Neural Spectral Decomposition for Dataset Distillation

Shaolei Yang[1], Shen Cheng[2], Mingbo Hong[2], Haoqiang Fan[2], Xing Wei[1], and Shuaicheng Liu[2,3,†]

[1] School of Software Engineering, Xi'an Jiaotong University, Xi'an, China
{yangshaolei@stu.xjtu.edu.cn, weixing@mail.xjtu.edu.cn}
[2] Megvii Technology, Beijing, China
{chengshen,mingbohong97,fhq}@megvii.com
[3] University of Electronic Science and Technology of China, Chengdu, China
liushuaicheng@uestc.edu.cn
[†]Corresponding Author

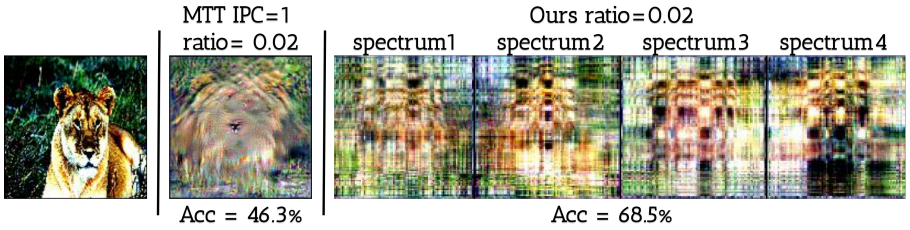

Fig. 1: *Left*: An example image belonging to the class 'lion'. *Middle*: MTT achieves an accuracy of 46.3% with a compression ratio of 0.02. *Right*: The proposed method enhances the accuracy to 68.5% under the same compression ratio.

Abstract. In this paper, we propose Neural Spectrum Decomposition, a generic decomposition framework for dataset distillation. Unlike previous methods, we consider the entire dataset as a high-dimensional observation that is low-rank across all dimensions. We aim to discover the low-rank representation of the entire dataset and perform distillation efficiently. Toward this end, we learn a set of spectrum tensors and transformation matrices, which, through simple matrix multiplication, reconstruct the data distribution. Specifically, a spectrum tensor can be mapped back to the image space by a transformation matrix, and efficient information sharing during the distillation learning process is achieved through pairwise combinations of different spectrum vectors and transformation matrices. Furthermore, we integrate a trajectory matching optimization method guided by a real distribution. Our experimental results demonstrate that our approach achieves state-of-the-art performance on benchmarks, including CIFAR10, CIFAR100, Tiny Imagenet, and ImageNet Subset. Our code are available at `https://github.com/slyang2021/NSD`.

Keywords: Dataset distillation · Dataset condensation · Image classification · Spectral decomposition

1 Introduction

Deep learning models have achieved remarkable success across a wide range of real-world applications [8, 13, 22, 26, 27], primarily owing to the availability of extensive datasets. These large-scale datasets empower models with enhanced generalization abilities, scalable expansion potential, and heightened accuracy. However, training models on such massive datasets presents significant challenges. Primarily, the vast volume of data presents considerable challenges in terms of storage, transmission, and data preprocessing. Moreover, publishing raw data inevitably gives rise to practical concerns regarding privacy and copyright issues. To address these issues, dataset distillation [1,2,4,12,17,19,24,33,34] techniques have emerged, aiming to distill the knowledge contained within large-scale datasets into smaller, synthetic ones. This approach reduces the storage while ensuring that the model captures consistent information from both the original and distillate datasets. One classical approach is coreset selection [1,4,24], which aims to identify a smaller subset from the original dataset that can preserve the essential features and structure while offering approximate or accurate results. Another highly efficient method involves the generation of synthetic data. For instance, DM (Distribution Matching) [33] minimizes the distribution discrepancy between real and synthetic data. DC (Dataset Condense) [34] matches the training gradients, while MTT (matching training trajectory) [2] emphasizes aligning the parameter trajectories throughout the training process. These strategies, namely optimization-based methods, enable the compression of large-scale datasets while maintaining the essential information required for effective learning and inference in deep learning models.

However, these methods distillate directly in the image space, lacking the fact that natural images satisfy regularity conditions that form a low-rank data subspace. IDC [12] has been introduced to generate more informative images and effectively utilize the condensed data elements. Specifically, IDC divides images into patches and treats each patch as an individual image, enabling learning within the low-frequency subspace of the image space. This approach allows for leveraging the inherent structure present in natural images. More recently, an advancement [6] has been made to calculate shared memory across data. The concept of shared memory can be understood as subspaces spanning across the dimensions of the dataset. A step further, generator-based methods [3,18,20,29] leverage additional prior knowledge to enhance the realism of synthetic images and effectively constrain the optimization direction. By incorporating the prior knowledge of the generator, the generated images closely resemble real-world examples. We note the above methods as parameterization techniques.

Unfortunately, these parameterization methods still have some limitations, such as the requirement for additional storage space, data, and even training costs. Our key observation is that the success of parameterization methods relies on 1) low-rankness and 2) information sharing. For example, Deng et al [6] demonstrates that the information between different samples is low-rank, allowing for more samples to be enhanced within the same storage space, while

the generator-based approach also shares the parameter space among different samples, making distillation more effective.

In this work, we propose Neural Spectrum Decomposition, which innovatively decomposes the entire dataset into high-dimensional spectrum tensors and kernel transformation matrices. The information expressed in each dimension of this tensor is low-rank, allowing for further reduction in storage space. Furthermore, we create a series of spectrum vectors and transformation matrices that can generate a large number of samples through pairwise combinations. This decomposition enables information sharing across dimensions, making distillation learning more efficient and enhancing the informativeness of the samples. As illustrated in Fig. 1, the network trained with this decomposition approach has a more discriminative feature space.

