Dataset Condensation via Efficient Synthetic-Data Parameterization

Jang-Hyun Kim[1] **Jinuk Kim**[1] **Seong Joon Oh**[2 3] **Sangdoo Yun**[2] **Hwanjun Song**[2] **Joonhyun Jeong**[4]
Jung-Woo Ha[2] **Hyun Oh Song**[1]

Abstract

The great success of machine learning with massive amounts of data comes at a price of huge computation costs and storage for training and tuning. Recent studies on dataset condensation attempt to reduce the dependence on such massive data by synthesizing a compact training dataset. However, the existing approaches have fundamental limitations in optimization due to the limited representability of synthetic datasets without considering any data regularity characteristics. To this end, we propose a novel condensation framework that generates multiple synthetic data with a limited storage budget via efficient parameterization considering data regularity. We further analyze the shortcomings of the existing gradient matching-based condensation methods and develop an effective optimization technique for improving the condensation of training data information. We propose a unified algorithm that drastically improves the quality of condensed data against the current state-of-the-art on CIFAR-10, ImageNet, and Speech Commands.

1. Introduction

Deep learning has achieved great success in various fields thanks to the recent advances in technology and the availability of massive real-world data (LeCun et al., 2015). However, this success with massive data comes at a price: huge computational and environmental costs for large-scale neural network training, hyperparameter tuning, and architecture search (Patterson et al., 2021; Brown et al., 2020; Cubuk et al., 2019; Zoph & Le, 2017).

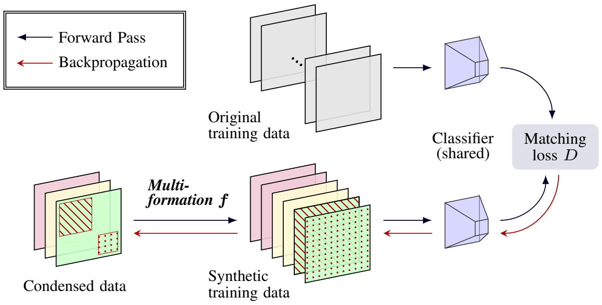


Figure 1. Illustration of the proposed dataset condensation framework with multi-formation. Under the fixed-size storage for condensed data, multi-formation synthesizes multiple data used to train models. We optimize the condensed data in an end-to-end fashion by using the differentiable multi-formation functions.

An approach to reduce the costs is to construct a compact dataset that contains sufficient information from the original dataset to train models. A classic approach to construct such a dataset is to select the *coreset* (Phillips, 2016). However, selection-based approaches have limitations in that they depend on heuristics and assume the existence of representative samples in the original data (Zhao & Bilen, 2021b). To overcome these limitations, recent studies, called *dataset condensation* or *dataset distillation*, propose to synthesize a compact dataset that has better storage efficiency than the coresets (Wang et al., 2018). The synthesized datasets have a variety of applications such as increasing the efficiency of replay exemplars in continual learning and accelerating neural architecture search (Zhao et al., 2021).

The natural data satisfy regularity conditions that form a low-rank data subspace (Huang & Mumford, 1999), *e.g.*, spatially nearby pixels in a natural image look similar and temporally adjacent signals have similar spectra in speech (Zhang et al., 2017). However, the existing condensation approaches directly optimize each data element, *e.g.,* pixel by pixel, without imposing any regularity conditions on the synthetic data (Nguyen et al., 2021; Zhao & Bilen, 2021a). Under the limited storage budget, this inefficient parameterization of synthetic datasets results in the synthesis of a limited number of data, having fundamental limitations on optimization. Furthermore, optimizing the synthetic data that have comparable training performance to the original data is challenging because it requires unrolling the entire training procedure. Recent studies propose surrogate ob-

[1]Department of Computer Science and Engineering, Seoul National University [2]NAVER AI Lab [3]University of Tübingen [4]Image Vision, NAVER Clova. Correspondence to: Hyun Oh Song <hyunoh@snu.ac.kr>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

jectives to address the challenge above, however, there are remaining questions on why certain objectives are better proxies for the true objective (Zhao & Bilen, 2021a;b).

In this work, we pay attention to making better use of condensed data elements and propose a novel optimization framework resolving the previous limitations. Specifically, we introduce a *multi-formation* process that creates multiple synthetic data under the same storage constraints as existing approaches (Figure 1). Our proposed process naturally imposes regularity on synthetic data while increasing the number of synthetic data, resulting in an enlarged and regularized dataset. In Section 3.3, we theoretically analyze the multi-formation framework and examine the conditions where the improvement is guaranteed. We further analyze the optimization challenges in the gradient matching method by Zhao & Bilen (2021b) in Section 4. Their approach induces imbalanced network gradient norms between synthetic and real data, which is problematic during optimization. Based on our analysis and empirical findings, we develop improved optimization techniques utilizing networks trained on the real data with stronger regularization and effectively mitigate the mentioned problems.

In this regard, we present an end-to-end optimization algorithm that creates information-intensive condensed data significantly outperforming all existing condensation methods. Given fixed storage and computation budgets, neural networks trained on our synthetic data show performance improvements of 10∼20%p compared to state-of-the-art methods in experimental settings with various datasets and domains including ImageNet and Speech Commands (Warden, 2018). We further verify the utility of our condensed data through experiments on continual learning, demonstrating significant performance improvements compared to existing condensation and coreset methods. We release the source code at https://github.com/snu-mllab/Efficient-Dataset-Condensation.

2. Preliminary

Given the storage budget, the goal of data condensation is to build a surrogate dataset $\mathcal{S}$ of the original training dataset $\mathcal{T}$ such that an arbitrary model trained on $\mathcal{S}$ is similar to the one trained on $\mathcal{T}$ (Wang et al., 2018). Oftentimes, the measure of similarity is in terms of the model performance on the test set because that leads to meaningful applications such as continual learning and neural architecture search (Zhao et al., 2021). Instead of solving this ultimate objective, previous methods have proposed different surrogates. For example, Wang et al. (2018) propose to optimize $\mathcal{S}$ such that a model trained on $\mathcal{S}$ minimizes the loss values over $\mathcal{T}$. However, this approach involves a nested optimization with unrolling multiple training iterations, requiring expensive computation costs.

Rather than direct optimization of model performance, Zhao et al. (2021) propose a simpler optimization framework that matches the network gradients on $\mathcal{S}$ to the gradients on $\mathcal{T}$. Let us assume a data point is m-dimensional and $\mathcal{S} \in \mathbb{R}^{n\times m}$, where n is the number of data points in $\mathcal{S}$. Zhao et al. (2021) optimize the synthetic data as

$$\begin{aligned}&\underset{\mathcal{S}\in\mathbb{R}^{n\times m}}{\text{maximize}} \sum_{t=0}^{\tau} \mathrm{Cos}\left(\nabla_\theta \ell(\theta_t;\mathcal{S}), \nabla_\theta \ell(\theta_t;\mathcal{T})\right) \qquad (1)\\ &\text{subject to } \theta_{t+1} = \theta_t - \eta\nabla_\theta \ell(\theta_t;\mathcal{S}) \text{ for } t = 0,\dots,\tau-1,\end{aligned}$$

where θ_t denotes the network weights at t^{th} training step from the randomly initialized weights θ_0 given $\mathcal{S}$, $\ell(\theta;\mathcal{S})$ denotes the training loss for weight θ and the dataset $\mathcal{S}$. $\mathrm{Cos}(\cdot,\cdot)$ denotes the channel-wise cosine similarity. Zhao et al. (2021) have reported that the class-wise gradient matching objective is effective for dataset condensation. They propose an alternating optimization algorithm with the following update rules for each class c:

$$\begin{aligned}&S_c \leftarrow S_c + \lambda\nabla_{S_c}\mathrm{Cos}\left(\nabla_\theta \ell(\theta;S_c), \nabla_\theta \ell(\theta;T_c)\right)\\ &\theta \leftarrow \theta - \eta\nabla_\theta \ell(\theta;\mathcal{S}),\end{aligned}$$

where S_c and T_c denote the mini-batches from the datasets $\mathcal{S}$ and $\mathcal{T}$, respectively. Under the formulation, Zhao & Bilen (2021b) propose to utilize differentiable siamese augmentation (DSA) for a better optimization of the synthetic data. DSA performs gradient matching on augmented data where the objective becomes $\mathbb{E}_{\omega\sim\mathcal{W}}\left[\mathrm{Cos}\left(\nabla_\theta \ell(\theta;a_\omega(\mathcal{S})), \nabla_\theta \ell(\theta;a_\omega(\mathcal{T}))\right)\right]$. Here, a_ω means a parameterized augmentation function and $\mathcal{W}$ denotes an augmentation parameter space. Subsequently, Zhao & Bilen (2021a) propose to match the hidden features rather than the gradients for fast optimization. However, the feature matching approach has some performance degradation compared to gradient matching (Zhao & Bilen, 2021a). Although this series of works have made great contributions, there are remaining challenges and questions on their surrogate optimization problems. In this work, we try to resolve the challenges by providing a new optimization framework with theoretical analysis and empirical findings.

3. Multi-Formation Framework

In this section, we pay attention to the synthetic-data parameterization in optimization and present a novel data formation framework that makes better use of condensed data. We first provide our motivating observations and introduce a multi-formation framework with theoretical analysis.

3.1. Observation

We first provide our empirical observations on the effects of the number and resolution of the synthetic data in the matching problem. The existing condensation approaches aim to synthesize a predetermined number of data about

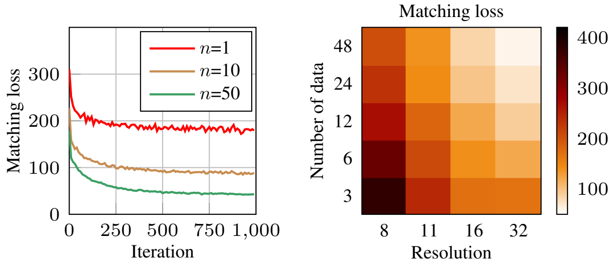


Figure 2. (Left) Matching loss curves over an increasing number of synthetic data per class (n). (Right) Matching loss heat map over various resolutions and numbers of data per class. The x-axis refers to the downsampled image resolution. We measure values on the same network after resizing data to the original size (CIFAR-10).

10 to 50 per class (Zhao & Bilen, 2021b; Nguyen et al., 2021). The left subfigure in Figure 2 shows the condensation matching loss curves of DSA over various numbers of synthetic data per class. As shown in the figure, more synthetic data lead to a smaller matching loss, indicating the importance of the number of synthetic data in the matching problem. For a comparison under the same data storage budget, we measure the matching loss on the same network after reducing the resolution of the optimized synthetic data and resizing the data to the original size. In the right subfigure in Figure 2, we find the resolution produces a moderate change in matching loss as the number of data does, even if we do not take the resolution modification into account during the condensation stage. For example, points at (16, 48) and (32, 12), which require an equal storage size, have similar loss values. Motivated by these results, we propose a multi-formation framework that makes better use of the condensed data and forms the increased number of synthetic data under the same storage budget.

