

# ATOM: Attention Mixer for Efficient Dataset Distillation

Samir Khaki<sup>1\*</sup>, Ahmad Sajedi<sup>1\*</sup>, Kai Wang<sup>2</sup>, Lucy Z. Liu<sup>3</sup>, Yuri A. Lawryshyn<sup>1</sup>,  
 and Konstantinos N. Plataniotis<sup>1</sup>

<sup>1</sup>University of Toronto    <sup>2</sup>National University of Singapore    <sup>3</sup>Royal Bank of Canada (RBC)

{samir.khaki, ahmad.sajedi}@mail.utoronto.ca

Code: <https://github.com/DataDistillation/ATOM>

## Abstract

Recent works in dataset distillation seek to minimize training expenses by generating a condensed synthetic dataset that encapsulates the information present in a larger real dataset. These approaches ultimately aim to attain test accuracy levels akin to those achieved by models trained on the entirety of the original dataset. Previous studies in feature and distribution matching have achieved significant results without incurring the costs of bi-level optimization in the distillation process. Despite their convincing efficiency, many of these methods suffer from marginal downstream performance improvements, limited distillation of contextual information, and subpar cross-architecture generalization. To address these challenges in dataset distillation, we propose the **ATtentiOn Mixer (ATOM)** module to efficiently distill large datasets using a mixture of channel and spatial-wise attention in the feature matching process. Spatial-wise attention helps guide the learning process based on consistent localization of classes in their respective images, allowing for distillation from a broader receptive field. Meanwhile, channel-wise attention captures the contextual information associated with the class itself, thus making the synthetic image more informative for training. By integrating both types of attention, our **ATOM** module demonstrates superior performance across various computer vision datasets, including CIFAR10/100 and Tiny-Imagenet. Notably, our method significantly improves performance in scenarios with a low number of images per class, thereby enhancing its potential. Furthermore, we maintain the improvement on cross-architectures and applications such as neural architecture search.

## 1. Introduction

Efficient deep learning has surged in recent years due to the increasing computational costs associated with train-

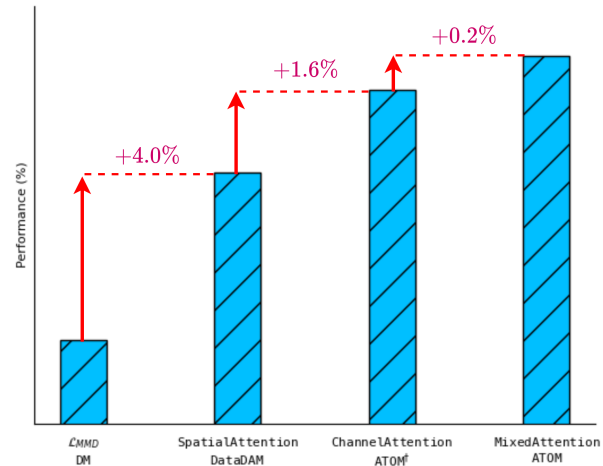


Figure 1. The ATOM Framework utilizes inherent information to capture both context and location, resulting in significantly improved performance in dataset distillation. We display the performance of various components within the ATOM framework, showcasing a 5.8% enhancement from the base distribution matching performance on CIFAR10 at IPC50. Complete numerical details can be found in Table 4.

ing and inferencing pipelines [24, 47–49, 58, 66, 70, 71]. This growth can be attributed to the escalating complexity of model architectures and the ever-expanding scale of datasets. Despite the increasing computational burden, two distinct approaches have emerged as potential avenues for addressing this issue: the model-centric and data-centric approaches. The model-centric approach is primarily concerned with mitigating computational costs by refining the architecture of deep learning models. Techniques such as pruning, quantization, knowledge distillation, and architectural simplification are key strategies employed within this paradigm [24, 27, 28, 60, 63, 66]. In contrast, the data-centric approach adopts a different perspective, focusing on exploring and leveraging the inherent redundancy within datasets. Rather than modifying model architectures, this

\*Equal contribution

approach seeks to identify or construct a smaller dataset that retains the essential information necessary for maintaining performance levels. Coreset selection was a fairly adopted method for addressing this gap [3, 5, 45, 50, 55]. In particular works such as Herding [61] and K-Center [50] offered a heuristic-based approach to intelligently select an informative subset of data. However, as a heuristic-based method, the downstream performance is limited by the information contained solely in the subset. More recently, shapely data selection [15] found the optimal subset of data by measuring the downstream performance for every subset combination achievable in the dataset. However inefficient this may be, the downstream performance is still limited by the diversity of samples selected. therefore, Dataset Distillation (DD) [58] has emerged as a front-runner wherein a synthetic dataset can be learned.

Dataset distillation aims to distill large-scale datasets into a smaller representation, such that downstream models trained on this condensed dataset will retain competitive performance with those trained on the larger original one [6, 58, 71]. Recently, many techniques have been introduced to address this challenge, including gradient matching [36, 69, 71], feature/distribution matching [47, 70, 72], and trajectory matching [6, 12, 19]. However, many of these methods suffer from complex and computationally heavy distillation pipelines [6, 19, 71] or inferior performance [47, 70, 71]. A promising approach, DataDAM [47], effectively tackled the computational challenges present in prior distillation techniques by employing untrained neural networks, in contrast to bi-level optimization methods. However, despite its potential, DataDAM faced several significant limitations: (1) it obscured relevant class-content-based information existing channel-wise in intermediate layers; (2) it only achieved marginal enhancements on previous dataset distillation algorithms; and (3) it exhibited inferior cross-architecture generalization.

In this work, we introduce **ATtentiOn Mixer**, dubbed **ATOM** as an efficient dataset distillation pipeline that strikes an impressive balance between computational efficiency and superior performance. Drawing upon spatial attention matching techniques from prior studies like DataDAM [47], we expand our receptive field of information in the matching process. Our key contribution lies in mixing spatial information with channel-wise contextual information. Intuitively, different convolutional filters focus on different localizations of the input feature; thus, channel-wise attention aids in the distillation matching process by compressing and aggregating information from multiple regions as evident by the performance improvemnets displayed in Figure 1. ATOM not only combines localization and context, but it also produces distilled images that are more generalizable to various downstream architectures, implying that the distilled features are true representations of the original

dataset. Moreover, our approach demonstrates consistent improvements across all settings on a comprehensive distillation test suite. In summary, the key contributions of this study can be outlined as follows:

**[C1]** We provide further insight into the intricacies of attention matching, ultimately introducing the use of channel-wise attention matching for capturing a higher level of information in the feature-matching process. Our mixing module combines both spatial localization awareness of a particular class, with distinctive contextual information derived channel-wise.

**[C2]** Empirically we show superior performance against previous dataset distillation methods including feature matching and attention matching works, without bi-level optimization on common computer vision datasets.

**[C3]** We extend our findings by demonstrating superior performance in cross-architecture and neural architecture search. In particular, we provide a channel-only setting that maintains the majority of the performance while incurring a lower computational cost.