In terms of optimization methods, we employ a trajectory matching strategy guided by real samples, where the trajectory matching is augmented with guidance from the real dataset distribution to compensate for the limitations of trajectory sampling. In our experiments, the results demonstrate a 47.9% improvement over the baseline method based on trajectory matching on CIFAR10 under the condition of $IPC = 1$. In addition, we achieve state-of-the-art performance on benchmarks, including CIFAR10, CIFAR100, and TinyImageNet. In summary, our contributions can be summarized as follows:

- We propose a generic decomposition framework based on two principles: low-rankness and information sharing.
- We introduce an optimization method that improves trajectory matching.
- Our experiments demonstrate the state-of-the-art (SOTA) performance of our method, and we provide an in-depth analysis of our approach.

2 Related Work

Dataset Condensation. [28] proposed for the first time dataset distillation (DD), which aims to synthesize small datasets so that the models trained using the small datasets have similar performance to the original dataset. Since then, a lot of interesting work has emerged. [34] proposed a dataset condensation(DC) method based on gradient matching, which formulates dataset condensation as a gradient matching problem between the gradients of the deep neural network weights trained on the original and synthetic data. Subsequently, [32] proposed Differentiable Siamese Augmentation(DSA), which can be effectively used to synthesize more informative synthetic images using data augmentation, resulting in better performance when using augmentation to train the network. Because previous methods focused on individual steps and were unable to model complete trajectories, [2] proposed MTT to match parametric trajectory segments trained on synthetic data with expert trajectory segments from models trained on real data, thus avoiding short-sightedness. Trajectory matching methods suffer from the so-called cumulative trajectory error caused by the discrepancy between distillation and subsequent evaluation. To mitigate the adverse effects

of this cumulative trajectory error, [7] proposed FTD to encourage optimization methods to seek flat trajectories.

Despite the good performance of gradient-matching-based methods, the computational cost of their synthesis is still high due to the complex two-layer optimization and second-order derivative computation. [33] proposed DM to synthesize condensed images by matching the feature distributions of the synthesized and original training images. Thus it can be applied to more realistic and larger datasets with complex neural architectures. [25] proposed CAFE, which can characterize the distribution of the original samples well by aligning layer-by-layer features between real and synthetic data while explicitly encoding discriminative capabilities into the synthetic set, and introducing a new two-layer optimization.

Efficient Distilled Dataset Parameterization. Unlike the above-mentioned methods for generating representative images, other methods focus on Dataset Parametrization. [12] proposed IDC for efficient parameterization of synthetic data, where four synthetic images are downsampled and thus stitched together into a single image under the same storage budget and recovered using bilinear interpolation in downstream training. [18] proposed HaBa, a further optimization of IDC, to divide the dataset into two components: data Hallucination networks and Bases, where the latter is fed into the former to reconstruct image samples. [6] achieve more efficient and effective distillation through shared common representation. Concretely, they learn a set of bases (aka "memories") which are shared between classes and combined through learned flexible addressing functions to generate a diverse set of training examples. [3] proposed GLaD, which employs a deep generative prior by parameterizing the synthetic dataset in the intermediate feature space of generative models. Although these methods achieve better data compression rates as well as higher accuracy, they are not conducive to migration to downstream tasks because they usually need to store additional information to generate images.

Spectral Decomposition. Spectral decomposition can convert images from the spatial domain to the spectrum, and process the image by analyzing the information in the spectrum. [31] found that noise levels in different spectral components have different contrasts, and the target was recovered from the low-frequency components of the image and then enhanced image details from the high-frequency components. [30] proposes a learning-based frequency selection method to remove redundant frequency components without losing accuracy. [11] designed an adaptive frequency band filtering operator to filter the characteristics of different frequency bands through position-by-position multiplication operations, achieving a better trade-off between accuracy and efficiency. In this work, we decompose the image to achieve cross-dimensional information sharing, so that the distilled image can better learn the features of different frequencies.

3 Method

In this section, we describe the proposed method where each image can be decomposed into a spectrum tensor and a transformation matrix. The entire synthetic

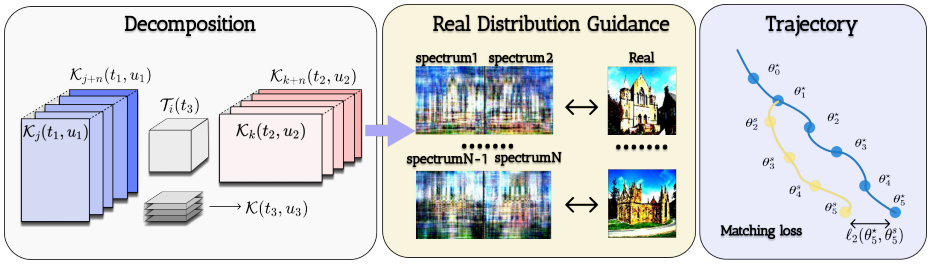

Fig. 2: The overall pipeline. The core lies in the decomposition approach of the image generation component. In terms of training strategy, in addition to using trajectory matching, we also incorporate guidance from the distribution of real samples.

dataset $\mathcal{D}_{syn}$ is a series of spectrum tensors and transformation matrices combinations. From this viewpoint, we can employ a flexible image decomposition approach, leading to improved performance through the utilization of a diminished parameter quantity. Fig. 2 illustrates our overall pipeline. In the following, we first give a general definition of the decomposition and then we step by step analyze the details of our method.