3.2. Multi-Formation

The existing approaches directly match condensed data $\mathcal{S}$ to the original training data $\mathcal{T}$ and use $\mathcal{S}$ as the synthetic training data. Instead, we add an intermediate process that creates an increased number of synthetic data from $\mathcal{S}$ by mapping a data element in $\mathcal{S}$ to multiple data elements in the synthetic data (Figure 1). The previous work by Zhao & Bilen (2021b) reports that the use of random augmentations in matching problems degrades performance due to the misalignment problem. They argue the importance of the deterministic design of the matching problem. In this regard, we propose to use a deterministic process rather than a random process.

Consistent to existing approaches, we optimize and store condensed data $\mathcal{S} \in \mathbb{R}^{n\times m}$. For $n' > n$, we propose a multi-formation function $f : \mathbb{R}^{n\times m} \to \mathbb{R}^{n'\times m}$ that augments the number of condensed data $\mathcal{S}$ and creates multiple synthetic training data $f(\mathcal{S})$ in a deterministic fashion. For any matching objective D (lower the better) and target task objective ℓ, the optimization and evaluation stages of condensed data $\mathcal{S}$ with multi-formation function f are

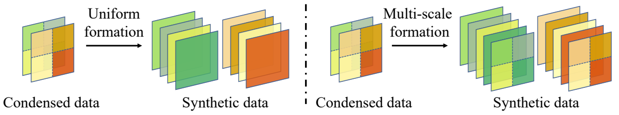


Figure 3. Illustration of the proposed multi-formation functions, in the case of multi-formation by a factor of 2.

$$\mathcal{S}^* = \operatorname*{argmin}_{\mathcal{S}\in\mathbb{R}^{n\times m}} D(f(\mathcal{S}), \mathcal{T}) \qquad \text{(Optimization)}$$

$$\theta^* = \operatorname*{argmin}_{\theta} \ell(\theta; f(\mathcal{S}^*)). \qquad \text{(Evaluation)}$$

That is, we perform matching on $\mathcal{T}$ using $f(\mathcal{S})$ and use them for evaluation. This enables us to optimize the synthetic dataset with an increased number of data, using the same storage budget. Figure 1 illustrates the optimization process with multi-formation. Note, we can use conventional data augmentations following the multi-formation.

Given a differentiable multi-formation function and matching objective, we optimize $\mathcal{S}$ in an end-to-end fashion by gradient descent. In this work, we design a simple differentiable multi-formation function and evaluate the effectiveness of our approach. The idea is to locally interpolate data elements while preserving the locality of natural data, *i.e.*, spatially nearby pixels in a natural image look similar and temporally adjacent signals have similar spectra in speech (Huang & Mumford, 1999; Zhang et al., 2017). Specifically, we partition each data and resize the partitioned data to the original size by using bilinear upsampling (Figure 3). Note, this formation function has negligible computation overhead. Furthermore, the formation function creates locally smooth synthetic data that might naturally regularize the optimization from numerous local minima. We use a fixed uniform partition function in our main experiments in Section 5 and further analyze multi-scale and learnable formation functions in Appendix D.

3.3. Theoretical Analysis

In this section, we aim to theoretically analyze our multi-formation framework. Here, we assume a data point is m-dimensional. The natural data have regularity that makes difference from random noise (Huang & Mumford, 1999). We assume that data satisfying this regularity form a subspace $\mathcal{N} \subset \mathbb{R}^m$. That is, the original training dataset $\mathcal{T} = \{t_i\}_{i=1}^{n_t}$ satisfies $t_i \in \mathcal{N}$ for $i = 1, \ldots, n_t$. With abuse of notation, we denote the space of datasets with n data points as $\mathbb{R}^{n\times m} = \{\{d_i\}_{i=1}^n \mid d_i \in \mathbb{R}^m \text{ for } i = 1, \ldots, n\}$. We further define the space of all datasets $\mathcal{D} = \cup_{n\in\mathbb{N}}\mathbb{R}^{n\times m}$ and the synthetic-dataset space of a multi-formation function $f : \mathbb{R}^{n\times m} \to \mathbb{R}^{n'\times m}$, $\mathcal{M}_f = \{f(\mathcal{S}) \mid \mathcal{S} \in \mathbb{R}^{n\times m}\}$. We now introduce our definition of distance measure between datasets. We say data d is closer to dataset $X = \{d_i\}_{i=1}^k$ than d', if $\forall i \in [1,\dots,k]$, $\|d - d_i\| \le \|d' - d_i\|$.

Definition 1. *A function $D : \mathcal{D} \times \mathcal{D} \to [0,\infty)$ is a dataset distance measure, if it satisfies the followings: $\forall X, X' \in \mathcal{D}$ where $X = \{d_i\}_{i=1}^k$, $\forall i \in [1,\dots,k]$,*

1. *$D(X,X) = 0$ and $D(X,X') = D(X',X)$.*
2. *$\forall d \in \mathbb{R}^m$ s.t. d is closer to X' than d_i, $D(X \setminus \{d_i\} \cup \{d\}, X') \le D(X,X')$.*
3. *$D(X, X' \cup \{d_i\}) \le D(X,X')$.*

The definition above states reasonable conditions for dataset distance measurement. Specifically, the second condition states that the distance decreases if a data point in a dataset moves closer to the other dataset. The third condition states that the distance decreases if a data point in a dataset is added to the other dataset. Based on the definition, we introduce the following proposition. We provide the proof in Appendix A.1.

Proposition 1. *If $\mathcal{N}^{n'} \subseteq \mathcal{M}_f$, then for any dataset distance measure D,*

$$\min_{\mathcal{S}\in\mathbb{R}^{n\times m}} D(f(\mathcal{S}),\mathcal{T}) \le \min_{\mathcal{S}\in\mathbb{R}^{n\times m}} D(\mathcal{S},\mathcal{T}).$$

Proposition 1 states that our multi-formation framework achieves the better optimum, *i.e.*, the synthetic dataset that is closer to the original dataset under any dataset distance measure. Note, the assumption $\mathcal{N}^{n'} \subseteq \mathcal{M}_f$ means that the synthetic-dataset space by f is sufficiently large to contain all data points in $\mathcal{N}$. In Appendix A.2, we provide theoretical results under a more relaxed assumption.

4. Improved Optimization Techniques

In this section, we develop optimization techniques for dataset condensation. We first analyze gradient matching (Zhao & Bilen, 2021b) and seek to provide an interpretation of why gradient matching on condensation works better than feature matching (Zhao & Bilen, 2021a). We then examine some of the shortcomings of existing gradient matching methods and propose improved techniques.

4.1. Interpretation

Convolutional or fully-connected layers in neural networks linearly operate on hidden features. From the linearity, it is possible to represent network gradients as features as in Proposition 2. For simplicity, we consider one-dimensional convolution on hidden features and drop channel notations.

Proposition 2. *Let $w_t \in \mathbb{R}^K$ and $h_t \in \mathbb{R}^W$ each denote the convolution weights and hidden features at the t^{th} layer given the input data x. Then, for a loss function ℓ, $\frac{d\ell(x)}{dw_t} = \sum_i a_{t,i} h_{t,i}$, where $h_{t,i} \in \mathbb{R}^K$ denotes the i^{th} convolution patch of h_t and $a_{t,i} = \frac{d\ell(x)}{dw_t^\intercal h_{t,i}} \in \mathbb{R}$.*

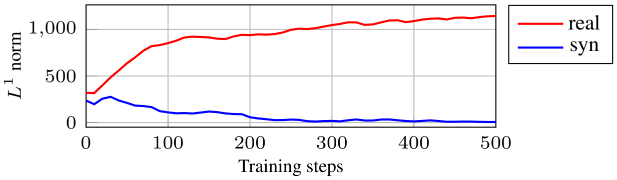


Figure 4. Evolution of L^1 norm of the network gradients given real or synthetic data. The x-axis represents the number of training steps of the networks. Here, both networks are trained on the synthetic data with augmentations. We measure the values on CIFAR-10 with ConvNet-3 used in DSA.

Proposition 2 states the gradients with respect to convolution weights can be regarded as the weighted sum of local features $h_{t,i}$. Note, the weight $a_{t,i}$ means the loss function sensitivity of the i^{th} output at the t^{th} layer, and we can interpret the network gradients as the saliency-weighted average local features. In this respect, we can view gradient matching as saliency-weighted average local feature matching.

Intuitively, saliency-weighting selectively extracts information corresponding to target labels. In addition, by matching averaged local features, we globally compare features regardless of location, which might be beneficial for datasets where target objects are non-aligned, *e.g.*, ImageNet (Deng et al., 2009). We conjecture these properties explain why gradient matching performs better than feature matching. In the following, we propose an improved gradient matching method by examining the shortcomings of existing gradient matching approaches.

4.2. Problems and Solutions

The existing gradient matching approach by DSA uses network weights θ_t trained on a condensed dataset $\mathcal{S}$ (see Equation (1)). However, this approach has some drawbacks: 1) In the optimization process, $\mathcal{S}$ and θ_t are strongly coupled, resulting in a chicken-egg problem that generally requires elaborate optimization techniques and initialization (McLachlan & Krishnan, 2007). 2) Due to the small size of $\mathcal{S}$ ($\sim 1\%$ of the original training set), overfitting occurs in the early stage of the training and the network gradients vanish quickly. Figure 4 shows that the gradient norm on $\mathcal{S}$ vanishes whereas the gradient norm on the real data $\mathcal{T}$ increases when the network is trained on $\mathcal{S}$. This leads to undesirable matching between two data sources, resulting in degraded performance when using distance-based matching objectives, such as mean squared error (Zhao et al., 2021).