## 2. Related Works

**Coreset Selection.** Coreset selection, an early data-centric approach, aimed to efficiently choose a representative subset from a full dataset to enhance downstream training performance and efficiency. Various methods have been proposed in the past, including geometry-based approaches [1, 9, 50, 52, 61], loss-based techniques as mentioned in [44, 54], decision-boundary-focused methods [14, 40], bilevel optimization strategies [30, 31], and gradient-matching algorithms outlined in [29, 41]. Notable among them are Random, which randomly selects samples as the coreset; Herding, which picks samples closest to the cluster center; K-Center, which selects multiple center points to minimize the maximum distance between data points and their nearest center; and Forgetting, which identifies informative training samples based on learning difficulties [3, 5, 50, 54]. While these selection-based methods have shown moderate success in efficient training, they inherently possess limitations in capturing rich information. Since each image in the selected subset is treated independently, they lack the rich features that could have been captured if the diversity within classes had been considered. These limitations have motivated the emergence of dataset distillation within the field.

**Dataset Distillation.** Dataset distillation has emerged as a learnable method of synthesizing a smaller, information-rich dataset from a large-scale real dataset. This approach offers a more efficient training paradigm, commonly applied in various downstream applications such as continual learning [8, 18, 47, 65, 71], neural architecture search [25, 53], and federated learning [26, 37, 38, 64]. The seminal work, initially proposed by Wang et al. [58], introduced

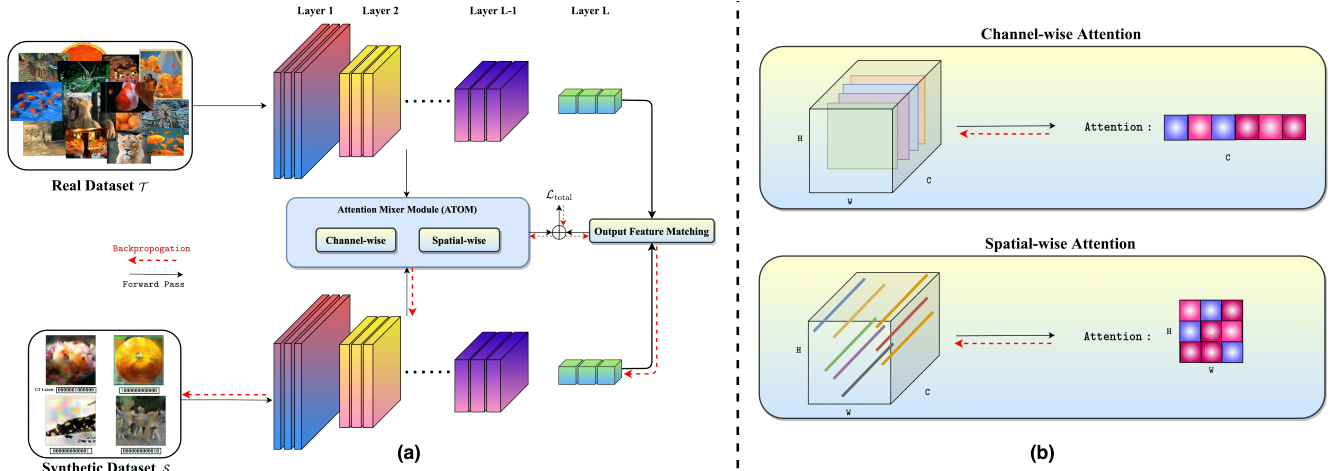


Figure 2. (a) An overview of the proposed ATOM framework. By mixing attention, ATOM is able to capture both spatial localization and class context. (b) Demonstration of the internal architecture for spatial- and channel-wise attention in the ATOM Module. The spatial-wise attention computes attention at specific locales through different filters, resulting in a matrix output, whereas the channel-wise attention calculates attention between each filter, naturally producing a vectorized output.

bilevel optimization, comprising an outer loop for learning the pixel-level synthetic dataset and an inner loop for training the matching network. Following this, several studies adopted surrogate objectives to tackle unrolled optimization problems in meta-learning. For example, gradient matching methods [13, 32, 36, 69, 71] learn images by aligning network gradients derived from real and synthetic datasets. Trajectory matching [6, 10, 12, 19] improves performance by minimizing differences in model training trajectories between original and synthetic samples. Meanwhile, feature matching strategies [47, 47, 56, 68, 70, 72] aim to align feature distributions between real and synthetic data within diverse latent spaces. Despite significant advancements in this field, methods still struggle to find a trade-off between the computational costs associated with the distillation pipeline and the model’s performance. A recent work, DataDAM [47], used spatial attention to improve the performance of feature-matching-based methods by selectively matching features based on their spatial attention scores. However, although this method operates without bilevel optimization, it only marginally improves performance on larger test suites. In this study, we delve deeper into the potential of attention-based methods and demonstrate superior performance compared to DataDAM and previous benchmarks across various computer vision datasets. Additionally, we achieve a lower computational cost compared to conventional attention-matching approaches by leveraging information in a channel-wise manner.

**Attention Mechanism.** Attention mechanisms have been widely adopted in deep learning to enhance performance across various tasks [2, 59, 67]. Initially applied in natural language processing [2], it has extended to com-

puter vision, with global attention models [59] improving image classification and convolutional block attention modules [62] enhancing feature map selection. Additionally, attention aids model compression in knowledge distillation [67]. They are lauded for their ability to efficiently incorporate global contextual information into feature representations. When applied to feature maps, attention can take the form of either spatial or channel-based methods. Spatial methods focus on identifying the informative regions (“where”), while channel-based methods complementarily emphasize the informative features (“what”). Both spatial localization and channel information are crucial for identifying class characteristics. Recently, Sajedi *et al.* proposed DataDAM [47] to concentrate only on spatial attention, capturing class correlations within image localities for efficient training purposes. However, inspired by the inherent obfuscation of the content in the attention maps, we propose an Attention Mixer module that uses a unique combination of spatial and channel-wise attention to capture localization and information content.

### 3. Methodology

Given the larger source dataset  $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{T}|}$  containing  $|\mathcal{T}|$  real image-label pairs, we generate a smaller learnable synthetic dataset  $\mathcal{S} = \{(s_j, y_j)\}_{j=1}^{|\mathcal{S}|}$  with  $|\mathcal{S}|$  synthetic image and label pairs. Following previous works [6, 47, 56, 69, 71], we use random sampling to initialize our synthetic dataset. For every class  $k$ , we obtain a batch of real and synthetic data ( $B_k^T$  and  $B_k^S$ , respectively) and use a neural network  $\phi_\theta(\cdot)$  with randomly initialized weights  $\theta$  [20] to extract intermediate and output features.

We illustrate our method in Figure 2 where an  $L$ -layer neural network  $\phi_\theta(\cdot)$  is used to extract features from the real and synthetic sets. The collection of feature maps from the real and synthetic sets can be expressed as  $\phi_\theta(\mathcal{T}_k) = [\mathbf{f}_{\theta,1}^{\mathcal{T}_k}, \dots, \mathbf{f}_{\theta,L}^{\mathcal{T}_k}]$  and  $\phi_\theta(\mathcal{S}_k) = [\mathbf{f}_{\theta,1}^{\mathcal{S}_k}, \dots, \mathbf{f}_{\theta,L}^{\mathcal{S}_k}]$ , respectively. The feature  $\mathbf{f}_{\theta,l}^{\mathcal{T}_k}$  comprises a multi-dimensional array within  $\mathbb{R}^{|B_k^{\mathcal{T}}| \times C_l \times W_l \times H_l}$ , obtained from the real dataset at the  $l^{\text{th}}$  layer, where  $C_l$  denotes the number of channels and  $H_l \times W_l$  represents the spatial dimensions. Correspondingly, a feature  $\mathbf{f}_{\theta,l}^{\mathcal{S}_k}$  is derived for the synthetic dataset.