Formulation. Given a target dataset $\mathcal{D}_{syn} = \{\boldsymbol{x}_i\}_{i=1}^{N}$, we aim to find a decomposition in which each image x_i can be represented by a spectrum tensor $\mathcal{T}(t_n) \in \mathbb{R}^{t_1 \times t_2 \ldots \times t_n}$ and a kernel matrix $\mathcal{K}(t_n, u_n) \in \mathbb{R}^{t_1 \times t_2 \ldots \times t_n \times u_1 \times u_2 \ldots \times u_n}$, where n is the number of dimensions, t_n and u_n represent the original and transformed length of dimension n, respectively. By performing a simple matrix multiplication, the kernel matrix can map the length t_n of each dimension of $\mathcal{T}$ to u_n.Then we define a generic decomposition as:

$$\begin{aligned} \boldsymbol{x}_{j+(i-1)N_{\mathcal{K}}} &= \mathcal{T}_i(t_n)\mathcal{K}_j(t_n, u_n), \\ \text{s.t.} \quad & 1 \leq i \leq N_{\mathcal{T}}, 1 \leq j \leq N_{\mathcal{K}}, \end{aligned} \tag{1}$$

where $N_{\mathcal{T}}, N_{\mathcal{K}}$ denote the numbers of spectrum tensors and transformation matrices respectively. Here, $j + (i-1)N_{\mathcal{K}}$ is the index of each image which is computed by the i-th $\mathcal{T}_i$ and the j-th $\mathcal{K}_j$.

For each dimension, t_n is projected to u_n. The compression behavior is due to $t_n \leq u_n$. As $\frac{t_n}{u_n}$ decreases, the compression rate increases; conversely, as $\frac{t_n}{u_n}$ increases, the compression rate decreases. Inspired by popular deep-learning frameworks, we simply set $\mathcal{T}$ as a 4-dimensional tensor (similar to BCHW format), namely $\mathcal{T} \in \mathbb{R}^{t_1 \times t_2 \times t_3 \times t_4}$. However, even for a 4-dimensional tensor, the space required to store the kernel matrix is quite significant. To this end, we treat the transformations of each dimension as independent. After this decomposition, we can use any existing optimization framework to learn $\mathcal{T}$ and $\mathcal{K}$. This decomposition can be viewed as an implicit structural constraint.

Separable Kernel. As different dimensions are independent of each other, the entire kernel matrix can be decomposed into multiple sub-kernel matrices, which

Algorithm 1

input: Real Dataset $\mathcal{D}_{real}$;
input: Spectrum Tensor $\mathcal{T}$, Kernel Matrix $\mathcal{K}$;
input: Distillation Step N, Expert Trajectory Length M;
input: Expert Weights $\{\theta^t\}$, learning rate α;
input: loss function $norm-2$ $\ell(\cdot,\cdot)$ and cross-entropy L
repeat
  Obtain synthetic dataset $\mathcal{D}_{syn}$ with eqn.(1)
  Initialize student weight: $\theta_i^s := \theta_i^t$
  for $t = 1$ **to** N **do**
    Sample a mini-batch $B \subset \mathcal{D}_{syn}$
    Compute gradients $g_L = \nabla_\theta L_B(\theta_{i+t-1}^s)$
    Update student weight: $\theta_{i+t}^s = \theta_{i+t-1}^s - \alpha g_L$
  end for
  Compute match loss $\ell(\theta_{i+N}^s, \theta_{i+M}^t)$
  Compute real guided loss $L(\theta_{i+N}^s, \mathcal{D}_{real})$
  Update $\mathcal{T}$ and $\mathcal{K}$ with respect to $\gamma L + \ell$
until Converge

are as follows:

$$\mathcal{K}(t_n, u_n) = \hat{\mathcal{K}}(t_1, u_1) \dots \hat{\mathcal{K}}(t_n, u_n), \tag{2}$$

where $\hat{\mathcal{K}}(t_n, u_n) \in \mathbb{R}^{t_n \times u_n}$. By performing dimension decomposition, the number of parameters can be further reduced, decreasing from $\prod t_n u_n$ to $\sum t_n u_n$.

Combination. For each spectrum tensor, it can be combined with different kernel matrices. Similarly, the same kernel matrix can also be combined with different spectrum tensors. This combination strategy increases the exposure of $\mathcal{T}$ and $\mathcal{K}$ in each sampling process, thereby improving training efficiency. This information sharing across different dimensions can be seen as a form of low-rankness. By using this combination, we obtain more diverse samples.

Training Strategy. This decomposition can be plugged into any optimization algorithm. Building upon the advantage of long-term information, we enhance the performance by using MTT [2] (matching training trajectory) as a baseline and subsequently incorporating distribution information into the optimization. A trajectory contains the parameters which are trained with T steps on real or synthetic data, denoted as $\tau^* = \{\theta_t^*\}_0^T$ or $\tau^s = \{\theta_t^s\}_0^T$. The objective is to minimize $\left\|\theta_{t+N}^s - \theta_{t+M}^*\right\|_2^2$. However, the effect of distillation depends on how many teacher trajectories are available, and it is difficult for teacher trajectories to cover the full parameter space of possibilities. Hence, extracting information from real data distributions to guide optimization can enhance performance. One intuition to use real data information is that *networks trained on synthetic data can classify correctly in real data*, which is similar to KIP [21]. Assuming the final parameter of the synthesized trajectory is θ_{i+N}^s, we sample a batch of data from real trajectories and compute a classification loss for this data $L(\theta_{i+N}^s, \mathcal{D}_{real})$, as depicted in Alg. 1.

Table 1: Top-1 test accuracy of test models comparison to state-of-the-art methods. We use ConvNet3 for training and testing and show the mean and standard deviation of the five tests. We compared the coreset selection method, the distilled dataset parametrization method, and the original method(optimized in the original image space) separately. IPC: Images Per Class(budget follows Eqn. 3), Ratio (%): the ratio of condensed images to the whole training set.