To overcome these issues, we propose to utilize networks trained on $\mathcal{T}$ instead. By doing so, we optimize $\mathcal{S}$ with networks that are no longer dependent on $\mathcal{S}$, resulting in a decoupled optimization problem:

$$\underset{\mathcal{S}\in\mathbb{R}^{n\times m}}{\text{minimize}}\ \bar{D}\left(\nabla_\theta \ell(\theta^{\mathcal{T}}; f(\mathcal{S})), \nabla_\theta \ell(\theta^{\mathcal{T}}; \mathcal{T})\right).$$

Here, $\theta^{\mathcal{T}}$ represents network weights trained on $\mathcal{T}$ and $\bar{D}$ de-

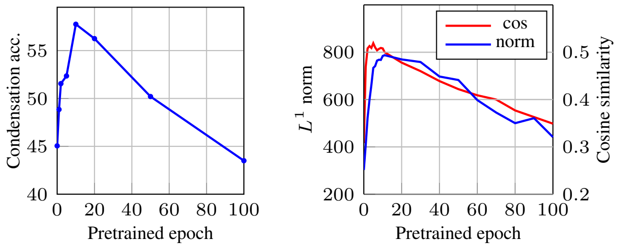


Figure 5. (Left) Condensation performance from fixed pretrained networks. The x-axis represents the number of epochs a network is trained on. (Right) Gradient analysis of the pretrained networks. The left axis measures the L^1 norm of the network gradients given a batch of data consisting of the same class. The right axis measures the average pairwise cosine-similarity between the gradients on a single data of the same class. The values are measured on ImageNet with 10 subclasses.

notes a distance-based matching objective. In addition, the large size of $\mathcal{T}$ alleviates the gradient vanishing from overfitting (Bishop, 2006). To further enhance the effect, we utilize stronger regularization for training networks. In detail, rather than a single random augmentation strategy adopted in DSA, we propose to use a sequence of augmentations and CutMix (Yun et al., 2019). Note, the mixup techniques such as CutMix effectively resolve the neural networks' over-confidence issue by using soft labels for training (Kim et al., 2020; 2021). To sum up, the proposed utilization of real data and stronger augmentations effectively resolve the gradient vanishing problem and enable the use of distance-based objective functions, resulting in the better distillation of learning information onto the synthetic data.

4.3. Algorithm

We further analyze the effect of network weights $\theta^{\mathcal{T}}$ on condensation. In detail, we examine when networks show the best condensation performance during the learning process on $\mathcal{T}$. Here, the performance means the test accuracy of neural networks trained on the condensed data. The left subfigure in Figure 5 shows the performance of condensed data optimized by a network trained for a specific epoch. We observe the best condensation performance by the networks in the early phase of training near 10 epochs.

To clarify the observation, we measure the networks' gradient norm given an intra-class mini-batch (right subfigure in Figure 5). As a result, we find that the gradient norm increases in the early phase of training and then decreases during the further training epochs. We also observe a similar pattern when we measure pairwise cosine-similarity between the gradients given a single data of the same class. These results indicate the gradient directions among intra-class data coincide at the early phase of training but diverge as the training progresses. This phenomenon is similarly observed by Jastrzebski et al. (2019); the first eigenvalue of the networks' hessian matrix increases in the early phase and decreases after a few epochs. Based on the observation, we argue that intra-class network gradients in the early training phase have more useful information to distill, and propose to utilize networks in the early training phase for condensation. Additionally, using the early phase neural networks has advantages in terms of the training cost.

Algorithm 1 Information-Intensive Dataset Condensation

```
Input: Training data $\mathcal{T}$
Notation: Multi-formation function $f$, parameterized augmentation function $a_\omega$, mixup function $h$, loss function $l$, number of classes $N_c$
Definition: $D(B, B'; \theta) = \|\nabla_\theta \ell(\theta; B)) - \nabla_\theta \ell(\theta; B')\|$
Initialize condensed dataset $\mathcal{S}$
repeat
  Initialize or load pretrained network $\theta_1$
  for $i = 1$ to $M$ do
    for $c = 1$ to $N_c$ do
      Sample an intra-class mini-batch $T_c \sim \mathcal{T}, S_c \sim \mathcal{S}$
      Update $S_c \leftarrow S_c - \lambda \nabla_{S_c} D(a_\omega(f(S_c)), a_\omega(T_c); \theta_i)$
    end for
    Sample a mini-batch $T \sim \mathcal{T}$
    Update $\theta_{i+1} \leftarrow \theta_i - \eta \nabla_\theta \ell(\theta_i; h(a_{\omega'}(T)))$
  end for
until convergence
Output: $\mathcal{S}$
```

We empirically observe that using multiple network weights for condensation rather than the fixed network weights improves the generalization of the condensed data over various test models. Therefore, we alternately update $\mathcal{S}$ and $\theta^{\mathcal{T}}$ during the optimization process. In detail, we first initialize $\theta^{\mathcal{T}}$ by random initialization or loading pretrained weights trained only for a few epochs, and then we alternatively update $\mathcal{S}$ and $\theta^{\mathcal{T}}$. In addition, we periodically reinitialize $\theta^{\mathcal{T}}$ to maintain the network to be in the early training phase. Putting together with our multi-formation framework, we propose a unified algorithm optimizing information-intensive condensed data that compactly contain the original training data information. We name the algorithm as *Information-intensive Dataset Condensation* (IDC) and describe the algorithm in Algorithm 1. Note, we adopt the siamese augmentation strategy by DSA.

5. Experimental Results

In this section, we evaluate the performance of our condensation algorithm over various datasets and tasks. We first evaluate our condensed data from CIFAR-10, ImageNet-subset, and Speech Commands by training neural networks from scratch on the condensed data (Krizhevsky et al., 2009; Deng et al., 2009; Warden, 2018). Next, we investigate the proposed algorithm by performing ablation analysis and controlled experiments. Finally, we validate the efficacy of our

Table 1. Top-1 test accuracy of test models trained on condensed datasets from CIFAR-10. We optimize the condensed data using ConvNet-3 and evaluate the data on three types of networks. Pixel/Class means the number of pixels per class of the condensed data and we denote the compression ratio to the original dataset in the parenthesis. We evaluate each case with 3 repetitions and denote the standard deviations in the parenthesis. † denotes the reported results from the original papers.

Pixel/Class	Test Model	Random	Herding	DSA	KIP	DM	IDC-I	IDC	Full dataset
10×32×32 (0.2%)	ConvNet-3	37.2	41.7	$52.1^{\dagger}$	$49.2^{\dagger}$	53.8	58.3 (0.3)	**67.5** (0.5)	88.1
	ResNet-10	34.1	35.9	32.9	-	42.3	50.2 (0.4)	**63.5** (0.1)	92.7
	DenseNet-121	36.5	36.7	34.5	-	39.0	49.5 (0.6)	**61.6** (0.6)	94.2
50×32×32 (1%)	ConvNet-3	56.5	59.8	$60.6^{\dagger}$	$56.7^{\dagger}$	65.6	69.5 (0.3)	**74.5** (0.1)	88.1
	ResNet-10	51.2	56.5	49.7	-	58.6	65.7 (0.7)	**72.4** (0.5)	92.7
	DenseNet-121	55.8	59.0	49.1	-	57.4	63.1 (0.2)	**71.8** (0.6)	94.2

condensed data on continual learning settings as a practical application (Parisi et al., 2019). We use multi-formation by a factor of 2 in our main experiments except for ImageNet where use a factor of 3. The other implementation details and hyperparameter settings of our algorithm are described in Appendix C.1. We also provide experimental results on SVHN, MNIST, and FashionMNIST in Appendix E.1.

5.1. Condensed Dataset Evaluation

A common evaluation method for condensed data is to measure the test accuracy of the neural networks trained on the condensed data (Zhao & Bilen, 2021b). It is widely known that test accuracy is affected by the type of test models as well as the quality of the data (Zoph & Le, 2017). However, some previous works overlook the contribution from test model types and compare algorithms on different test models (Nguyen et al., 2021). In this work, we emphasize specifying the test model and comparing the condensation performance on an identical test model for fair comparison. This procedure isolates the effect of the condensed data, thus enabling us to purely measure the condensation quality. We further evaluate the condensed data on multiple test models to measure the generalization ability of the condensed data across different architectures.

Baselines we consider are a random selection, Herding coreset selection (Welling, 2009), and the previous state-of-the-art condensation methods; DSA, KIP, and DM (Zhao & Bilen, 2021b; Nguyen et al., 2021; Zhao & Bilen, 2021a). We downloaded the publicly available condensed data, and otherwise, we re-implement the algorithms following the released author codes. For the implementation details of the baselines, please refer to Appendix C.2. We denote our condensed data with multi-formation as IDC and without multi-formation as IDC-I which can also be regarded as a method with the identity formation function. Finally, it is worth noting that KIP considers test models with ZCA pre-processing (Nguyen et al., 2021). However, we believe test models with standard normalization pre-processing are much more common to be used in classification and continual learning settings (Cubuk et al., 2019; Dosovitskiy et al., 2021; Rebuffi et al., 2017). In this section, we focus on test models with standard normalization pre-processing. For experimental results with ZCA, please refer to Appendix E.6.

Table 2. Top-1 test accuracy of test models with the fixed training steps. Each row matches the same dataset storage size and evaluation cost. *CN* denotes ConvNet-3, *RN* denotes ResNet-10, and *DN* denotes DenseNet-121. We measure training times on an RTX-3090 GPU.

Pixel Ratio	Test Model	DSA	KIP	DM	IDC-I	IDC	Evaluation Time
0.2%	CN	52.1	49.1	53.8	58.3	**65.3**	10s
	RN	32.9	40.8	42.3	50.2	**57.7**	20s
	DN	34.5	42.1	39.0	49.5	**60.6**	100s
1%	CN	60.6	57.9	65.6	69.5	**73.6**	50s
	RN	49.7	52.9	58.6	65.7	**72.3**	90s
	DN	49.1	54.4	57.4	63.1	**71.6**	400s

CIFAR-10. The CIFAR-10 training set consists of 5,000 images per class each with 32 × 32 pixels. Following the condensation baselines, we condense the training set with the storage budgets of 10 and 50 images per class by using 3-layer convolutional networks (ConvNet-3). For network architecture effect on condensation, please refer to Appendix E.4. We evaluate the condensed data on multiple test models: ConvNet-3, ResNet-10, and DenseNet-121 (He et al., 2016; Huang et al., 2017). It is worth noting that Zhao & Bilen (2021b) used data augmentation when evaluating DSA but did not apply any data augmentation when evaluating simple baselines Random and Herding. This is not a fully fair way to compare the quality of data. In our paper, we re-evaluate all baselines including DSA by using the same augmentation strategy as ours and report the best performance for fair comparison. For the more detailed results on augmentation, please refer to Appendix E.2.

Table 1 summarizes the test accuracy of neural networks trained on each condensed data. From the table, we confirm that both IDC and IDC-I significantly outperform all the baselines. Specifically, IDC outperforms the best baseline by over 10%p across all the test models and compression ratios. However, IDC requires additional training steps to

Table 3. Top-1 test accuracy of test models trained on condensed datasets from ImageNet-subset. We optimize the condensed data using ResNetAP-10 and evaluate the data on three types of networks. We evaluate the condensed data by using the identical training strategy.