We now introduce the Attention Mixer Module (ATOM) which generates attention maps for the intermediate features derived from both the real and synthetic datasets. Leveraging a feature-based mapping function  $A(\cdot)$ , ATOM takes the intermediate feature maps as input and produces a corresponding attention map for each feature. Formally, we express this as:  $A(\phi_\theta(\mathcal{T}_k)) = [\mathbf{a}_{\theta,1}^{\mathcal{T}_k}, \dots, \mathbf{a}_{\theta,L-1}^{\mathcal{T}_k}]$  and  $A(\phi_\theta(\mathcal{S}_k)) = [\mathbf{a}_{\theta,1}^{\mathcal{S}_k}, \dots, \mathbf{a}_{\theta,L-1}^{\mathcal{S}_k}]$  for the real and synthetic sets, respectively. Previous works [47, 67] have shown that spatial attention, which aggregates the absolute values of feature maps across the channel dimension, can emphasize common spatial locations associated with high neuron activation. The implication of this is retaining the most informative regions, thus generating an efficient feature descriptor. In this work, we also consider the effect of channel-wise attention, which emphasizes the most significant information captured by each channel based on the magnitude of its activation. Since different filters explore different regions or locations of the input feature, channel-wise activation yields the best aggregation of the global information. Ultimately, we convert the feature map  $\mathbf{f}_{\theta,l}^{\mathcal{T}_k}$  of the  $l^{\text{th}}$  layer into an attention map  $\mathbf{a}_{\theta,l}^{\mathcal{T}_k}$  representing spatial or channel-wise attention using the corresponding mapping functions  $A_s(\cdot)$  or  $A_c(\cdot)$  respectively. Formally, we can denote the spatial and channel-wise attention maps as:

$$A_s(\mathbf{f}_{\theta,l}^{\mathcal{T}_k}) = \sum_{i=1}^{C_l} |(\mathbf{f}_{\theta,l}^{\mathcal{T}_k})_i|^{p_s}, \quad (1)$$

$$A_c(\mathbf{f}_{\theta,l}^{\mathcal{T}_k}) = \sum_{i=1}^{H_l \times W_l} |(\mathbf{f}_{\theta,l}^{\mathcal{T}_k})_i^*|^{p_c}, \quad (2)$$

where,  $(\mathbf{f}_{\theta,l}^{\mathcal{T}_k})_i = \mathbf{f}_{\theta,l}^{\mathcal{T}_k}(:, i, :, :)$  is the feature map of channel  $i$  from the  $l^{\text{th}}$  layer, and the power and absolute value operations are applied element-wise; meanwhile, the symbol  $*$  flattens the feature map along the spatial dimension  $((\mathbf{f}_{\theta,l}^{\mathcal{T}_k})^* \in \mathbb{R}^{|B_k^{\mathcal{T}}| \times C_l \times W_l \times H_l})$ , such that  $(\mathbf{f}_{\theta,l}^{\mathcal{T}_k})_i^* = (\mathbf{f}_{\theta,l}^{\mathcal{T}_k})^*(:, :, i)$ . By leveraging both types of attention, we can better encapsulate the relevant information in the intermediate features, as investigated in Section 4.3. Further, the

effect of power parameters for spatial and channel-wise attention, *i.e.*  $p_s$  and  $p_c$  is studied in the Section 4.3.

Given our generated spatial and channel attention maps for the intermediate features, we apply standard normalization such that we can formulate a matching loss between the synthetic and real datasets. We denote our generalized loss  $\mathcal{L}_{\text{ATOM}}$  as:

$$\mathbb{E}_{\theta \sim P_\theta} \left[ \sum_{k=1}^K \sum_{l=1}^{L-1} \left\| \mathbb{E}_{\mathcal{T}_k} \left[ \frac{\mathbf{z}_{\theta,l}^{\mathcal{T}_k}}{\|\mathbf{z}_{\theta,l}^{\mathcal{T}_k}\|_2} \right] - \mathbb{E}_{\mathcal{S}_k} \left[ \frac{\mathbf{z}_{\theta,l}^{\mathcal{S}_k}}{\|\mathbf{z}_{\theta,l}^{\mathcal{S}_k}\|_2} \right] \right\|^2 \right], \quad (3)$$

where, in the case of spatial attention, we denote  $\mathbf{z}_{\theta,l}^{\mathcal{T}_k} = \text{vec}(\mathbf{a}_{\theta,l}^{\mathcal{T}_k}) \in \mathbb{R}^{|B_k^{\mathcal{T}}| \times (W_l \times H_l)}$  and  $\mathbf{z}_{\theta,l}^{\mathcal{S}_k} = \text{vec}(\mathbf{a}_{\theta,l}^{\mathcal{S}_k}) \in \mathbb{R}^{|B_k^{\mathcal{S}}| \times (W_l \times H_l)}$  to represent the vectorized spatial attention map pairs at the  $l^{\text{th}}$  layer for the real and synthetic datasets, respectively. Meanwhile, for channel-based attention, we have  $\mathbf{z}_{\theta,l}^{\mathcal{T}_k} = \text{vec}(\mathbf{a}_{\theta,l}^{\mathcal{T}_k}) \in \mathbb{R}^{|B_k^{\mathcal{T}}| \times (C_l)}$  and  $\mathbf{z}_{\theta,l}^{\mathcal{S}_k} = \text{vec}(\mathbf{a}_{\theta,l}^{\mathcal{S}_k}) \in \mathbb{R}^{|B_k^{\mathcal{S}}| \times (C_l)}$  to represent the flattened channel attention map pairs at the  $l^{\text{th}}$  layer for the real and synthetic datasets, respectively. The parameter  $K$  is the number of categories in a dataset, and  $P_\theta$  denotes the distribution of network parameters. We estimate the expectation terms in Equation (3) empirically if ground-truth data distributions are not available.

Following previous works [47, 56, 68, 70, 72], we leverage the features in the final layer to regularize our matching process. In particular, the features of the penultimate layer represent a high-level abstraction of information from the input images in an embedded representation and can thus be used to inject semantic information in the matching process [17, 46, 47, 70]. Thus, we employ  $\mathcal{L}_{\text{MMD}}$  as described in [47, 70] out-of-the-box.

Finally, we learn the synthetic dataset by minimizing the following optimization problem using SGD optimizer:

$$S^* = \arg \min_S (\mathcal{L}_{\text{ATOM}} + \lambda \mathcal{L}_{\text{MMD}}), \quad (4)$$

where  $\lambda$  is the task balance parameter inherited from [47]. In particular, we highlight that  $\mathcal{L}_{\text{MMD}}$  brings semantic information from the final layer, while  $\mathcal{L}_{\text{ATOM}}$  mixes the spatial and channel-wise attention information from the intermediate layers. Note that our approach assigns a fixed label to each synthetic sample and keeps it constant during training. A summary of the learning algorithm can be found in Algorithm 1.