	Dataset	CIFAR10			CIFAR100		TinyImageNet
	IPC	1	10	50	1	10	1
	Ratio%	0.02	0.2	1	0.2	2	0.2
Coreset	Random	$14.4_{\pm 2.0}$	$26.0_{\pm 1.2}$	$43.4_{\pm 1.0}$	$4.2_{\pm 0.3}$	$14.6_{\pm 0.5}$	$1.4_{\pm 0.1}$
	Herding	$21.5_{\pm 1.2}$	$31.6_{\pm 0.7}$	$40.4_{\pm 0.6}$	$8.4_{\pm 0.3}$	$17.3_{\pm 0.3}$	$2.8_{\pm 0.2}$
	Forgetting	$13.5_{\pm 1.2}$	$23.3_{\pm 1.0}$	$23.3_{\pm 1.1}$	$4.5_{\pm 0.2}$	$15.1_{\pm 0.3}$	$1.6_{\pm 0.1}$
Parametrization	IDC [12]	$50.0_{\pm 0.4}$	$67.5_{\pm 0.5}$	$74.5_{\pm 0.1}$	$25.1_{\pm 0.3}$	$44.8_{\pm 0.2}$	-
	HaBa [18]	$48.3_{\pm 0.8}$	$69.9_{\pm 0.4}$	$74.0_{\pm 0.2}$	$33.4_{\pm 0.4}$	$40.2_{\pm 0.2}$	-
	Deng et al. [6]	$66.4_{\pm 0.4}$	$71.2_{\pm 0.4}$	$73.6_{\pm 0.5}$	$34.0_{\pm 0.4}$	$42.9_{\pm 0.7}$	$16.0_{\pm 0.7}$
Original	DC [34]	$28.3_{\pm 0.5}$	$44.9_{\pm 0.5}$	$53.9_{\pm 0.5}$	$12.8_{\pm 0.3}$	$25.2_{\pm 0.3}$	-
	DSA [32]	$28.8_{\pm 0.7}$	$52.1_{\pm 0.5}$	$60.6_{\pm 0.5}$	$13.9_{\pm 0.3}$	$32.3_{\pm 0.3}$	-
	DM [33]	$26.0_{\pm 0.8}$	$48.9_{\pm 0.6}$	$63.0_{\pm 0.4}$	$11.4_{\pm 0.3}$	$29.7_{\pm 0.3}$	$3.9_{\pm 0.2}$
	KIP to NN [21]	$49.9_{\pm 0.2}$	$62.7_{\pm 0.3}$	$68.6_{\pm 0.2}$	$15.7_{\pm 0.2}$	$28.3_{\pm 0.1}$	-
	CAFE + DSA [25]	$31.6_{\pm 0.8}$	$50.9_{\pm 0.5}$	$62.3_{\pm 0.4}$	$14.0_{\pm 0.3}$	$31.5_{\pm 0.2}$	-
	FRePo [35]	$46.8_{\pm 0.7}$	$65.5_{\pm 0.4}$	$71.7_{\pm 0.2}$	$28.7_{\pm 0.1}$	$42.5_{\pm 0.2}$	$15.4_{\pm 0.3}$
	FTD [7]	$46.8_{\pm 0.3}$	$66.6_{\pm 0.3}$	$73.8_{\pm 0.2}$	$25.2_{\pm 0.2}$	$43.4_{\pm 0.3}$	$10.4_{\pm 0.3}$
	MTT [2]	$46.3_{\pm 0.8}$	$65.3_{\pm 0.7}$	$71.6_{\pm 0.2}$	$24.3_{\pm 0.3}$	$40.1_{\pm 0.4}$	$8.8_{\pm 0.3}$
	ours	$\mathbf{68.5}_{\pm \mathbf{0.8}}$	$\mathbf{73.4}_{\pm \mathbf{0.2}}$	$\mathbf{75.2}_{\pm \mathbf{0.6}}$	$\mathbf{36.5}_{\pm \mathbf{0.3}}$	$\mathbf{46.1}_{\pm \mathbf{0.2}}$	$\mathbf{21.3}_{\pm \mathbf{0.2}}$
	Full dataset	$84.8_{\pm 0.1}$			$56.2_{\pm 0.3}$		$37.6_{\pm 0.4}$

4 Experiments

In this section, we first introduce the datasets used and the specific experimental details. Next, we compare the proposed method with the state-of-the-art methods. Subsequently, we also conduct the ablation experiments and give an in-depth analysis. Finally, we will visualize our condensed dataset and show the impact of the decomposition.

4.1 Datasets & Implementation Details

CIFAR-10/100 The *CIFAR-10/100* [14] consists of 60,000 32x32 color nature images, each with 50,000 training images and 10,000 test images equally divided into 10/100 categories.
TinyImageNet The *TinyImageNet* [16] dataset contains 200 categories, each with 500 training images, 50 validation images, and 50 test images of 64×64 pixels, and the images are from the ImageNet dataset [5].
ImageNet Subset The ImageNet [5] dataset is a large-scale and diverse image database containing over 14 million images. It is utilized for training and

evaluating computer vision algorithms. Next, we run our method on a subset of the ImageNet, which is divided into 6 subsets following the MTT [2], namely ImageNette (various objects), ImageWoof (dog breeds), ImageFruit (fruits), ImageMeow (cats), ImageSquawk (birds) and ImageYellow (yellow objects).Each subset contains more than 10,000 images, and we adjusted the resolution of all images to 128×128 according to the original settings.

Implementation details. For a fair comparison, we adopt the same parameter settings as previous work, and in all datasets, the amount of condensed image parameters is the same as when Images Per Class (IPC) is 1/10/50, respectively. The model uses a simple three-layer convolutional neural network, each layer consists of a 3×3 convolution kernel, a convolutional layer with 128 channels, Instance normalization, RELU, and 2×2 average pooling with stride 2. Following MTT [2], 5 experiments are performed during evaluation, and the mean and variance of 5 experiments are finally taken. The learning rate is 0.01, epoch is 1000, batchsize is 256, momentum is 0.9, weight decay is 0.0005, and the default data enhancement is the same as DSA [32]. Our experiments were run on eight RTX2080ti GPUs.