Class	Pixel/Class	Test Model	Random	Herding	DSA	DM	IDC-I	IDC	Full Dataset
10	10×224×224 (0.8%)	ResNetAP-10	46.9	50.4	52.7	52.3	61.4 (0.8)	**72.8** (0.6)	90.8
		ResNet-18	43.3	47.0	44.1	41.7	56.2 (1.2)	**73.6** (0.4)	93.6
		EfficientNet-B0	46.3	50.2	48.3	45.0	58.7 (1.4)	**74.7** (0.5)	95.9
10	20×224×224 (1.6%)	ResNetAP-10	51.8	57.5	57.4	59.3	65.5 (1.0)	**76.6** (0.4)	90.8
		ResNet-18	54.3	57.9	56.9	53.7	66.0 (0.7)	**75.7** (1.0)	93.6
		EfficientNet-B0	60.3	59.0	62.5	57.7	66.3 (0.5)	**78.1** (1.0)	95.9
100	10×224×224 (0.8%)	ResNetAP-10	20.7	22.6	21.8	22.3	29.2 (0.4)	**46.7** (0.2)	82.0
		ResNet-18	15.7	15.9	13.5	15.8	23.3 (0.3)	**40.1** (0.5)	84.6
		EfficientNet-B0	22.4	24.5	19.9	20.7	27.7 (0.6)	**36.3** (0.6)	85.6
100	20×224×224 (1.6%)	ResNetAP-10	29.7	31.1	30.7	30.4	34.5 (0.1)	**53.7** (0.9)	82.0
		ResNet-18	24.3	23.4	20.0	23.4	29.8 (0.2)	**46.4** (1.6)	84.6
		EfficientNet-B0	33.2	35.6	30.6	31.0	33.2 (0.5)	**49.6** (1.2)	85.6

converge due to the formation process in general. Considering applications where training cost matters, such as architecture search, we compare methods under the fixed training steps and report the results in Table 2. That is, we reduce the training epochs when evaluating IDC, and match the number of gradient descent steps identical to the other baselines. In the case of KIP, which originally uses a neural tangent kernel for training networks, we re-evaluate the dataset by using stochastic gradient descent as others to match the computation costs. Table 2 shows IDC still consistently outperforms baselines by a large margin.

ImageNet. Existing condensation methods only perform the evaluation on small-scale datasets, such as MNIST or CIFAR-10. To the best of our knowledge, our work is the first to evaluate condensation methods on challenging high-resolution data, ImageNet (Deng et al., 2009), to set a benchmark and analyze how the condensation works on large-scale datasets. We implement condensation methods on ImageNet-subset consisting of 10 and 100 classes (Tian et al., 2020), where each class consists of approximately 1200 images. We provide a detailed dataset description in Appendix B. Note, KIP requires hundreds of GPUs for condensing CIFAR-10 and does not scale on ImageNet. In the ImageNet experiment, we use ResNetAP-10 for condensation, which is a modified ResNet-10 by replacing strided convolution as average pooling for downsampling (Zhao & Bilen, 2021a). For test models, we consider ResNetAP-10, ResNet-18, and EfficientNet-B0 (Tan & Le, 2019).

Table 3 summarizes the test accuracy of neural networks trained on the condensed data. The table shows IDC and IDC-I significantly outperform all the baselines across the various numbers of classes, compression ratios, and test models. One of the notable results is that the existing condensation methods do not transfer well to other test models. For example, DM performs better on ResNetAp-10 compared to Random selection but performs poorly on other test models. On contrary, IDC consistently outperforms other methods regardless of test model types. This indicates that our networks trained on large real datasets extract more task-relevant information with less architectural inductive bias than randomly initialized networks (DM) or networks trained on synthetic datasets (DSA). In Figure 6, we provide representative condensed samples from IDC-I. Note, these samples are initialized by random real training samples. We provide the qualitative comparison of our condensed data and real training data in Appendix F.

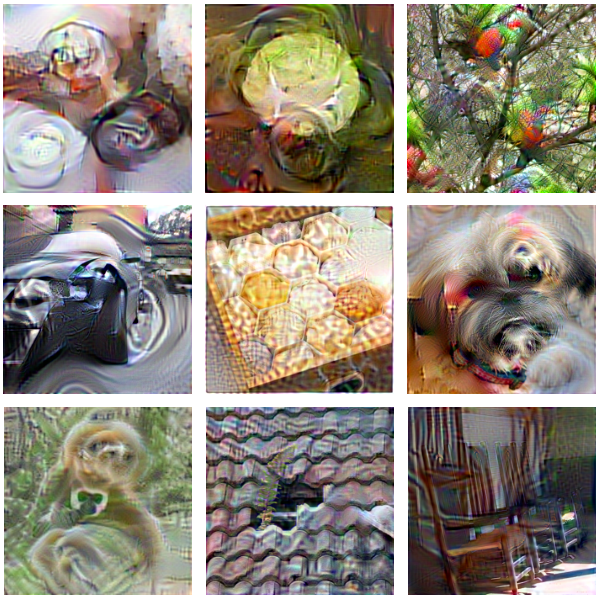


Figure 6. Representative samples from IDC-I condensed data on ImageNet-100. The corresponding class labels are as follows: bottle cap, cabbage, lorikeet, car wheel, honeycomb, Shih-Tzu, gibbon, tile roof, and rocking chair.

Speech Domain. We evaluate our algorithm on speech domain data to verify the generality of our algorithm.

Table 4. Top-1 test accuracy of ConvNet-4 trained on condensed spectrograms. *Rand.* and *Herd.* denote Random and Herding.

Spectrogram/ Class	Rand.	Herd.	DSA	DM	IDC-I	IDC	Full Dataset
10×64×64 (1%)	42.6	56.2	65.0	69.1	73.3	**82.9**	93.4
20×64×64 (2%)	57.0	72.9	74.0	77.2	83.0	**86.6**	

Table 5. Ablation study of the proposed techniques (50 images per class on CIFAR-10). *Syn* denotes condensing with networks trained on the synthetic dataset and *Real* denotes condensing with networks trained on the real dataset. *Cos* denotes cosine-similarity matching objective, *MSE* denotes mean-square-error matching objective, and *Reg.* denotes our proposed stronger regularization.

Test Model	Syn+ Cos (DSA)	Syn+ MSE	Real+ Cos	Real+ MSE	Real+Reg.+ MSE (Ours)
ConvNet-3	60.6	25.8	63.4	67.0	**69.5**
ResNet-10	49.7	25.7	59.1	61.6	**65.7**

In detail, we condense Mini Speech Commands that contains 8,000 one-second audio clips of 8 command classes (Warden, 2018). We preprocess speech data and obtain magnitude spectrograms each of size 64×64. For a detailed description of the dataset and preprocessing, please refer to Appendix B. In the case of speech data, we use a one-dimensional multi-formation function by a factor of 2 along the time-axis of a spectrogram. Table 4 shows the test accuracy on the speech dataset. IDC consistently outperforms baseline methods by large margins and achieves test performance close to the full dataset training, verifying its effectiveness on speech domain as well as on image domain.

5.2. Analysis

Ablation Study. In this section, we perform an ablation study on our gradient matching techniques described in Section 4. Specifically, we measure the isolated effect of 1) networks trained on real training data, 2) distance-based matching objective, and 3) stronger regularization on networks. Table 5 shows the ablation results of IDC-I on CIFAR-10 condensed with 50 images per class. From the table, we find that using MSE matching objective with networks trained on the synthetic dataset (*Syn+MSE*) degenerates the performance significantly. However, when we use the MSE objective with networks trained on the real training dataset, the performance significantly increases compared to the baseline (*DSA*), especially on ResNet-10. Furthermore, we find that strong regularization on networks brings additional performance improvements on both test models. The results demonstrate that the distance-based objective (*MSE*) better distills training information than the similarity-based objective (*Cos*) when using well-trained networks.

Table 6. Test performance comparison of IDC and IDC-I with post-downsampling (IDC-I-post) on CIFAR-10. We denote the number of stored pixels in parenthesis.

Test Model	IDC (50×32×32)	IDC-I-post (200×16×16)	IDC-I (200×32×32)
ConvNet-3	74.5	68.8	76.6
ResNet-10	72.4	63.1	74.9

Table 7. Condensation performance over various multi-formation factors on CIFAR-10 and ImageNet-10.

Dataset	Test	Multi-Formation Factor			
(Pixel/Class)	Model	1	2	3	4
CIFAR-10	ConvNet-3	69.5	**74.5**	68.9	62.0
(50×32×32)	ResNet-10	65.7	**72.4**	62.9	59.1
ImageNet-10	ResNetAP-10	65.5	73.3	76.6	**77.5**
(20×224×224)	ResNet-18	66.0	70.8	**75.7**	75.2

Comparison to Post-Downsampling. One of the simple ways to save storage budget is to reduce the resolution of the synthesized data. In this subsection, we compare our end-to-end optimization framework to a post-downsampling approach which reduces the resolution of the optimized synthetic data and resizes the data to the original size at evaluation. Table 6 shows IDC significantly outperforms IDC-I with post-downsampling under the same number of stored pixels, even approaching the performance of IDC-I without downsampling which stores 4 times more pixels. This result verifies the effectiveness of the end-to-end approach considering the formation function during the optimization process, *i.e.*, finding the optimal condensed data given a fixed formation function.

On Multi-Formation Factor. We further study the effect of multi-formation factor (*i.e.*, upsampling factor). Table 7 summarizes the test accuracy of condensed data with different multi-formation factors on various data scales. Note, the higher multi-formation factor results in a larger number of synthetic data but each with a lower resolution. Table 7 shows that datasets have different optimal multi-formation factors; 2 is optimal for CIFAR-10 and 3-4 are optimal for ImageNet. These results mean that there is a smaller room for trading off resolution in the case of CIFAR-10 than ImageNet where the input size is much larger.

5.3. Application: Continual Learning

Recent continual learning approaches include the process of constructing a small representative subset of data that has been seen so far and training it with newly observed data (Rebuffi et al., 2017; Bang et al., 2021). This implies that the quality of the data subset is bound to affect the continual learning performance. In this section, we utilize the condensed data as exemplars for the previously seen classes or tasks and evaluate its effectiveness under the two types of continual learning settings: class incremental and task incremental (Zhao & Bilen, 2021a;b). Due to lack of space, we describe the detailed settings and results of task incremental setting in Appendix C.3.

We follow the class incremental setting from Zhao & Bilen (2021a), where the CIFAR-100 dataset is given across 5 steps with a memory budget of 20 images per class. This setting trains a model continuously and purely on the latest data memory at each stage (Prabhu et al., 2020). We synthesize the exemplars by only using the data samples of currently available classes at each stage with ConvNet-3. We evaluate the condensed data on two types of networks, ConvNet-3 and ResNet-10, and compare our condensation methods with Herding, DSA, and DM.