## 4. Experiments

### 4.1. Experimental Setup

**Datasets.** Our method is evaluated on the CIFAR-10 and CIFAR-100 datasets [33], which maintain a resolution of  $32 \times 32$ , aligning with state-of-the-art benchmarks. Furthermore, we resize the Tiny ImageNet [35] datasets to 64



---

**Algorithm 1** Attention Mixer for Dataset Distillation

---

**Input:** Real training dataset  $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{|\mathcal{T}|}$

**Required:** Initialized synthetic samples for  $K$  classes, Deep neural network  $\phi_\theta$  parameterized with  $\theta$ , Probability distribution over randomly initialized weights  $P_\theta$ , Learning rate  $\eta_S$ , Task balance parameter  $\lambda$ , Number of training iterations  $I$ .

- 1: Initialize synthetic dataset  $\mathcal{S}$
- 2: **for**  $i = 1, 2, \dots, I$  **do**
- 3:   Sample  $\theta$  from  $P_\theta$
- 4:   Sample mini-batch pairs  $B_k^T$  and  $B_k^S$  from the real and synthetic sets for each class  $k$
- 5:   Compute  $\mathcal{L}_{\text{ATOM}}$  and  $\mathcal{L}_{\text{MMD}}$
- 6:   Calculate  $\mathcal{L} = \mathcal{L}_{\text{ATOM}} + \lambda \mathcal{L}_{\text{MMD}}$
- 7:   Update the synthetic dataset using  $\mathcal{S} \leftarrow \mathcal{S} - \eta_S \nabla_{\mathcal{S}} \mathcal{L}$
- 8: **end for**

**Output:** Synthetic dataset  $\mathcal{S} = \{(\mathbf{s}_i, y_i)\}_{i=1}^{|\mathcal{S}|}$

---

$\times 64$  for additional experimentation. The supplementary materials provide more detailed dataset information.

**Network Architectures.** We employ a ConvNet architecture [16] for distillation, following prior studies. The default ConvNet comprises three convolutional blocks, each consisting of a 128-kernel  $3 \times 3$  convolutional layer, instance normalization, ReLU activation, and  $3 \times 3$  average pooling with a stride of 2. To accommodate the increased resolutions in Tiny ImageNet, we append a fourth convolutional block. Network parameters are initialized using normal initialization [20] in all experiments.

**Evaluation Protocol.** We evaluate the methods using standard measures from previous studies [47, 56, 69–71]. Five sets of synthetic images are generated from a real training dataset with 1, 10, and 50 images per class. Then, 20 neural network models are trained on each synthetic set using an SGD optimizer with a fixed learning rate of 0.01. Each experiment reports the mean and standard deviation values for 100 models to assess the efficacy of distilled datasets. Furthermore, computational costs are assessed by calculating run-time per step over 100 iterations, as well as peak GPU memory usage during 100 iterations of training.

**Implementation Details.** We use the SGD optimizer with a fixed learning rate of 1 to learn synthetic datasets containing 1, 10, and 50 IPCs over 8000 iterations with task balances ( $\lambda$ ) set at 0.01. Previous works have shown that  $p_s = 4$  is sufficient for spatial attention matching [47]. As such we set our default case as:  $p_c = p_s = 4$ . This is further ablated in Section 4.3. We adopt differentiable augmentation for both training and evaluating the synthetic set, following [47, 71]. For dataset reprocessing, we utilized the Kornia implementation of Zero Component Analysis (ZCA) with default parameters, following previous works

[6, 42, 47]. All experiments are performed on a single A100 GPU with 80 GB of memory. Further hyperparameter details can be found in the supplementary materials.

**Competitive Methods.** In this paper, we compare the empirical results of ATOM on three computer vision datasets: CIFAR10/100 and TinyImageNet. We evaluate ATOM against four corset selection approaches and thirteen distillation methods for training set synthesis. The corset selection methods include Random selection [45], Herding [3, 5], K-Center [50], and Forgetting [55]. We also compare our approach with state-of-the-art distillation methods, including Dataset Distillation [58] (DD), Flexible Dataset Distillation [4] (LD), Dataset Condensation [71] (DC), Dataset Condensation with Contrastive (DCC) [36], Dataset Condensation with Differentiable Siamese Augmentation [69] (DSA), Distribution Matching [70] (DM), Deep Generative Priors (GLaD), Aligning Features [56] (CAFE), VIG [39], Kernel Inducing Points [42, 43] (KIP), Matching Training Trajectories [6] (MTT), and Attention Matching [47] (DAM).

## 4.2. Comparison with State-of-the-art Methods

**Performance Comparison.** In this section, we present a comparative analysis of our method against coresets and dataset distillation approaches. ATOM consistently outperforms these studies, especially at smaller distillation ratios, as shown in Table 1. Since the goal of dataset distillation is to generate a more compact synthetic set, we emphasize our significant performance improvements at low IPCs. We achieve almost 4% improvement over the previous attention matching framework [47], DataDAM when evaluated on CIFAR-100 at IPC1. Notably, our performance on CIFAR-100 at IPC50 is 50.2% – that is nearly 90% of the baseline accuracy at a mere 10% of the original dataset. These examples motivate the development of dataset distillation works as downstream models can achieve relatively competitive performance with their baselines at a fraction of the training costs. Our primary objective in this study is to investigate the impact of channel-wise attention within the feature-matching process. Compared to prior attention-based and feature-based methodologies, our findings underscore the significance of channel-wise attention and the ATOM module, as validated also in the ablation studies in Section 4.3.

**Cross-architecture Generalization.** In this section, we assess the generalization capacity of our refined dataset by training various unseen deep neural networks on it and then evaluating their performance on downstream classification tasks. Following established benchmarks [47, 56, 70, 71], we examine classic CNN architectures such as AlexNet [34], VGG-11 [51], ResNet-18 [21], and additionally, a standard Vision Transformer (ViT) [11]. Specifically, we utilize synthetic images learned from CIFAR-10 with IPC50 using ConvNet as the reference model and subsequently