Storage budgets. Because our method saves the spectrum tensor and kernel matrix rather than the synthesized image directly, we ensure that our compression rate is roughly equal to what it would be if IPC = 1/10/50, i.e.

$$\begin{aligned} size(Spectrum\ Tensor) + size(Kernel\ Matrix) \\ \approx IPC \times C \times size(Image) \end{aligned} \tag{3}$$

where $size(\cdot)$ denotes the size of the occupied storage and C denotes the number of classes.

Model details. We compare our method with the baseline method in the eqn. 3 premise. Specifically, we make $t_3 = t_4$ and $t_2 = 3$ in all experiments. When $IPC = 1$ and $IPC = 10$, we take t_3, t_4 to be one-half of the original image resolution, and when $IPC > 10$, we take t_3, t_4 to be seven-eighths of the original image size. Finally, the choice of t_1 and u_1 depends on the storage budget, and in principle we make $t_1 < u_1$. All the synthetic images are trained for 10k iterations and the spectrum tensor and kernel matrix are used with the SGD optimizer, We set the momentum rate to 0.9, and the spectrum tensor and kernel matrix are initialized with random initialization.

4.2 Evaluation on the Condensed Data

In this subsection, we evaluated the performance of our method on low-resolution classification datasets and the higher-resolution ImageNet dataset. Compared to previous methods, our approach achieved state-of-the-art results.

Low-Resolution Data: We compare our method to three state-of-the-art dataset distillation methods on various benchmark datasets [14, 16]. First, we compare our method with the three core subset selection methods, Random [4], Herding [1], and Forgetting [24]. Second, we compare our method with the original method(optimized in the original image space) for comparison, including Dataset

Table 2: We show the test results of different decomposition methods on the distillation dataset, and all of the decomposition methods achieve higher accuracy gains compared to baseline, and the learnable kernel matrix method also performs better relative to the fixed kernel matrix.

Dataset	IPC	Ratio%	BaseLine	DWT	SVD	LSVD	DCT	LDCT	ours
CIFAR10	1	0.02	46.3±0.8	51.3±0.6	49.0±0.6	53.8±0.2	57.9±0.3	62.3±0.4	68.5±0.8
	10	0.2	65.3±0.7	66.9±0.7	67.9±0.3	69.4±0.5	69.5±0.2	69.9±0.5	73.4±0.2
CIFAR100	1	0.2	24.3±0.3	26.4±0.4	25.3±0.4	27.7±0.3	30.1±0.3	35.5±0.2	36.5±0.3

Condensation(DC) [34], Differentiable Siamese Augmentation (DSA) [32], Distribution Matching(DM) [33], Kernel-Inducing-Point(KIP) [21], Aligning Features(CAFE) [25], Flat Trajectory Distillation(FTD) [7], Feature Regression with Poolin(FRePo) [35], Matching Training Trajectories(MTT) [2]. Finally, we compare our method with the Distilled Dataset Parametrization method, including Information-intensive Dataset Condensation(IDC) [12], Hallucination networks and Bases(HaBa) [18], Addressable Memories [6]. We present the performance of our method and the method mentioned above in Table 1, our method achieves state-of-the-art on all benchmark datasets compared to the previous three types of methods. In particular, We can readily observe that our method obtains a great performance gain relative to the original method, especially when the synthesized data is fewer. In CIFAR10 with IPC=1, we obtain a 37.4% improvement in accuracy relative to the previous state-of-the-art method, and even on CIFAR100 and TinyImageNet, we obtain 26.1% and 22.7% improvements, respectively. In addition, our method also obtains consistency improvement relative to the parametrization method, by 3.2%, 6.4%, and 18.1% on the CIFAR10, CIFAR100, and TinyImageNet datasets, respectively, with IPC = 1.

Table 3: Performance comparison of our method with other methods on ImageNet Subsets with IPC=1.

Dataset	MTT [2]	HaBa [18]	GLaD [3]	ours	Full Dataset
ImageFruit	26.6±0.8	34.7±1.1	23.1±0.4	**39.8±0.2**	63.9±2.0
ImageNette	47.7±0.9	51.9±1.6	38.7±1.6	**68.6±0.8**	87.4±1.0
ImageWoof	28.6±0.8	32.4±0.7	23.4±1.1	**35.2±0.4**	67.0±1.3
ImageSquawk	37.3±0.8	41.9±1.4	35.8±1.4	**52.9±0.7**	87.5±0.3
ImageMeow	26.6±0.4	36.9±0.9	26.0±1.1	**45.2±0.1**	66.7±1.1
ImageYellow	45.2±0.8	50.4±1.6	26.6±0.4	**61.0±0.5**	84.4±0.6

ImageNet Subsets: We also experimented our approach on the more challenging ImageNet [5] dataset, following the MTT [2] setup, we additionally extended the distillation model from the original three-layer ConvNet to a five-layer Con-

vNet and evaluated it on six ImageNet Subsets, and the test results are displayed in Table 3. An astute observation reveals our method distinctly outperforms all others in every experiment, simultaneously achieving a remarkable improvement over the baseline method. Most notably, on the ImageNette dataset, we attained a staggering 43% enhancement in accuracy compared to MTT [2]. This unequivocally substantiates the efficacy of our method and further corroborates its adeptness at more profoundly understanding and enhancing the informative value of samples.

4.3 Cross-architecture generalization

To facilitate more effective application to other task scenarios, the synthetic dataset must demonstrate the ability to generalize across different architectures. However many of the current dataset distillation methods suffer from performance degradation when synthetic datasets are trained on one network structure and tested on another.