Figure 7 shows that IDC-I and IDC are superior to other baselines, both in ConvNet-3 and ResNet-10. Particularly, our multi-formation approach considerably increases the performance by over 10%p on average. In addition, from the results on ResNet-10, we find that DSA and DM do not maintain their performance under the network transfer, whereas our condensation methods outperform the baselines regardless of the networks types. That is, it is possible to efficiently condense data with small networks (ConvNet-3) and use the data on deeper networks when using our methods.

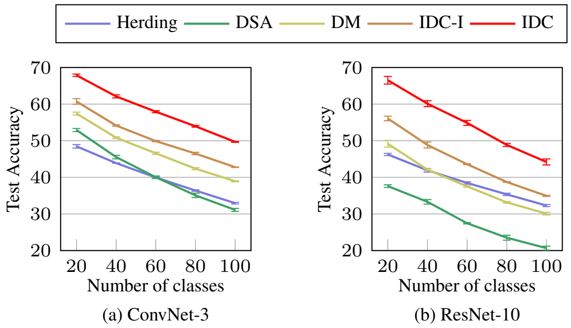


Figure 7. Top-1 test accuracy of continual learning with condensed exemplars on CIFAR-100.

6. Related Work

One of the classic approaches to establishing a compact representative subset of a huge dataset is coreset selection (Phillips, 2016; Toneva et al., 2019). Rather than selecting a subset, Maclaurin et al. (2015) originally proposed synthesizing a training dataset by optimizing the training performance. Following the work, Such et al. (2020) introduce generative modeling for the synthetic dataset. However, these works do not consider storage efficiency. The seminal work by Wang et al. (2018) studies synthesizing small training data with a limited storage budget. Building on this work, Sucholutsky & Schonlau (2021) attempt to co-optimize soft labels as well as the data, but they suffer from overfitting. Subsequently, Nguyen et al. (2021) formulate the problem as kernel ridge regression and optimize the data based on neural tangent kernel. However, this approach requires hundreds of GPUs for condensation. Zhao et al. (2021) propose a scalable algorithm by casting the original bi-level optimization as a simpler matching problem. Following the work, Zhao & Bilen (2021b) exploit siamese augmentation to improve performance, and Zhao & Bilen (2021a) suggest feature matching to accelerate optimization. Concurrently, Cazenavette et al. (2022) proposes to optimize the condensed data by matching training trajectories on the networks trained on real data.

Discussion on Dataset Structure In this work, we constrain the condensation optimization variables (*i.e.*, $\mathcal{S}$) to have the same shape as the original training data. This enables us to design an intuitive and efficient formation function that has negligible computation and storage overhead. However, if we deviate from pursuing the same shape, there exist a variety of considerable condensed data structures. For example, we can parameterize a dataset as dictionary phrase coding or neural network generator (Mairal et al., 2009; Goodfellow et al., 2014). Nonetheless, it is not trivial to tailor these approaches for efficient data condensation. That is, it may require more storage or expensive computation costs for synthesis. For example, Sitzmann et al. (2020) use multi-layer neural networks that require much more storage than a single image to completely reconstruct a single image.

7. Conclusion

In this study, we address difficulties in optimization and propose a novel framework and techniques for dataset condensation. We propose a multi-formation process that defines enlarged and regularized data space for synthetic data optimization. We further analyze the shortcomings of the existing gradient matching algorithm and provide effective solutions. Our algorithm optimizes condensed data that achieve state-of-the-art performance in various experimental settings including speech domain and continual learning.

Acknowledgement

We are grateful to Junsuk Choe for helpful discussions. This work was supported by SNU-NAVER Hyperscale AI Center, Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00882, (SW STAR LAB) Development of deployable learning intelligence via self-sustainable and trustworthy machine learning and No. 2022-0-00480, Development of Training and Inference Methods for Goal-Oriented Artificial Intelligence Agents), and Basic Science Research Program through the National Research Foundation of Korea (NRF) (2020R1A2B5B03095585). Hyun Oh Song is the corresponding author.

References

Bang, J., Kim, H., Yoo, Y., Ha, J.-W., and Choi, J. Rainbow memory: Continual learning with a memory of diverse samples. In *CVPR*, 2021.

Bishop, C. M. *Pattern recognition and machine learning*. springer, 2006.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In *NeurIPS*, 2020.

Cazenavette, G., Wang, T., Torralba, A., Efros, A. A., and Zhu, J.-Y. Dataset distillation by matching training trajectories. *arXiv preprint arXiv:2203.11932*, 2022.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *CVPR*, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Dong, C., Loy, C. C., and Tang, X. Accelerating the super-resolution convolutional neural network. In *ECCV*, pp. 391–407. Springer, 2016.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *NeurIPS*, 2014.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *CVPR*, 2017.

Huang, J. and Mumford, D. Statistics of natural images and models. In *CVPR*, 1999.

Jastrzebski, S., Kenton, Z., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. On the relation between the sharpest directions of dnn loss and the sgd step length. In *ICLR*, 2019.

Kim, J.-H., Choo, W., and Song, H. O. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning (ICML)*, 2020.

Kim, J.-H., Choo, W., Jeong, H., and Song, H. O. Co-mixup: Saliency guided joint mixup with supermodular diversity. In *ICLR*, 2021.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553), 2015.

Li, Z. and Hoiem, D. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12), 2017.

Maclaurin, D., Duvenaud, D., and Adams, R. Gradient-based hyperparameter optimization through reversible learning. In *ICML*, 2015.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. Online dictionary learning for sparse coding. In *ICML*, 2009.

McLachlan, G. J. and Krishnan, T. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

Nguyen, T., Novak, R., Xiao, L., and Lee, J. Dataset distillation with infinitely wide convolutional networks. In *NeurIPS*, 2021.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113, 2019.

Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., So, D., Texier, M., and Dean, J. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

Phillips, J. M. Coresets and sketches. *arXiv preprint arXiv:1601.00617*, 2016.

Prabhu, A., Torr, P. H., and Dokania, P. K. Gdumb: A simple approach that questions our progress in continual learning. In *ECCV*, 2020.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning. In *CVPR*, 2017.

Sitzmann, V., Martel, J. N., Bergman, A. W., Lindell, D. B., and Wetzstein, G. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020.

Such, F. P., Rawal, A., Lehman, J., Stanley, K., and Clune, J. Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data. In *ICML*, 2020.

Sucholutsky, I. and Schonlau, M. Soft-label dataset distillation and text dataset distillation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021.

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. In *ECCV*, 2020.

Toneva, M., Sordoni, A., Combes, R. T. d., Trischler, A., Bengio, Y., and Gordon, G. J. An empirical study of example forgetting during deep neural network learning. In *ICLR*, 2019.

Wang, T., Zhu, J.-Y., Torralba, A., and Efros, A. A. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

Warden, P. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

Welling, M. Herding dynamical weights to learn. In *ICML*, 2009.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

Zhang, Y., Pezeshki, M., Brakel, P., Zhang, S., Bengio, C. L. Y., and Courville, A. Towards end-to-end speech recognition with deep convolutional neural networks. *arXiv preprint arXiv:1701.02720*, 2017.

Zhao, B. and Bilen, H. Dataset condensation with distribution matching. *arXiv preprint arXiv:2110.04181*, 2021a.

Zhao, B. and Bilen, H. Dataset condensation with differentiable siamese augmentation. In *ICML*, 2021b.

Zhao, B., Mopuri, K. R., and Bilen, H. Dataset condensation with gradient matching. In *ICLR*, 2021.

Zoph, B. and Le, Q. V. Neural architecture search with reinforcement learning. In *ICLR*, 2017.

A. Theoretical Analysis

A.1. Proofs

Definition 1. *A function $D : \mathcal{D} \times \mathcal{D} \rightarrow [0, \infty)$ is a dataset distance measure, if it satisfies the followings: $\forall X, X' \in \mathcal{D}$ where $X = \{d_i\}_{i=1}^k$, $\forall i \in [1, \dots, k]$,*

1. *$D(X, X) = 0$ and $D(X, X') = D(X', X)$.*
2. *$\forall d \in \mathbb{R}^m$ s.t. d is closer to X' than d_i, $D(X \setminus \{d_i\} \cup \{d\}, X') \leq D(X, X')$.*
3. *$D(X, X' \cup \{d_i\}) \leq D(X, X')$.*

Note, we say data d is closer to dataset $X = \{d_i\}_{i=1}^k$ than d', if $\forall i \in [1, \dots, k]$, $\|d - d_i\| \leq \|d' - d_i\|$.

Proposition 1. *If $\mathcal{N}^{n'} \subseteq \mathcal{M}_f$, then for any dataset distance measure D,*

$$\min_{\mathcal{S} \in \mathbb{R}^{n \times m}} D(f(\mathcal{S}), \mathcal{T}) \leq \min_{\mathcal{S} \in \mathbb{R}^{n \times m}} D(\mathcal{S}, \mathcal{T}).$$

Proof. For simplicity, we denote $[1, \dots, n]$ as $[n]$. Let us denote $\mathcal{T} = \{t_i\}_{i=1}^{n_t}$ and $\mathcal{S} = \{s_j\}_{j=1}^{n}$, where $t_i \in \mathcal{N} \subset \mathbb{R}^m$ and $s_j \in \mathbb{R}^m$, $\forall i \in [n_t]$ and $\forall j \in [n]$. Under the assumption that $\mathcal{N}$ is a subspace of $\mathbb{R}^m$, there exists the projection of s_j onto $\mathcal{N}$, $\bar{s}_j \in \mathcal{N}$. Because $t_i \in \mathcal{N}$ for $i = 1, \dots, n_t$, $\|\bar{s}_j - t_i\| \leq \|s_j - t_i\|, \forall j \in [n]$ and $\forall i \in [n_t]$. This means the projection $\bar{s}_j$ is closer to $\mathcal{T}$ than $s_j, \forall j \in [n]$. Let us define a partially projected dataset $\bar{S}_k = \{\bar{s}_j\}_{j=1}^k \cup \{s_j\}_{j=k+1}^n$. Then by the second axiom of Definition 1,

$$D(\bar{S}_n, \mathcal{T}) \leq D(\bar{S}_{n-1}, \mathcal{T}) \leq \ldots \leq D(\mathcal{S}, \mathcal{T}).$$