| Dataset<br>IPC<br>Ratio % | CIFAR-10                       |                                |                                | CIFAR-100                      |                                |                                | Tiny ImageNet                 |                                |                                |
|---------------------------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|-------------------------------|--------------------------------|--------------------------------|
|                           | 1                              | 10                             | 50                             | 1                              | 10                             | 50                             | 1                             | 10                             | 50                             |
|                           | 0.02                           | 0.2                            | 1                              | 0.2                            | 2                              | 10                             | 0.2                           | 2                              | 10                             |
| 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                 | 30.0 $\pm$ 0.4                 | 1.4 $\pm$ 0.1                 | 5.0 $\pm$ 0.2                  | 15.0 $\pm$ 0.4                 |
| Herdling [61]             | 21.5 $\pm$ 1.2                 | 31.6 $\pm$ 0.7                 | 40.4 $\pm$ 0.6                 | 8.3 $\pm$ 0.3                  | 17.3 $\pm$ 0.3                 | 33.7 $\pm$ 0.5                 | 2.8 $\pm$ 0.2                 | 6.3 $\pm$ 0.2                  | 16.7 $\pm$ 0.3                 |
| K-Center [50]             | 21.5 $\pm$ 1.3                 | 14.7 $\pm$ 0.9                 | 27.0 $\pm$ 1.4                 | 8.4 $\pm$ 0.3                  | 17.3 $\pm$ 0.3                 | 30.5 $\pm$ 0.3                 | -                             | -                              | -                              |
| Forgetting [54]           | 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                 | 5.1 $\pm$ 0.2                  | 15.0 $\pm$ 0.3                 |
| DD <sup>†</sup> [58]      | -                              | 36.8 $\pm$ 1.2                 | -                              | -                              | -                              | -                              | -                             | -                              | -                              |
| LD <sup>†</sup> [4]       | 25.7 $\pm$ 0.7                 | 38.3 $\pm$ 0.4                 | 42.5 $\pm$ 0.4                 | 11.5 $\pm$ 0.4                 | -                              | -                              | -                             | -                              | -                              |
| DC [71]                   | 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                 | 30.6 $\pm$ 0.6                 | 5.3 $\pm$ 0.1                 | 12.9 $\pm$ 0.1                 | 12.7 $\pm$ 0.4                 |
| DCC [36]                  | 32.9 $\pm$ 0.8                 | 49.4 $\pm$ 0.5                 | 61.6 $\pm$ 0.4                 | 13.3 $\pm$ 0.3                 | 30.6 $\pm$ 0.4                 | -                              | -                             | -                              | -                              |
| DSA [69]                  | 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                 | 42.8 $\pm$ 0.4                 | 5.7 $\pm$ 0.1                 | 16.3 $\pm$ 0.2                 | 15.1 $\pm$ 0.2                 |
| DM [70]                   | 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                 | 43.6 $\pm$ 0.4                 | 3.9 $\pm$ 0.2                 | 12.9 $\pm$ 0.4                 | 25.3 $\pm$ 0.2                 |
| GLaD [7]                  | 28.0 $\pm$ 0.8                 | 46.7 $\pm$ 0.5                 | 59.9 $\pm$ 0.7                 | -                              | -                              | -                              | -                             | -                              | -                              |
| CAFE [56]                 | 30.3 $\pm$ 1.1                 | 46.3 $\pm$ 0.6                 | 55.5 $\pm$ 0.6                 | 12.9 $\pm$ 0.3                 | 27.8 $\pm$ 0.3                 | 37.9 $\pm$ 0.3                 | -                             | -                              | -                              |
| CAFE+DSA [56]             | 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                 | 42.9 $\pm$ 0.2                 | -                             | -                              | -                              |
| VIG [39]                  | 26.5 $\pm$ 1.2                 | 54.6 $\pm$ 0.1                 | 35.6 $\pm$ 0.6                 | 17.8 $\pm$ 0.1                 | 29.3 $\pm$ 0.1                 | -                              | -                             | -                              | -                              |
| KIP [42]                  | 29.8 $\pm$ 1.0                 | 46.1 $\pm$ 0.7                 | 53.2 $\pm$ 0.7                 | 12.0 $\pm$ 0.2                 | 29.0 $\pm$ 0.3                 | -                              | -                             | -                              | -                              |
| MTT [6]                   | 31.9 $\pm$ 1.2                 | 56.4 $\pm$ 0.7                 | 65.9 $\pm$ 0.6                 | 13.8 $\pm$ 0.6                 | 33.1 $\pm$ 0.4                 | 42.9 $\pm$ 0.3                 | 6.2 $\pm$ 0.4                 | 17.3 $\pm$ 0.2                 | 26.5 $\pm$ 0.3                 |
| DAM [47]                  | 32.0 $\pm$ 1.2                 | 54.2 $\pm$ 0.8                 | 67.0 $\pm$ 0.4                 | 14.5 $\pm$ 0.5                 | 34.8 $\pm$ 0.5                 | 49.4 $\pm$ 0.3                 | 8.3 $\pm$ 0.4                 | 18.7 $\pm$ 0.3                 | 28.7 $\pm$ 0.3                 |
| <b>ATOM (Ours)</b>        | <b>34.8<math>\pm</math>1.0</b> | <b>57.9<math>\pm</math>0.7</b> | <b>68.8<math>\pm</math>0.5</b> | <b>18.1<math>\pm</math>0.4</b> | <b>35.7<math>\pm</math>0.4</b> | <b>50.2<math>\pm</math>0.3</b> | <b>9.1<math>\pm</math>0.2</b> | <b>19.5<math>\pm</math>0.4</b> | <b>29.1<math>\pm</math>0.3</b> |
| Full Dataset              | 84.8 $\pm$ 0.1                 |                                |                                | 56.2 $\pm$ 0.3                 |                                |                                | 37.6 $\pm$ 0.4                |                                |                                |

Table 1. Comparison with previous dataset distillation methods on CIFAR-10, CIFAR-100 and Tiny ImageNet. The works DD<sup>†</sup> and LD<sup>†</sup> use AlexNet [34] for CIFAR-10 dataset. All other methods use ConvNet for training and evaluation. **Bold entries** are the best results.

train the aforementioned networks on the refined dataset to assess their performance on downstream tasks. The results, as depicted in Table 2, indicate that ATOM demonstrates superior generalization across a spectrum of architectures. Notably, it achieves a significant performance boost of over 4% compared to the prior state-of-the-art on ResNet-18 [21]. This implies that the channel-wise attention mechanism effectively identifies features not only relevant to ConvNet but also to a wider range of deep neural networks, thereby enhancing the refined dataset with this discerned information.

|                    | ConvNet                        | AlexNet                        | VGG-11                         | ResNet-18                      | ViT                            | Avg.                           |
|--------------------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|
| DC [71]            | 53.9 $\pm$ 0.5                 | 28.8 $\pm$ 0.7                 | 38.8 $\pm$ 1.1                 | 20.9 $\pm$ 1.0                 | 30.1 $\pm$ 0.5                 | 34.5 $\pm$ 0.8                 |
| CAFE [56]          | 62.3 $\pm$ 0.4                 | 43.2 $\pm$ 0.4                 | 48.8 $\pm$ 0.5                 | 43.3 $\pm$ 0.7                 | 22.7 $\pm$ 0.7                 | 44.1 $\pm$ 0.5                 |
| DSA [69]           | 60.6 $\pm$ 0.5                 | 53.7 $\pm$ 0.6                 | 51.4 $\pm$ 1.0                 | 47.8 $\pm$ 0.9                 | 43.3 $\pm$ 0.4                 | 51.4 $\pm$ 0.7                 |
| DM [70]            | 63.0 $\pm$ 0.4                 | 60.1 $\pm$ 0.5                 | 57.4 $\pm$ 0.8                 | 52.9 $\pm$ 0.4                 | 45.2 $\pm$ 0.4                 | 55.7 $\pm$ 0.5                 |
| KIP [42]           | 56.9 $\pm$ 0.4                 | 53.2 $\pm$ 1.6                 | 53.2 $\pm$ 0.5                 | 47.0 $\pm$ 0.8                 | 18.3 $\pm$ 0.6                 | 45.8 $\pm$ 0.8                 |
| MTT [6]            | 66.2 $\pm$ 0.6                 | 43.9 $\pm$ 0.9                 | 48.7 $\pm$ 1.3                 | 60.0 $\pm$ 0.7                 | 47.7 $\pm$ 0.6                 | 53.3 $\pm$ 0.8                 |
| DAM [47]           | 67.0 $\pm$ 0.4                 | 63.9 $\pm$ 0.9                 | 64.8 $\pm$ 0.5                 | 60.2 $\pm$ 0.7                 | 48.2 $\pm$ 0.8                 | 60.8 $\pm$ 0.7                 |
| <b>ATOM (Ours)</b> | <b>68.8<math>\pm</math>0.4</b> | <b>64.1<math>\pm</math>0.7</b> | <b>66.4<math>\pm</math>0.6</b> | <b>64.5<math>\pm</math>0.6</b> | <b>49.5<math>\pm</math>0.7</b> | <b>62.7<math>\pm</math>0.6</b> |

Table 2. Cross-architecture testing performance (%) on CIFAR-10 with 50 images per class. The ConvNet architecture is employed for distillation. **Bold entries** are the best results.