To evaluate the ability of our method to generalize on other structures, we distill the CIFAR10 on ConvNet(3-layer) [9] with IPC=10, then we use four different structures to do the evaluation, ConvNet3 [9], ResNet18 [10], VGG11 [23], AlexNet [15].Table 4 shows the results of the generalization performance of our method on different structures, and our method achieves better generalization performance compared to previous methods(DC [34],KIP [21],MTT [2],FRePo [35],Deng et al. [6]). In particular, our method exceeds Deng et al. [6] by 3.7%, 1.6%, 11.8%, 11.9% on ConvNet3, ResNet18, VGG11, AlexNet, respectively. In addition, our method only drops 7.5, 4.6, and 1.0 on ResNet18, VGG11, and AlexNet, respectively. Especially on AlexNet, our method has almost no performance degradation. Because our method learns the decomposed spectrum tensor and kernel matrix, our method can learn the information of each frequency band well, and the cross-structural generalizability experiments also prove that our method has excellent generalizability.

Table 4: Cross-Architecture Results trained with ConvNet on CIFAR-10 with `ipc` = 10. We report test results of our method and previous state-of-the-art methods on ConvNet, ResNet18, VGG11, AlexNet.

Method	Evaluation Model			
	ConvNet	ResNet18	VGG11	AlexNet
DC [34]	44.9±0.5	18.4±0.4	35.9±0.7	22.4±1.4
KIP [21]	47.6±0.9	36.8±1.0	42.1±0.4	24.4±3.9
MTT [2]	65.3±0.7	46.6±0.6	50.3±0.8	34.2±2.6
FRePo [35]	65.5±0.4	58.1±0.6	59.4±0.7	61.9±0.7
Deng et al. [6]	71.2±0.4	63.9±0.3	60.6±0.2	63.8±0.8
ours	**73.4±0.2**	**64.9±1.3**	**67.8±0.2**	**71.4±0.3**

4.4 Exploring Spectral Decomposition

We also explored different variants of neural spectral decomposition. Regarding the selection of different $\mathcal{K}$, we have discrete wavelet transform (DWT), singular value decomposition (SVD), and discrete cosine transform (DCT).

Discrete Wavelet Transform. The discrete wavelet transform is specifically designed to emphasize the high-frequency components of an image, capturing intricate details and making it highly suitable for image compression. On the other hand, the wavelet transform considers both frequency and spatial information, effectively preserving the spatial properties of the image. We use the haar format as our base for the wavelet transform to categorize the image into LL,LH,HL,HH where L denotes low frequency and H denotes high-frequency. We randomly sample some frequency bands at a time during the inverse transform, so that we can better learn the information of each band during the distillation.

Singular Value Decomposition. Since singular value decomposition is a generalization of the theory of spectral analysis to arbitrary matrices, singular value decomposition can be well applied to signal processing, statistics and other fields. The formula for singular value decomposition is $M = U\Sigma V$, where the columns of U and V are the left and right singular vectors in the singular values, respectively, and the elements on the diagonal of the matrix Σ are equal to the singular values of M. Specifically, we transform the picture from (B, C, H, W) to $(B, C \times H \times W)$, and then perform singular value decomposition on the transformed matrix. After transformation, the spectrum tensor is V, and the kernel matrix is $U\Sigma$(in order to better compress the image, we only take the first n rows of V as well as the first n columns of U, and the number of parameters follows eqn. 3), then we only optimize the spectrum tensor, when synthesizing the dataset.

For the sake of general discussion, we introduce learnable SVD(LSVD), which optimizes the kernel matrix along with the spectrum tensor optimization.

Discrete Cosine Transform. The discrete cosine transform is often used by signal processing and image processing because the discrete cosine transform can bring together the more important information of an image in one piece. It is commonly utilized for lossy data compression of signals and various types of images, including still and moving images. Specifically, we first construct two transformation matrices $\mathcal{K}_l, \mathcal{K}_r$ using the discrete cosine transform, where $\mathcal{K}_l \in \mathbb{R}^{m \times n}, \mathcal{K}_r \in \mathbb{R}^{n \times m}$.Then we construct two inverse transform matrices $\mathcal{IK}_l, \mathcal{IK}_r$, where $\mathcal{IK}_l \in \mathbb{R}^{n \times m}, \mathcal{IK}_r \in \mathbb{R}^{m \times n}$. Therefore spectrum tensor $\mathcal{T} = \mathcal{K}_l \times \mathcal{O}_{img} \times \mathcal{K}_r$, where $\mathcal{O}_{img} \in \mathbb{R}^{3 \times n \times n}$ denotes the original image. The number of parameters follows eqn. 3. The discrete cosine transform formula is shown below:

$$\begin{aligned} \mathcal{K}_{l_{i,j}} = cos(\frac{\pi}{n} \times (j + 0.5) \times (i + 0.5)), \\ \text{s.t.} \quad 1 \leq i \leq m, 1 \leq j \leq n, \end{aligned} \tag{4}$$

$\mathcal{K}_r$ is calculated in the same way as in eqn. 4, and the inverse transformation only needs to be multiplied by $\frac{2}{n}, \frac{2}{m}$ for pairs of $\mathcal{K}_l$ and $\mathcal{K}_r$, respectively.

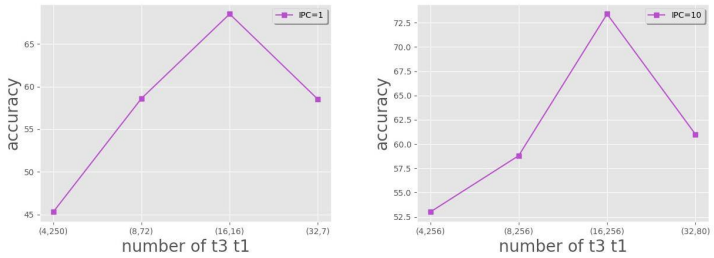

Fig. 3: Accuracy at different number(t3,t1) on the CIFAR10 dataset with IPC=1/10.

In the meantime, we introduce the learnable discrete cosine transform (LDCT), i.e., we also learn the kernel matrix, which allows the synthesized image to better utilize the information on different time domains.