This result means that the optimum $\mathcal{S}^* = \operatorname{argmin} D(\mathcal{S}, \mathcal{T})$ satisfies $\mathcal{S}^* \in \mathcal{N}^n$. Note our multi-formation augments the number of data from n to n' where $n < n'$. Let us denote $k' = n' - n$ and $S^*_{add} = \mathcal{S}^* \cup \{t_i\}_{i=1}^{k'}$. By the third axiom of Definition 1,

$$D(S^*_{add}, \mathcal{T}) \leq D(\mathcal{S}^*, \mathcal{T}).$$

The elements of S^*_{add} lie in $\mathcal{N}$ and $S^*_{add} \in \mathcal{N}^{n'}$. From the assumption $\mathcal{N}^{n'} \subseteq \mathcal{M}_f$, $\exists \mathcal{S} \in \mathbb{R}^{n \times m}$ *s.t.* $f(\mathcal{S}) = S^*_{add}$. Thus,

$$\begin{aligned}\min_{\mathcal{S} \in \mathbb{R}^{n \times m}} D(f(\mathcal{S}), \mathcal{T}) &\leq D(S^*_{add}, \mathcal{T}) \\ &\leq D(\mathcal{S}^*, \mathcal{T}) = \min_{\mathcal{S} \in \mathbb{R}^{n \times m}} D(\mathcal{S}, \mathcal{T}).\end{aligned}$$

□

Proposition 2. *Let $w_t \in \mathbb{R}^K$ and $h_t \in \mathbb{R}^W$ each denote the convolution weights and hidden features at the t^{th} layer given the input data x. Then, for a loss function ℓ, $\frac{d\ell(x)}{dw_t} = \sum_i a_{t,i} h_{t,i}$, where $h_{t,i} \in \mathbb{R}^K$ denotes the i^{th} convolution patch of h_t and $a_{t,i} = \frac{d\ell(x)}{dw_t^\intercal h_{t,i}} \in \mathbb{R}$.*

Proof. To clarify, we note that we are learning flipped kernel during training. Then the convolution output of the t^{th} layer o_t becomes $[w_t^\intercal h_{t,1}, \dots, w_t^\intercal h_{t,n_o}]$, where $n_o = W - K - 1$ denotes the number of convolution patches given the convolution stride of 1. Then from the chain rule,

$$\begin{aligned}\frac{d\ell(x)}{dw_t} &= \frac{d\ell(x)}{do_t}\frac{do_t}{dw_t} \\ &= \left[\frac{d\ell(x)}{dw_t^\intercal h_{t,1}}, \dots, \frac{d\ell(x)}{dw_t^\intercal h_{t,n_o}}\right][h_{t,1}, \dots, h_{t,n_o}]^\intercal \\ &= \sum_{i=1}^{n_0} \frac{d\ell(x)}{dw_t^\intercal h_{t,i}} h_{t,i}.\end{aligned}$$

□

A.2. Proposition 1 with Relaxed Assumption

In Proposition 1, we assume $\mathcal{N}^{n'} \subseteq \mathcal{M}_f$ that the synthetic-dataset space by f is sufficiently large to contain all data points in $\mathcal{N}$. Relaxing the assumption, we consider when $\mathcal{M}_f$ approximately covers $\mathcal{N}^{n'}$. With the following notion of ϵ-cover, we describe the trade-off between the effects from the increase in the number of data and the decrease in representability of the synthetic datasets.

Definition 2. *Given a dataset distance measure D, $\mathcal{M}_f$ is a ϵ-cover of $\mathcal{N}^{n'}$ on D if $\forall X' \in \mathcal{N}^{n'}$, $\exists S \in \mathbb{R}^{n \times m}$ s.t. $D(f(S), X') \leq \epsilon$.*

Here, we assume a dataset distance measure D satisfies the triangular inequality. From the proof above in Proposition 1, $\exists S^*_{add} \in \mathcal{N}^{n'}$ *s.t.* $D(S^*_{add}, \mathcal{T}) \leq \min_{\mathcal{S} \in \mathbb{R}^{n \times m}} D(\mathcal{S}, \mathcal{T})$. Let us denote the gain $G = \min_{\mathcal{S}} D(\mathcal{S}, \mathcal{T}) - D(S^*_{add}, \mathcal{T})$. If $\mathcal{M}_f$ is a ϵ-cover of $\mathcal{N}^{n'}$ on D, then $\exists \mathcal{S} \in \mathbb{R}^{n \times m}$ *s.t.*

$$\begin{aligned}D(f(S), \mathcal{T}) &\leq D(S^*_{add}, \mathcal{T}) + D(f(S), S^*_{add}) \\ &\leq D(S^*_{add}, \mathcal{T}) + \epsilon.\end{aligned}$$

Note, we use the triangular inequality in the first inequality above and use the definition of ϵ-cover in the second inequality. We can conclude that

$$\begin{aligned}\min_{\mathcal{S} \in \mathbb{R}^{n \times m}} D(f(\mathcal{S}), \mathcal{T}) &\leq D(S^*_{add}, \mathcal{T}) + \epsilon \\ &= \min_{\mathcal{S} \in \mathbb{R}^{n \times m}} D(\mathcal{S}, \mathcal{T}) - G + \epsilon.\end{aligned}$$

To summarize, the optimization with multi-formation function f can generate a synthetic dataset that has at least $G - \epsilon$ smaller distance to the original data $\mathcal{T}$ compared to when not using f. We can interpret G as a possible gain by the increase in the number of data, *i.e.*, from n to n', and ϵ as the representability loss in parameterization by f. This formulates a new research problem on data condensation: parameterization of a larger synthetic dataset that sufficiently satisfies the data regularity conditions.

B. Datasets

ImageNet-subset. Many recent works in machine learning evaluate their proposed algorithms using subclass samples from IamgeNet to validate the algorithms on large-scale data with a reasonable amount of computation resources (Rebuffi et al., 2017; Tian et al., 2020). In our ImageNet experiment, we borrow a subclass list containing 100 classes from Tian et al. (2020)[1]. We use the first 10 classes from the list in our ImageNet-10 experiments. We performed the experiments after preprocessing the images to a fixed size of 224×224 using resize and center crop functions.

Mini Speech Commands. This dataset contains one-second audio clips of 8 command classes, which is a subset of the original Speech Commands dataset (Warden, 2018). The dataset consists of 1,000 samples for each class. We split the dataset randomly and use 7 of 8 as training data and 1 of 8 as test data. We downloaded the Mini Speech Commands from the official TensorFlow page[2]. Following the guideline provided on the official page, we load the audio clips with 16,000 sampling rates and process the waveform data with Short-time Fourier transform to obtain the magnitude spectrogram of size 128×125. Then, we apply zero-padding and perform downsampling to reduce the input size to 64×64. Finally, we use log-scale magnitude spectrograms for experiments.

C. Implementation Details

C.1. Ours

In all of the experiments, we fix the number of inner iterations $M = 100$ (Algorithm 1). For CIFAR-10, we use data learning rate $\lambda = 0.005$, network learning rate $\eta = 0.01$, and the MSE objective. For other datasets, we use L^1 matching objective. For ImageNet-10, we use data learning rate $\lambda = 0.003$ and network learning rate $\eta = 0.01$. For ImageNet-100, we use data learning rate $\lambda = 0.001$ and network learning rate $\eta = 0.1$. For the speech dataset, we use data learning rate $\lambda = 0.003$ and network learning rate $\eta = 0.0003$. Rather than a single random augmentation strategy by DSA, we use a sequence of color transform, crop, and cutout for gradient matching. We train networks that are used for the matching with a sequence of color transform, crop, and CutMix (Yun et al., 2019). Following Zhao & Bilen (2021b), we initialize the synthetic data as random real training data samples, which makes the optimization faster compared to the random noise initialization.

We follow the evaluation setting by DSA in the case of CIFAR-10 (Zhao & Bilen, 2021b). We train neural networks on the condensed data for 1,000 epochs with a 0.01 learning rate. We use the DSA augmentation and CutMix. Note, we apply CutMix for other baselines unless it degrades the performance. In the case of ImageNet, we train networks on the condensed data by using random resize-crop and CutMix. We use 0.01 learning rate and train models until convergence: 2,000 epochs for 10 image/class and 1,500 epochs for 20 image/class. We use an identical evaluation strategy for all cases in Table 3.

C.2. Baselines

In the case of DSA (Zhao & Bilen, 2021b), we download the author-released condensed dataset and evaluate the dataset[3]. We train neural networks by following the training strategy from the official Github codes. We find that evaluation with CutMix degrades the performance of DSA, and report better results without CutMix in Table 1. For all of the other baselines, we use an identical evaluation strategy to ours.

In the case of DM (Zhao & Bilen, 2021a), the codes are not released. Following the paper, we implemented the algorithm and tuned learning rates. As a result, we obtain superior performance than the reported values in Zhao & Bilen (2021a). Specifically, the original paper reports CIFAR-10 performance on ConvNet-3 of 63.0 whereas we obtain the performance of 65.6 (50 images per class). We report our improved results of DM in Table 1.

C.3. Continual Learning

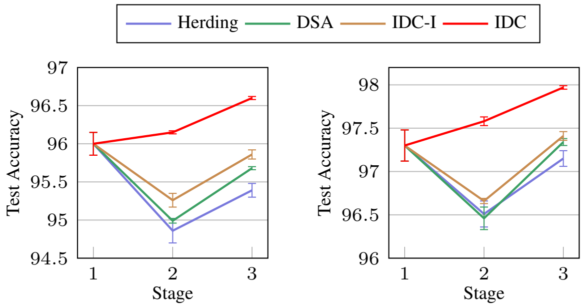
Class Incremental Setting. We reproduce the result based on the author released condensed dataset by Zhao & Bilen (2021b). Instead of training the network from scratch as in previous works (Prabhu et al., 2020; Zhao & Bilen, 2021a), we adopt distillation loss described in Li & Hoiem (2017) and train continuously by loading weights from the previous step and expanding the output dimension of the last fully-connected layer, which is a more realistic scenario in continual learning (Rebuffi et al., 2017). We train the model for 1000 epochs each stage using SGD with a learning rate of 0.01, decaying by a factor of 0.2 at epochs 600 and 800. We use 0.9 for momentum and 0.0005 for weight decay.

Task Incremental Setting. We follow the task incremental setting by Zhao & Bilen (2021b), which consists of three digit-datasets (SVHN → MNIST → USPS). At each training stage, a new set of the corresponding data is provided, whereas the previously seen datasets are prohibited except for a few exemplars. We compare our methods with Herding and DSA, excluding DM where the data under this setting is not released. As shown in Figure 8, we verify that our condensed data significantly outperform the baselines.

[1] https://github.com/HobbitLong/CMC

[2] https://www.tensorflow.org/tutorials/audio/simple_audio

[3] https://github.com/VICO-UoE/DatasetCondensation



(a) ConvNet-3

(b) ResNet-10

Figure 8. Continual learning performance with exemplars on digit datasets (SVHN-MNIST-USPS).

D. Other Multi-Formation Functions

In this section, we study other types of multi-formation functions that are worth considering under our framework.