**Distillation Cost Analysis.** In this section, we delve into an examination of the training costs required for the distillation process. Although the main goal of dataset distillation is to reduce training costs across different applications such as neural architecture search and continual learning, the distillation technique itself must be efficient, enabling smooth operation on consumer-grade hardware. Approaches such as DC, DSA, and MTT introduce additional computational

| Method                         | Run Time (Sec.) |                 |                  | GPU memory (MB) |       |       |
|--------------------------------|-----------------|-----------------|------------------|-----------------|-------|-------|
|                                | IPC1            | IPC10           | IPC50            | IPC1            | IPC10 | IPC50 |
| DC [71]                        | 0.16 $\pm$ 0.01 | 3.31 $\pm$ 0.02 | 15.74 $\pm$ 0.10 | 3515            | 3621  | 4527  |
| DSA [69]                       | 0.22 $\pm$ 0.02 | 4.47 $\pm$ 0.12 | 20.13 $\pm$ 0.58 | 3513            | 3639  | 4539  |
| DM [70]                        | 0.08 $\pm$ 0.02 | 0.08 $\pm$ 0.02 | 0.08 $\pm$ 0.02  | 3323            | 3455  | 3605  |
| MTT [6]                        | 0.36 $\pm$ 0.23 | 0.40 $\pm$ 0.20 | OOM              | 2711            | 8049  | OOM   |
| DAM [47]                       | 0.09 $\pm$ 0.01 | 0.08 $\pm$ 0.01 | 0.16 $\pm$ 0.04  | 3452            | 3561  | 3724  |
| <b>ATOM<sup>†</sup> (Ours)</b> | 0.08 $\pm$ 0.02 | 0.08 $\pm$ 0.02 | 0.13 $\pm$ 0.03  | 3152            | 3263  | 4151  |
| <b>ATOM (Ours)</b>             | 0.10 $\pm$ 0.02 | 0.10 $\pm$ 0.01 | 0.17 $\pm$ 0.02  | 3601            | 4314  | 5134  |

Table 3. Comparisons of training time and GPU memory usage for prior dataset distillation methods. Run time is averaged per step over 100 iterations, while GPU memory usage is reported as peak memory during the same 100 iterations of training on an A100 GPU for CIFAR-10. Methods that surpass the GPU memory threshold and fail to run are denoted as OOM (out-of-memory). **ATOM<sup>†</sup>** represents our method with on-channel attention, hence offering a better tradeoff in computational complexity.

overhead due to bi-level optimization and training an expert model. In contrast, our method, akin to DM and DAM, capitalizes on randomly initialized networks, obviating the need for training and thereby reducing the computational cost per step involved in the matching stage. As illustrated in Table 3 utilizing solely the channel-based ATOM<sup>†</sup> decreases the computational burden of matching compared to the default ATOM configuration. This efficiency is crucial, as channel-wise attention offers a more effective distillation process while maintaining superior performance (refer to Section 4.3).

**Convergence Speed Analysis.** In Figure 3, we plot the downstream testing accuracy evolution for the synthetic im-

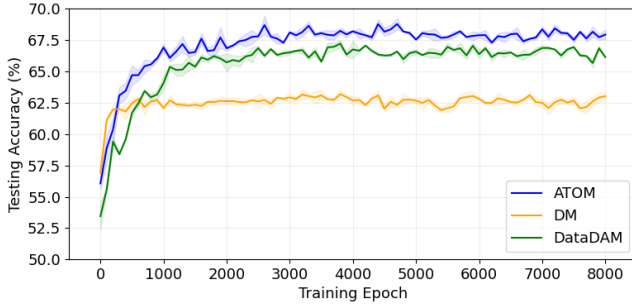


Figure 3. Test accuracy evolution of synthetic image learning on CIFAR10 with IPC50 for ATOM (ours), DM [70] and DataDAM [47].

ages on CIFAR10 IPC50. Comparing with previous methods, DM [70] and DataDAM [47], we can explicitly see an improvement in convergence speed and a significantly higher steady state achieved with the ATOM framework. Our included convergence analysis supports the practicality of our method and the consistency to which we outperform previous baselines.

### 4.3. Ablation Studies and Analysis

**Evaluation of loss components in ATOM.** In Table 4, we evaluate the effect of different attention-matching mechanisms with respect to pure feature matching in intermediate layers and distribution matching in the final layer ( $\mathcal{L}_{\text{MMD}}$ ). The results clearly demonstrate that attention-matching improves the performance of the distillation process. In particular, the attention-matching process improves feature matching by 8.0%. Further, it seems that channel attention is able to capture the majority of relevant information from the intermediate features, as evidenced by an improvement of over 1.5% from spatial attention matching. Ultimately, this provides an incentive to favor channel attention in the distillation process.

| $\mathcal{L}_{\text{MMD}}$ | Feature Map | Spatial Attn. | Channel Attn. | Performance (%)                |
|----------------------------|-------------|---------------|---------------|--------------------------------|
| ✓                          | -           | -             | -             | 63.0 $\pm$ 0.4                 |
| ✓                          | ✓           | -             | -             | 60.8 $\pm$ 0.6                 |
| ✓                          | -           | ✓             | -             | 67.0 $\pm$ 0.7                 |
| ✓                          | -           | -             | ✓             | 68.6 $\pm$ 0.3                 |
| ✓                          | -           | ✓             | ✓             | <b>68.8<math>\pm</math>0.5</b> |

Table 4. Evaluation of loss components and attention components in ATOM using CIFAR-10 with IPC50.