Table 2 shows the test results between the datasets distilled by different decomposition methods, the baseline is MTT [2].It's easy to see that the original image after Spectral Decomposition can be better utilized for the information in the time domain, so that the synthesized dataset not only can obtain the low-frequency information but also can be better obtained to the high-frequency information. Meanwhile, the learnable kernel matrix has higher accuracy than the fixed kernel matrix, which indicates that the learnable kernel matrix is also more efficient in utilizing the information in different frequency bands.

Table 5: Test accuracy of our method on different baseline methods with (CIFAR10,IPC=10).

Method	Evaluation Model			
	ConvNet	ResNet18	VGG11	AlexNet
DC [34]	44.9±0.5	18.4±0.4	35.9±0.7	22.4±1.4
w. ours	55.6±0.3	29.5±1.1	40.1±0.3	33.5±0.2
Gain	10.7	11.1	4.2	11.1
DM [33]	48.9±0.6	36.8±1.2	41.2±1.8	34.9±1.1
w. ours	50.1±0.3	38.9±0.4	42.5±0.5	46.2±0.3
Gain	1.2	2.1	1.3	11.3
MTT [2]	65.3±0.7	46.6±0.6	50.3±0.8	34.2±2.6
w. ours	73.4±0.2	64.9±1.3	67.8±0.2	71.4±0.3
Gain	8.5	18.3	17.5	37.2

4.5 Ablation Studies

In this subsection, we study ablations to investigate the effectiveness of each module and the influence of the hyper parameters.

Building upon Different Baselines. To demonstrate the pervasiveness of our method, we implement our method on the training pipeline of multiple state-of-the-art methods, including DC [34], DM [33], and MTT [2], and we evaluate the performance of the synthetic dataset on the CIFAR10 dataset and with IPC=10. As Table 5 demonstrates, our method achieves significant improvements on the baseline method when both training and testing on ConvNet. Specifically, our method improves 23.8%, 10.6%, and 13.0% on ConvNet with respect to DC, DM, and MTT, respectively, and for cross-structure testing, our method improves even more for accuracy than on ConvNet. This shows that our method not only migrates well to other methods but also leads to better generalization of synthetic datasets.

Number of t . We further investigate the effects of t_1, t_2 and t_3, and Figure 3 presents the test accuracy of the synthetic dataset with different t_1 and t_3 choices. We default $t_3 = t_4$, keeping the same size scale as the original image. We can discover that it is more beneficial to learn the synthetic dataset when t3 takes a moderate size.

Table 6: Ablation study on different modules of our method with CIFAR10 dataset. Dec: Spectral Decomposition, Guided loss: Real guided loss, Acc: Top-1 test accuracy .

MTT [2]	Dec	Guided loss	IPC	Acc
✓			1	46.3±0.8
✓	✓		1	67.9±0.5(↑ **21.6**)
✓	✓	✓	1	68.5±0.8(↑ **22.2**)
✓			10	65.3±0.7
✓	✓		10	71.8±0.3(↑ **6.5**)
✓	✓	✓	10	73.4±0.2(↑ **8.1**)

Real Distribution Guidance. We conduct an ablation study in Table 6 to verify the effectiveness of our method, we add our module on baseline(MTT [2]) respectively, since MTT learns the long-term information of the expert model well, the addition of the Decomposition module is a great improvement for the accuracy, which is improved by 46.6%(cifar10,ipc=1), at the same time, because MTT does not have constraints on the student model on real data and, so the addition of Real guided loss also improved by 0.8%. This result validates the effectiveness of our method.

4.6 Visualizations

In this subsection, we visualize the inter-dimensional similarity to show the effectiveness of our method.

Similarity. In Figure 4, a subset comprising 1,000 images is randomly sampled from the airplane category within the CIFAR10 [14] dataset. Subsequently,

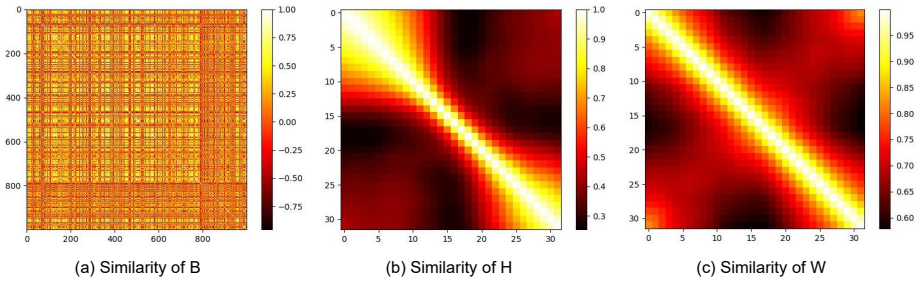

Fig. 4: We compute the similarity between B-dimension, H-dimension, and W-dimension, respectively, for the original graph with dimensions BCHW.

we conduct an analysis to compute the inter-dimensional similarity across the Batch (B), Height (H), and Width (W) dimensions within the context of the original image tensor format, denoted as BCHW. It is imperative to note that the analysis reveals a discernible local correlation among the dimensions. This observed correlation suggests the presence of redundant information along these axes, which is indicative of the underlying low-rank structure of the image data. The low-rank characteristic of an image implies that the image can be represented by a smaller number of features than that of the original dimensionality without significant loss of information. This is due to the inherent redundancy present within the image; for instance, neighboring pixels often share similar color values and patterns, leading to this correlation.

5 Conclusion

In this study, we primarily revisit parameterization methods in dataset distillation and summarize two key points for the success of these methods: low-rank and information sharing. In contrast to previous methods that only leverage specific dimensional information, such as low-frequency information from individual images, the proposed method fully utilizes the low-rankness across the entire dataset. Additionally, the combination of different spectral tensors and kernel matrices effectively enhances learning efficiency. Considering the effectiveness of low-rank, we believe that there is significant potential for exploring the proposed method in high-resolution image learning in the future.