Multi-Scale Multi-Formation Function. The synthetic data by the uniform multi-formation function do not share data elements with each other (Figure 3). Here, we design a multi-scale formation function that increases the number of synthetic data by sharing condensed data elements across multiple synthetic data (right subfigure in Figure 3). Table 8 compares the test accuracy to the default formation function on CIFAR-10. The table shows that the multi-scale approach outperforms the uniform formation function under the small storage budget where the uniform approach does not create sufficiently many synthetic data.

Learnable Multi-Formation Function. We further study the potential direction of exploiting learnable multi-formation function which can synthesize diverse representative images at the cost of additional computation overhead and storage. In this experiment, we replace the upsampling by a learnable function using Fast Super-Resolution Convolutional Neural Networks (FSRCNN) with a reduced number of parameters (Dong et al., 2016). Table 9 summarizes the condensation performance of learnable multi-formation function with different factors on CIFAR-10. While the extra learnable module does not improve performance with the formation factor of 2, it improves the performance with the factor of 3. We conjecture that the upsampling can generate sufficiently informative synthetic data in the lower factor, but suffers from the lack of representability in the higher factor. In such scenarios with the larger factor, the learnable multi-formation function shows promising results.

Table 8. Test performance comparison of the uniform and multi-scale formation functions on CIFAR-10.

Pixel/Class	Test Model	Uniform (default)	Multi-Scale
10×32×32 (0.2%)	ConvNet-3 ResNet-10	67.5 63.5	**69.2** **64.8**
50×32×32 (1.0%)	ConvNet-3 ResNet-10	**74.5** **72.4**	73.1 69.7

Table 9. Condensation performance comparison of learnable multi-formation functions to upsampling (10 images per class on CIFAR-10). *CN* denotes ConvNet-3 and *RN* denotes ResNet-10.

Test	Factor 2		Factor 3	
Model	Upsample	FSRCNN	Upsample	FSRCNN
CN	67.5	66.2	66.7	**67.9**
RN	63.5	62.0	60.6	**64.4**

E. Additional Experiments

E.1. Experimental Results on Other Datasets

We evaluate our method on SVHN, MNIST, FashionMNIST, and CIFAR-10 including 1 img/cls setting and verify that our methods consistently outperform baselines. Table 10 shows multi-formation is much more effective at low compression rates (1 img/cls) and improves performance by up to 30%p (on SVHN) compared to the best baseline. We also find that the effect of multi-formation is diminishing at (FashionMNIST, 50 img/cls) where IDC-I is the best. We conjecture that the representation loss by multi-formation at this point is greater than the gain by an increased number of data, which can be backed up by analysis in Appendix A.2.

E.2. Isolated Effect of Strong Augmentation

We conduct an ablation study investigating the effect of strong augmentation (S.A.), i.e., CutMix, in Table 11. We implement our algorithm without S.A. and evaluate all baselines under the two evaluation strategies: with or without S.A. The table shows that the gain by S.A. is only about 1%p whereas the gain by multi-formation and algorithmic development is about 14%p and 9%p (by comparing IDC *w/o* S.A. with DSA and DM). The result verifies that our algorithm does not mainly rely on augmentation.

E.3. Larger Data Storage

In this subsection, we measure the performance of condensation with larger storage budgets. Figure 9 shows the performance of condensed data with large storage budgets of up to 500 images per class on CIFAR-10. The figure shows that IDC outperforms other methods under the storage budgets of 200 images per class, however, IDC underperforms at 500 images per class. This result indicates that increasing the number of synthetic data via multi-formation shows diminishing returns when there are enough storage budgets to represent the original training data diversity (see Appendix A.2 for theoretical analysis). Nonetheless, IDC-I outperforms baselines in all settings, demonstrating the effectiveness of our condensation algorithm with large storage budgets.

Table 10. Top-1 test accuracy of ConvNet-3 trained on condensed datasets (average score with 3 evaluation repetitions).

Img/	SVHN					MNIST					FashionMNIST					CIFAR-10				
Cls	DSA†	KIP†	DM	IDC-I	IDC	DSA†	KIP†	DM	IDC-I	IDC	DSA†	KIP†	DM	IDC-I	IDC	DSA†	KIP†	DM	IDC-I	IDC
1	27.5	39.5	24.2	46.7	**68.5**	88.7	90.1	89.7	88.9	**94.2**	70.6	73.5	70.0	70.7	**81.0**	28.8	38.6	28.9	36.7	**50.6**
10	79.2	64.2	72.0	77.0	**87.5**	97.8	97.5	97.5	98.0	**98.4**	84.6	**86.8**	84.8	85.3	86.0	52.1	49.2	53.8	58.3	**67.5**
50	84.4	73.2	84.3	87.9	**90.1**	**99.2**	98.3	98.6	98.8	**99.1**	88.7	88.0	88.6	**89.1**	86.2	60.6	56.7	65.6	69.5	**74.5**

Table 11. Ablation study on strong augmentation (S.A.), CIFAR-10 (ConvNet, 50 img/cls). We report bold values in Tables 1 and 2. Evaluation *w/o* S.A. is identical to the method by DSA and DM.

Evalution	Random	Herding	DSA	KIP	DM	IDC-I *w/o* S.A.	IDC-I	IDC *w/o* S.A.	IDC
w/o S.A.	54.7	57.5	**60.6**	55.8	63.0	67.4	68.6	72.8	73.5
w/ S.A.	**56.5**	**59.8**	59.5	**57.9**	**65.6**	67.0	**69.5**	74.3	**74.5**

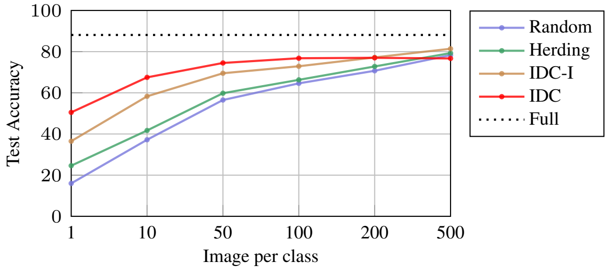


Figure 9. Top-1 test accuracy of ConvNet-3 trained on condensed datasets with increasing data storage (CIFAR-10).

E.4. Network Architecture Effect on Condensation

We analyze the effects of networks' architecture by comparing the performance of condensed data on CIFAR-10. In Table 12, we compare the performance of condensed data with different network architectures; simple convolution networks with various widths and depths, ResNet-10, and ResNetAP-10. Interestingly, simple ConvNets perform better than the deeper ResNet architectures on both test models. Furthermore, ConvNet-4 (*CN-D*) has lower condensation performance than ConvNet-3 (*CN*), although it has more convolution layers. The results indicate that a complex network is not always effective when compressing a large amount of learning information on a small storage capacity.

E.5. Multi-Formation with Another Algorithm

Our multi-formation strategy can be orthogonally applied to other condensation methods. To verify the generality of our strategy, we apply the multi-formation function on another condensation method, DM (Zhao & Bilen, 2021a), which uses a feature matching objective. Table 13 summarizes the test performance of condensed data on CIFAR-10. The table shows that our multi-formation framework consistently improves the performance of DM over various test models, demonstrating the general applicability of our framework.

E.6. Dataset Condensation with ZCA

We implement ZCA following the official KIP code (Nguyen et al., 2021) and test IDC on CIFAR-10 (Table 14).

Table 12. Condensation network architecture comparison (10 img/cls on CIFAR-10). *CN* denotes ConvNet-3, *CN-W* denotes ConvNet-3 with twice more channels (256), *CN-D* denotes ConvNet-4 (4 convolution layers), *RN* denotes ResNet-10, and *RN-AP* denotes ResNet-10 with average pooling instead of strided convolutions for downsampling. Note, we use instance normalization as in Zhao et al. (2021).

Test Model	Condensation Network Architecture				
	CN	CN-W	CN-D	RN	RNAP
ConvNet-3	58.3	**58.8**	56.6	51.4	53.5
ResNet-10	50.2	**50.5**	48.7	47.5	48.8

Table 13. Test accuracy of feature matching objective by DM with our multi-formation strategy (DM+MF) on CIFAR-10.

Pixel/Class	Test Model	DM	DM+MF	IDC
50×32×32	ConvNet-3	65.6	68.4	74.5
(1.0%)	ResNet-10	58.6	63.1	72.4

Table 14. Effects of ZCA whitening on IDC (CIFAR-10 with ConvNet-3). Here, *S.A.* means strong augmentation.

Pixel/Class	IDC *w/o* S.A. + ZCA	IDC + ZCA	IDC
10×32×32	66.6	66.7	**67.5**
50×32×32	72.0	72.5	**74.5**

We find that ZCA results in mild degradation in performance. We speculate that ZCA whitening, which removes pixel correlation, is not suitable for IDC's upsampling process.

F. Visual Examples

We provide visual examples of IDC on CIFAR-10, ImageNet, MNIST, and SVHN in the following pages. In Figures 10 to 13, we compare our synthetic data samples to the real training data samples, which we used as initialization of the synthetic data. From the figure, we find that our synthesized data looks more class-representative. We provide the full condensed data in Figures 14 to 17, under the storage budget of 10 images per class.

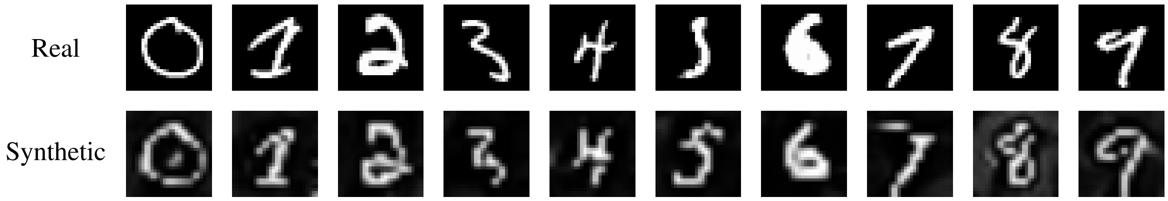


Figure 10. Comparison of real and synthetic images on MNIST.

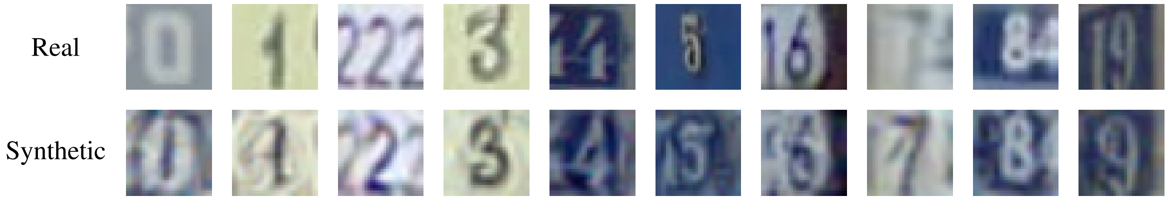


Figure 11. Comparison of real and synthetic images on SVHN.