**Evaluating attention balance in ATOM.** In this section, we evaluate the balance between spatial and channel-wise attention through the power value  $p$ . Referencing Equation (1) and Equation (2), modulating the values of  $p_s$  and  $p_c$  ultimately affects the balance of spatial and channel-wise attention in  $\mathcal{L}_{\text{ATOM}}$ . In Table 5, we examine the impact of different exponentiation powers  $p$  in the attention-

matching mechanisms. Specifically, we conduct a grid-based search to investigate how varying the exponentiation of spatial ( $p_s$ ) and channel ( $p_c$ ) attention influences subsequent performance. Our findings reveal that optimal performance (nearly 1% improvement over our default) occurs when the exponentiation for channel attention significantly exceeds that of spatial attention. This suggests that assigning a higher exponential value places greater emphasis on channel-attention matching over spatial-wise matching. This aligns with our observations from the loss component ablation, where channel-wise matching was found to encapsulate the majority of information within the feature map. Consequently, we deduce that prioritizing channel-wise matching will enhance downstream performance outcomes.

| Channel Attention $p_c$ | Spatial Attention $p_s$ |       |       |       |
|-------------------------|-------------------------|-------|-------|-------|
|                         | 1                       | 2     | 4     | 8     |
| 1                       | 57.4%                   | 57.5% | 57.0% | 56.2% |
| 2                       | 58.2%                   | 57.5% | 57.2% | 56.3% |
| 4                       | 58.4%                   | 58.5% | 57.9% | 57.6% |
| 8                       | <b>58.8%</b>            | 58.7% | 58.2% | 57.8% |

Table 5. Evaluation of power values in the spatial and channel attention computations for  $\mathcal{L}_{\text{ATOM}}$  using CIFAR-10 with IPC10.

**Visualization of Synthetic Images.** We include samples of our distilled images in Figure 4. The images appear to be interleaved with artifacts that assimilate the background and object information into a mixed collage-like appearance. The synthetic images effectively capture the correlation between background and object elements, suggesting their potential for generalizability across various architectures, as empirically verified in Table 2. Additional visualizations are available in the supplementary material.

### 4.4. Applications

**Neural Architecture Search.** In Table 6 we leverage our distilled synthetic datasets as proxy sets to accelerate Neural Architecture Search. In line with previous state-of-the-art, [47, 69, 71], we outline our architectural search space, comprising 720 ConvNets on the CIFAR-10 dataset. We commence with a foundational ConvNet and devise a consistent grid, varying in depth  $D \in \{1, 2, 3, 4\}$ , width  $W \in \{32, 64, 128, 256\}$ , activation function  $A \in \{\text{Sigmoid}, \text{ReLU}, \text{LeakyReLU}\}$ , normalization technique  $N \in \{\text{None}, \text{BatchNorm}, \text{LayerNorm}, \text{InstanceNorm}, \text{GroupNorm}\}$ , and pooling operation  $P \in \{\text{None}, \text{MaxPooling}, \text{AvgPooling}\}$ . Additionally, we benchmark our approach against several state-of-the-art methods, including Random, DSA [71], DM [70], CAFE [56], DAM [47], and Early-Stopping. Our method demonstrates superior performance, accompanied by a heightened Spearman’s correlation (0.75), thereby reinforcing the robustness of ATOM and its potential in neural architecture search.



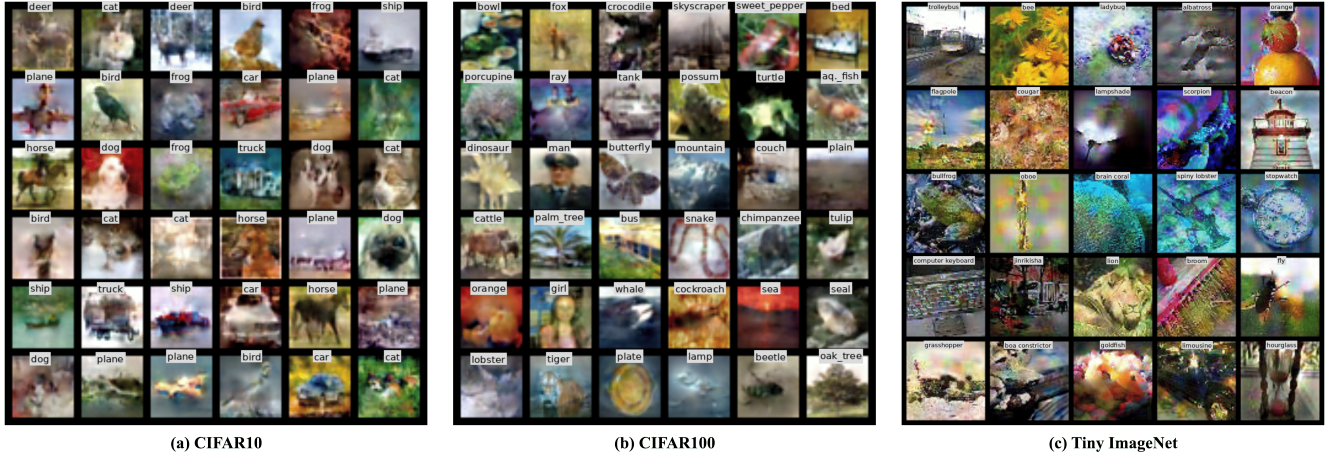


Figure 4. Sample learned synthetic images for CIFAR-10/100 (32×32 resolution) IPC10 and TinyImageNet (64×64 resolution) IPC 1.

|                 | Random     | DSA        | DM         | CAFE       | DAM         | ATOM        | Early-stopping  | Full Dataset    |
|-----------------|------------|------------|------------|------------|-------------|-------------|-----------------|-----------------|
| Performance (%) | 88.9       | 87.2       | 87.2       | 83.6       | <b>89.0</b> | 88.9        | 88.9            | 89.2            |
| Correlation     | 0.70       | 0.66       | 0.71       | 0.59       | 0.72        | <b>0.75</b> | 0.69            | 1.00            |
| Time cost (min) | 206.4      | 206.4      | 206.6      | 206.4      | 206.4       | 206.2       | 206.2           | 5168.9          |
| Storage (imgs)  | <b>500</b> | <b>500</b> | <b>500</b> | <b>500</b> | <b>500</b>  | <b>500</b>  | $5 \times 10^4$ | $5 \times 10^4$ |

Table 6. Neural architecture search on CIFAR-10 with IPC50.

## 5. Limitations

Many studies in dataset distillation encounter a constraint known as *re-distillation costs* [22, 23, 57]. This limitation becomes apparent when adjusting the number of images per class (IPC) or the distillation ratios. Like most other distillation methods, our approach requires re-distillation on the updated setting configuration, which limits flexibility regarding configuration changes and storage allocation. Additionally, we observed in Table 2 that dataset distillation methods often struggle with generalizing to transformer architectures. Despite ATOM outperforming other methods, there is still a noticeable performance drop compared to convolutional neural networks. This suggests that the effectiveness of transformers for downstream training might be constrained by the distilled data.

## 6. Conclusion

In this work, we introduced an Attention Mixer (ATOM) for efficient dataset distillation. Previous approaches have struggled with marginal performance gains, obfuscating channel-wise information, and high computational overheads. ATOM addresses these issues by effectively combining information from different attention mechanisms, facilitating a more informative distillation process with untrained neural networks. Our approach utilizes a broader receptive field to capture spatial information while preserving distinct content information at the channel level, thus better aligning synthetic and real datasets. By capturing information across intermediate layers, ATOM facilitates multi-scale

distillation. We demonstrated the superior performance of ATOM on standard distillation benchmarks and its favorable performance across multiple architectures. We conducted several ablative studies to justify the design choices behind ATOM. Furthermore, we applied our distilled data to Neural Architecture Search, showing a superior correlation with the real large-scale dataset. In the future, we aim to extend attention mixing to various downstream tasks, including image segmentation and localizations. We also hope to address limitations of ATOM, such as re-distillation costs and cross-architecture generalizations on transformers.

