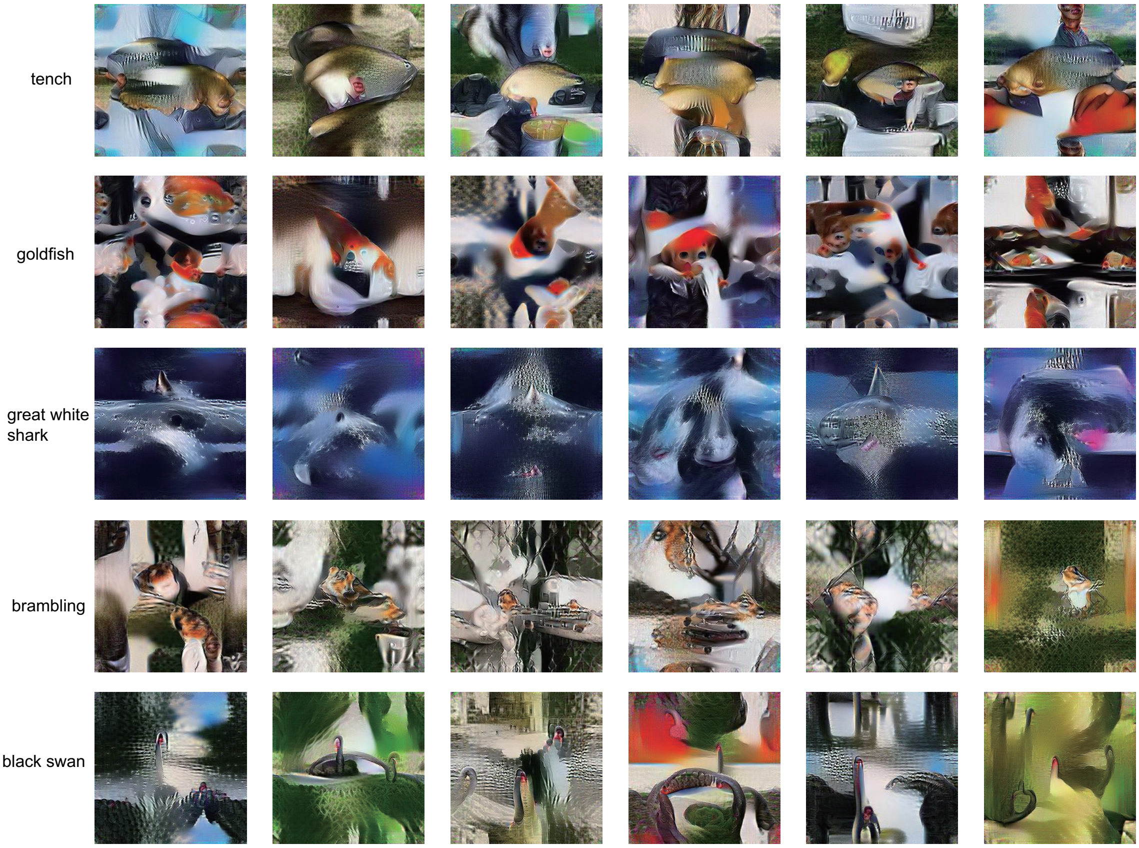

Figure 2: Visualization of synthetic images from DB-SRe^{2}L on ImageNet-1k.

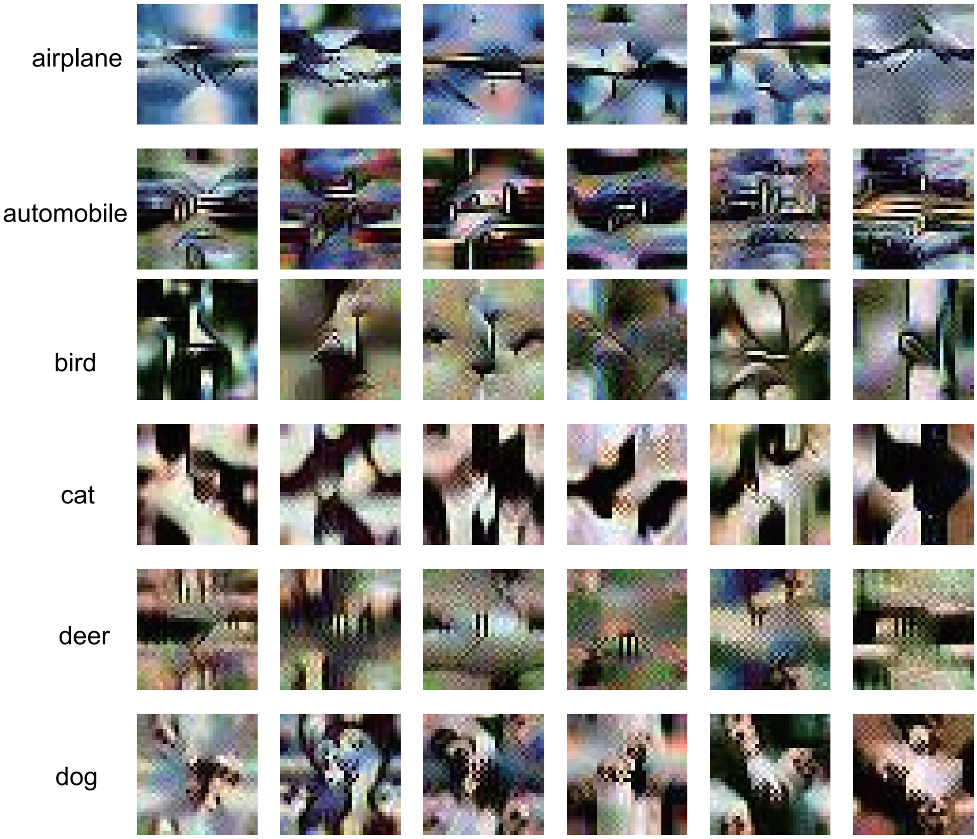

Figure 3: Visualization of synthetic images from DB-SRe^{2}L on CIFAR10.