Acknowledgement: This work was supported by National Key R&D Program of China (No. 2021ZD0110400), the Fundamental Research Funds for the Central Universities (No. xxj032023020), National Natural Science Foundation of China (No.62372091) and Sichuan Science and Technology Program of China (No.2023NSFSC0462), and sponsored by the CAAI-MindSpore Open Fund, developed on OpenI Community.

References

1. Belouadah, E., Popescu, A.: Scail: Classifier weights scaling for class incremental learning. In: WACV. pp. 1266–1275 (2020)
2. Cazenavette, G., Wang, T., Torralba, A., Efros, A.A., Zhu, J.Y.: Dataset distillation by matching training trajectories. In: CVPR. pp. 4750–4759 (2022)
3. Cazenavette, G., Wang, T., Torralba, A., Efros, A.A., Zhu, J.Y.: Generalizing dataset distillation via deep generative prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3739–3748 (2023)
4. Chen, Y., Welling, M., Smola, A.: Super-samples from kernel herding. arXiv:1203.3472 (2012)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
6. Deng, Z., Russakovsky, O.: Remember the past: Distilling datasets into addressable memories for neural networks. Advances in Neural Information Processing Systems **35**, 34391–34404 (2022)
7. Du, J., Jiang, Y., Tan, V.Y., Zhou, J.T., Li, H.: Minimizing the accumulated trajectory error to improve dataset distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3749–3758 (2023)
8. Floridi, L., Chiriatti, M.: Gpt-3: Its nature, scope, limits, and consequences. Minds and Machines **30**, 681–694 (2020)
9. Gidaris, S., Komodakis, N.: Dynamic few-shot visual learning without forgetting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4367–4375 (2018)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
11. Huang, Z., Zhang, Z., Lan, C., Zha, Z.J., Lu, Y., Guo, B.: Adaptive frequency filters as efficient global token mixers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6049–6059 (2023)
12. Kim, J.H., Kim, J., Oh, S.J., Yun, S., Song, H., Jeong, J., Ha, J.W., Song, H.O.: Dataset condensation via efficient synthetic-data parameterization. arXiv:2205.14959 (2022)
13. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
14. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012)
16. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N **7**(7), 3 (2015)
17. Li, H., Liu, D., Zeng, Y., Liu, S., Gan, T., Rao, N., Yang, J., Zeng, B.: Single-image-based deep learning for segmentation of early esophageal cancer lesions. IEEE Transactions on Image Processing (2024)
18. Liu, S., Wang, K., Yang, X., Ye, J., Wang, X.: Dataset distillation via factorization. Advances in Neural Information Processing Systems **35**, 1100–1113 (2022)

19. Ma, Z., Cao, A., Yang, F., Wei, X.: Curriculum dataset distillation. arXiv preprint arXiv:2405.09150 (2024)
20. Ma, Z., Gao, D., Yang, S., Wei, X., Gong, Y.: Dataset condensation via expert subspace projection. Sensors **23**(19) (2023). https://doi.org/10.3390/s23198148, https://www.mdpi.com/1424-8220/23/19/8148
21. Nguyen, T., Novak, R., Xiao, L., Lee, J.: Dataset distillation with infinitely wide convolutional networks. NeurIPS **34**, 5186–5198 (2021)
22. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
24. Toneva, M., Sordoni, A., Combes, R.T.d., Trischler, A., Bengio, Y., Gordon, G.J.: An empirical study of example forgetting during deep neural network learning. arXiv preprint arXiv:1812.05159 (2018)
25. Wang, K., Zhao, B., Peng, X., Zhu, Z., Yang, S., Wang, S., Huang, G., Bilen, H., Wang, X., You, Y.: Cafe: Learning to condense dataset by aligning features. In: CVPR. pp. 12196–12205 (2022)
26. Wang, Q., Chang, Y.Y., Cai, R., Li, Z., Hariharan, B., Holynski, A., Snavely, N.: Tracking everything everywhere all at once. arXiv preprint arXiv:2306.05422 (2023)
27. Wang, T., Zhang, J., Fei, J., Ge, Y., Zheng, H., Tang, Y., Li, Z., Gao, M., Zhao, S., Shan, Y., Zheng, F.: Caption anything: Interactive image description with diverse multimodal controls. arXiv preprint arXiv:2305.02677 (2023)
28. Wang, T., Zhu, J.Y., Torralba, A., Efros, A.A.: Dataset distillation. arXiv preprint arXiv:1811.10959 (2018)
29. Wei, X., Cao, A., Yang, F., Ma, Z.: Sparse parameterization for epitomic dataset distillation. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023 (2023), http://papers.nips.cc/paper_files/paper/2023/hash/9e8889198d16fb79926e71adbe38cae4-Abstract-Conference.html
30. Xu, K., Qin, M., Sun, F., Wang, Y., Chen, Y.K., Ren, F.: Learning in the frequency domain. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1740–1749 (2020)
31. Xu, K., Yang, X., Yin, B., Lau, R.W.: Learning to restore low-light images via decomposition-and-enhancement. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2281–2290 (2020)
32. Zhao, B., Bilen, H.: Dataset condensation with differentiable siamese augmentation. In: ICML. pp. 12674–12685. PMLR (2021)
33. Zhao, B., Bilen, H.: Dataset condensation with distribution matching. arXiv:2110.04181 (2021)
34. Zhao, B., Mopuri, K.R., Bilen, H.: Dataset condensation with gradient matching. ICLR **1**(2), 3 (2021)
35. Zhou, Y., Nezhadarya, E., Ba, J.: Dataset distillation using neural feature regression. Advances in Neural Information Processing Systems **35**, 9813–9827 (2022)