## References

- [1] Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 137–153. Springer, 2020. 2
- [2] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations*, 2015. 3
- [3] Eden Belouadah and Adrian Popescu. Scail: Classifier weights scaling for class incremental learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1266–1275, 2020. 2, 5
- [4] Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales. Flexible dataset distillation: Learn labels instead of images. *arXiv preprint arXiv:2006.08572*, 2020. 5, 6
- [5] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 233–248, 2018. 2, 5
- [6] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the*



- IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4750–4759, 2022. 2, 3, 5, 6
- [7] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Generalizing dataset distillation via deep generative prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3739–3748, 2023. 6
- [8] Xuxi Chen, Yu Yang, Zhangyang Wang, and Baharan Mirza-soleiman. Data distillation can be like vodka: Distilling more times for better quality. In *The Twelfth International Conference on Learning Representations*, 2024. 2
- [9] Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, pages 109–116, 2010. 2
- [10] 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. 3
- [11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 5
- [12] Jiawei Du, Yidi Jiang, Vincent YF Tan, Joey Tianyi Zhou, and Haizhou Li. Minimizing the accumulated trajectory error to improve dataset distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3749–3758, 2023. 2, 3
- [13] Jiawei Du, Qin Shi, and Joey Tianyi Zhou. Sequential subset matching for dataset distillation. *Advances in Neural Information Processing Systems*, 36, 2024. 3
- [14] Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018. 2
- [15] Amirata Ghorbani, James Zou, and Andre Esteva. Data shapley valuation for efficient batch active learning. In *2022 56th Asilomar Conference on Signals, Systems, and Computers*, pages 1456–1462. IEEE, 2022. 2
- [16] 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. 5
- [17] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 4
- [18] Jianyang Gu, Kai Wang, Wei Jiang, and Yang You. Summarizing stream data for memory-restricted online continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2024. 2
- [19] Ziyao Guo, Kai Wang, George Cazenavette, HUI LI, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 3
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 3, 5
- [21] 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. 5, 6
- [22] Yang He, Lingao Xiao, and Joey Tianyi Zhou. You only condense once: Two rules for pruning condensed datasets. *Advances in Neural Information Processing Systems*, 36, 2024. 8
- [23] Yang He, Lingao Xiao, Joey Tianyi Zhou, and Ivor Tsang. Multisize dataset condensation. In *The Twelfth International Conference on Learning Representations*, 2024. 8
- [24] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1
- [25] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016. 2
- [26] Yuqi Jia, Saeed Vahidian, Jingwei Sun, Jianyi Zhang, Vyacheslav Kungurtsev, Neil Zhenqiang Gong, and Yiran Chen. Unlocking the potential of federated learning: The symphony of dataset distillation via deep generative latents. *arXiv preprint arXiv:2312.01537*, 2023. 2
- [27] Samir Khaki and Weihai Luo. Cfdp: Common frequency domain pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4714–4723, 2023. 1
- [28] Samir Khaki and Konstantinos N Plataniotis. The need for speed: Pruning transformers with one recipe. *arXiv preprint arXiv:2403.17921*, 2024. 1
- [29] Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pages 5464–5474. PMLR, 2021. 2
- [30] Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glisten: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8110–8118, 2021. 2
- [31] Krishnateja Killamsetty, Xujiang Zhao, Feng Chen, and Rishabh Iyer. Retrieve: Coreset selection for efficient and robust semi-supervised learning. *Advances in Neural Information Processing Systems*, 34:14488–14501, 2021. 2
- [32] 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. 3
- [33] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4
- [34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural net-

- works. *Communications of the ACM*, 60(6):84–90, 2017. 5, 6
- [35] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 4
- [36] 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. 2, 3, 5, 6
- [37] Ping Liu, Xin Yu, and Joey Tianyi Zhou. Meta knowledge condensation for federated learning. In *The Eleventh International Conference on Learning Representations*, 2023. 2
- [38] Songhua Liu, Jingwen Ye, Runpeng Yu, and Xinchao Wang. Slimmable dataset condensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3759–3768, 2023. 2
- [39] 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. 5, 6
- [40] Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 650–663, 2021. 2
- [41] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR, 2020. 2
- [42] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. In *International Conference on Learning Representations*, 2021. 5, 6
- [43] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel-ridge regression. In *International Conference on Learning Representations*, 2021. 5
- [44] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34:20596–20607, 2021. 2
- [45] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 2, 5
- [46] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018. 4
- [47] 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. 1, 2, 3, 4, 5, 6, 7
- [48] Ahmad Sajedi, Samir Khaki, Konstantinos N. Plataniotis, and Mahdi S. Hosseini. End-to-end supervised multilabel contrastive learning, 2023.
- [49] Ahmad Sajedi, Samir Khaki, Yuri A Lawryshyn, and Konstantinos N Plataniotis. Probmcl: Simple probabilistic contrastive learning for multi-label visual classification. *arXiv preprint arXiv:2401.01448*, 2024. 1
- [50] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. 2, 5, 6
- [51] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5
- [52] Samarth Sinha, Han Zhang, Anirudh Goyal, Yoshua Bengio, Hugo Larochelle, and Augustus Odena. Small-gan: Speeding up gan training using core-sets. In *International Conference on Machine Learning*, pages 9005–9015. PMLR, 2020. 2
- [53] 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. 2
- [54] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2018. 2, 6
- [55] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019. 2, 5
- [56] 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. 3, 4, 5, 6, 7
- [57] Kai Wang, Jianyang Gu, Daquan Zhou, Zheng Zhu, Wei Jiang, and Yang You. Dim: Distilling dataset into generative model. *arXiv preprint arXiv:2303.04707*, 2023. 8
- [58] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. 1, 2, 5, 6
- [59] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 3
- [60] Yi Ru Wang, Samir Khaki, Weihang Zheng, Mahdi S. Hosseini, and Konstantinos N. Plataniotis. Conetv2: Efficient auto-channel size optimization for cnns. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 998–1003, 2021. 1
- [61] Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1121–1128, 2009. 2, 6
- [62] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In

*Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 3

- [63] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4820–4828, 2016. 1
- [64] Yuanhao Xiong, Ruochen Wang, Minhao Cheng, Felix Yu, and Cho-Jui Hsieh. Feddm: Iterative distribution matching for communication-efficient federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16323–16332, 2023. 2
- [65] Enneng Yang, Li Shen, Zhenyi Wang, Tongliang Liu, and Guibing Guo. An efficient dataset condensation plugin and its application to continual learning. *Advances in Neural Information Processing Systems*, 36, 2024. 2
- [66] Xiyu Yu, Tongliang Liu, Xinchao Wang, and Dacheng Tao. On compressing deep models by low rank and sparse decomposition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7370–7379, 2017. 1
- [67] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 3, 4
- [68] Hansong Zhang, Shikun Li, Pengju Wang, and Shiming Zeng, Dan Ge. M3D: Dataset condensation by minimizing maximum mean discrepancy. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2024. 3, 4
- [69] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021. 2, 3, 5, 6, 7
- [70] 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. 1, 2, 3, 4, 5, 6, 7
- [71] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021. 1, 2, 3, 5, 6, 7
- [72] Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7856–7865, 2023. 2, 3, 4