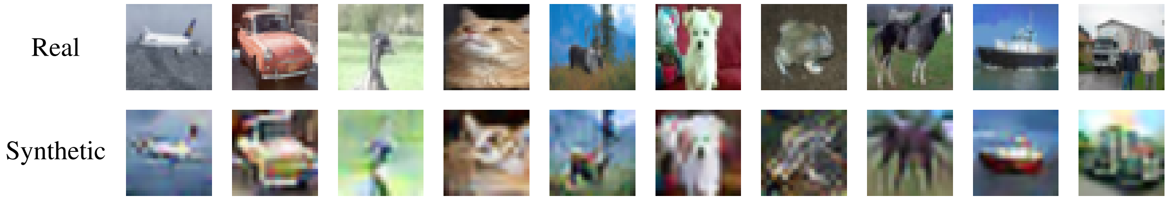


Figure 12. Comparison of real and synthetic images on CIFAR-10.

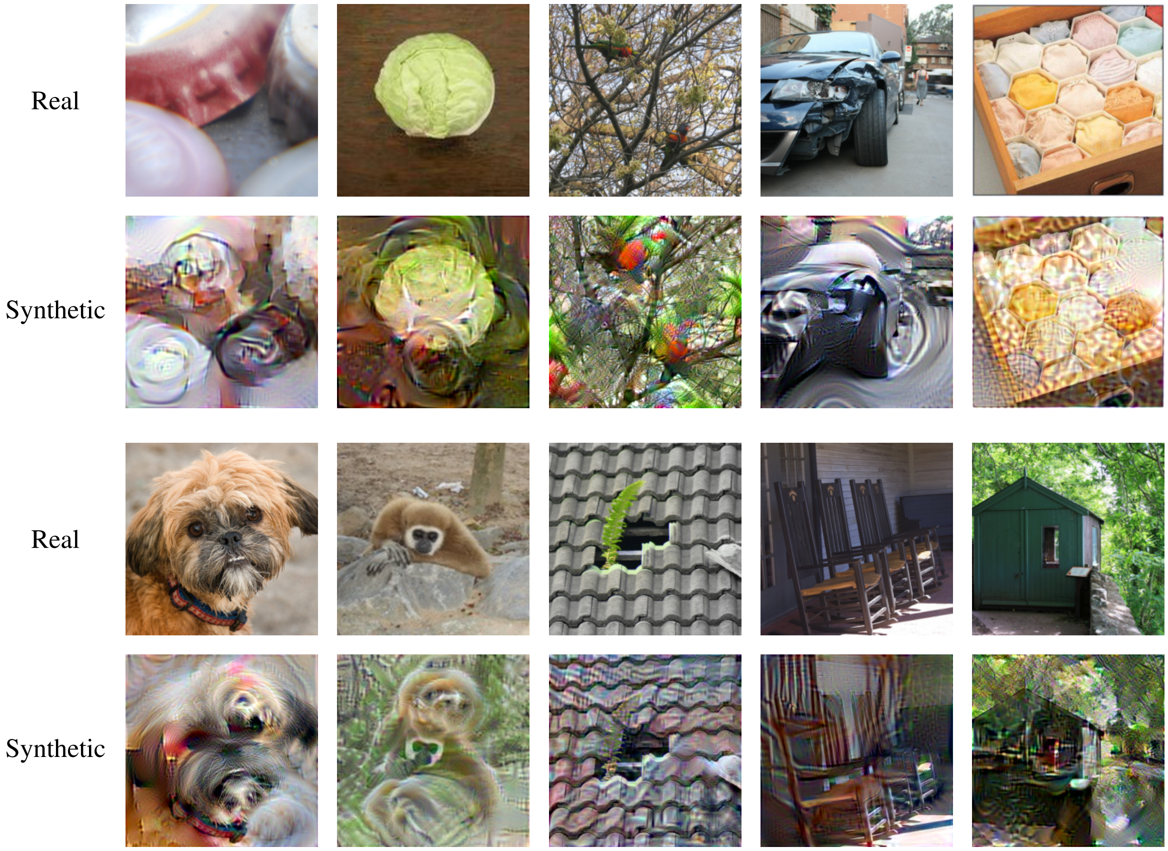


Figure 13. ImageNet: bottle cap, cabbage, lorikeet, car wheel, honeycomb, Shih-Tzu, gibbon, tile roof, rocking chair, and boat house.

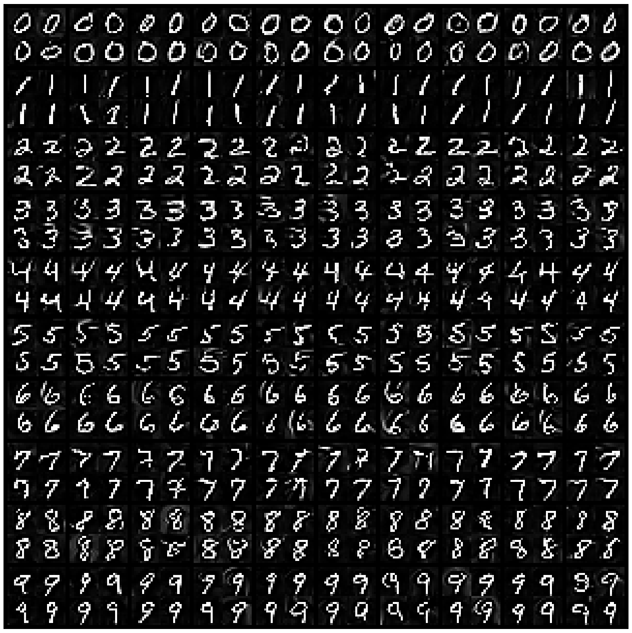


Figure 14. Condensed images of MNIST dataset with IDC (10 × 28 × 28 pixels per class).

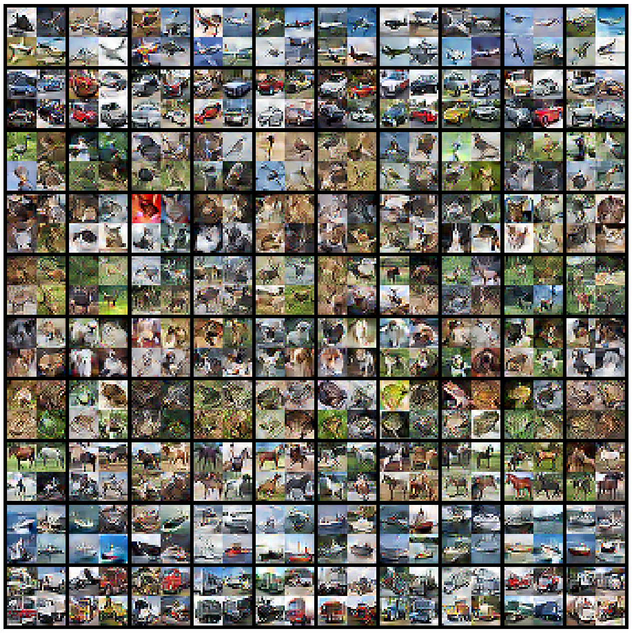


Figure 15. Condensed images of CIFAR-10 dataset with IDC (10× 32 × 32 pixels per class). Each row correspond to the condensed data of a single class. We list the class labels from the first row as follows: 1) airplane, 2) automobile, 3) bird, 4) cat, 5) deer, 6) dog, 7) frog, 8) horse, 9) ship, and 10) truck.

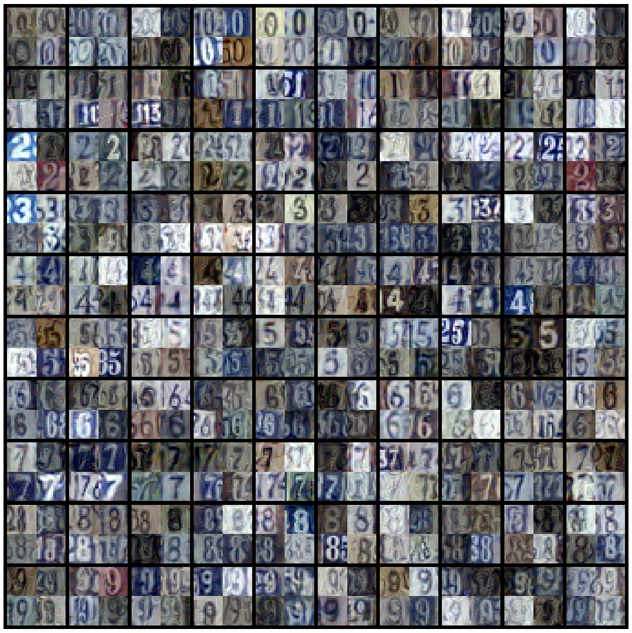


Figure 16. Condensed images of SVHN dataset with IDC (10 × 32 × 32 pixels per class).

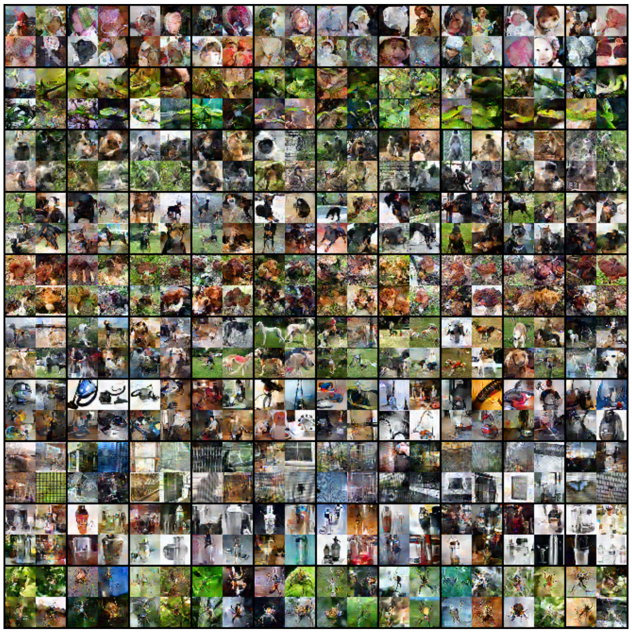


Figure 17. Condensed images of ImageNet-10 dataset with IDC with a multi-formation factor of 2 (10×224×224 pixels per class). Each row correspond to the condensed data of a single class. We list the class labels from the first row as follows: 1) poke bonnet, 2) green mamba, 3) langur, 4) Doberman pinscher, 5) gyromitra, 6) gazelle hound, 7) vacuum cleaner, 8) window screen, 9) cocktail shaker, and 10) garden spider.