Ameliorate Spurious Correlations in Dataset Condensation

Justin Cui [1] **Ruochen Wang** [1] **Yuanhao Xiong** [1] **Cho-Jui Hsieh** [1]

Abstract

Dataset Condensation has emerged as a technique for compressing large datasets into smaller synthetic counterparts, facilitating downstream training tasks. In this paper, we study the impact of bias inside the original dataset on the performance of dataset condensation. With a comprehensive empirical evaluation on canonical datasets with color, corruption and background biases, we found that color and background biases in the original dataset will be amplified through the condensation process, resulting in a notable decline in the performance of models trained on the condensed dataset, while corruption bias is suppressed through the condensation process. To reduce bias amplification in dataset condensation, we introduce a simple yet highly effective approach based on a sample reweighting scheme utilizing kernel density estimation. Empirical results on multiple real-world and synthetic datasets demonstrate the effectiveness of the proposed method. Notably, on CMNIST with 5% bias-conflict ratio and IPC 50, our method achieves 91.5% test accuracy compared to 23.8% from vanilla DM, boosting the performance by 67.7%, whereas applying state-of-the-art debiasing method on the same dataset only achieves 53.7% accuracy. Our findings highlight the importance of addressing biases in dataset condensation and provide a promising avenue to address bias amplification in the process.

1. Introduction

Dataset plays a central role in machine learning model performances. The rapidly growing size of contemporary datasets not only poses challenges to data storage and preprocessing, but also makes it increasingly expensive to train machine learning models and design new methods, such as architectures, hyperparameters, and loss functions. As a result, dataset condensation (also known as dataset distillation (Wang et al., 2018)) emerges as a promising direction for solving this issue. Dataset Condensation aims at compressing the original large-scale dataset into a small subset of information-rich examples, enabling models trained on them to achieve competitive performance compared with training on the whole dataset. The distilled dataset with a significantly reduced size can therefore be used to accelerate model training and reduce data storage. Dataset condensation has been shown to benefit a wide range of machine learning tasks, such as Neural Architecture Search (Wang et al., 2021), Federated Learning (Xiong et al., 2023), Continual Learning (Yang et al., 2023; Gu et al., 2023), Graph Compression (Jin et al., 2021; Feng et al., 2023a), and Multimodality (Wu et al., 2023).

Recent dataset condensation methods mainly focus on performance enhancements on standard datasets (Loo et al., 2023; Zhao et al., 2023; Du et al., 2023; Cui et al., 2023). However, this emphasis often overlooks dataset bias, a critical issue in machine learning. Dataset bias (Tommasi et al., 2017) occurs when collected data unintentionally reflects existing biases, leading to skewed predictions and potential ethical concerns. Despite extensive research on bias detection and mitigation strategies in recent years (Sagawa et al., 2019; Li & Vasconcelos, 2019; Nam et al., 2020; Lee et al., 2021; Hwang et al., 2022), the impact of dataset bias on data condensation remains unexplored. Given that a biased synthetic set can lead to inaccurate or unfair decisions, understanding the role of bias in dataset condensation and developing effective mitigation strategies is crucial.

This paper provides the first study of how biases in the original training set affect dataset distillation process. Specifically, we are interested in the following questions: *1). How does bias propagate from the original dataset to the distilled dataset? 2). How do we mitigate biases present in the distilled dataset?*

To answer the first question, we assess existing dataset condensation algorithms across several benchmark datasets. Our findings reveal that the condensation process is significantly influenced by the type of bias, with color and background biases being amplified and noise bias suppressed.

[1]Department of Computer Science, University of California, Los Angeles. Correspondence to: Justin Cui <justincui@ucla.edu>, Cho-Jui Hsieh <chohsieh@cs.ucla.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

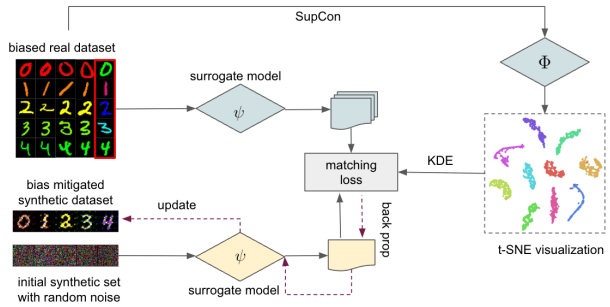

Figure 1. Workflow of our method that utilizes Supervised Contrastive model and Kernel Density Estimation to mitigate bias in the dataset distillation process.

For datasets exhibiting amplified biases, we found that even state-of-the-art de-biasing training methods are insufficient in restoring the original performance. This highlights the urgency of developing a bias-mitigating dataset condensation algorithm.

To counter bias in condensed datasets, we propose a simple yet effective debiasing algorithm based on data point reweighting. Leveraging the insight that biased data points cluster in the model's embedding space, we down-weight samples within such clusters using kernel density estimation. This re-weighting rebalances the significance of biased and unbiased samples, ameliorating biases in the condensation process. Empirical results on diverse bias-injected datasets demonstrate that the proposed reweighting scheme significantly reduces bias in the distilled datasets. For example, on Colored MNIST with a 5% bias in conflicting samples and 50 images per class, the original Distribution Matching (DM) method leads to a biased synthetic set. A model trained on such a distilled set achieves only 23.8% accuracy. In contrast, our reweighting method produces a more balanced dataset, resulting in 91.5% accuracy, representing a 67.7% gain over DM. In summary, our contributions are:

1. We provide the first study on the impact of biases in dataset condensation process.
2. We propose a simple yet effective re-weighting scheme to mitigate biases in two canonical types of dataset condensation methods.
3. Through extensive experiments and ablation studies on both real-world and synthesized datasets, we demonstrate the effectiveness of our method.

2. Related Work

2.1. Dataset Condensation

Dataset Condensation (DC) or Distillation (DD) (Wang et al., 2018; Li et al., 2022) aims to compress a large dataset into a small but informative one to achieve competitive performances compared to the whole dataset. Naturally, it's framed as a bi-level optimization problem using surrogate objectives such as model gradients (Zhao & Bilen, 2021b; Feng et al., 2023b; Du et al., 2024) or training trajectories (Cazenavette et al., 2022; Du et al., 2023; Cui et al., 2023; Guo et al., 2023). The inner loop of the bi-level optimization usually optimizes a surrogate model using the distilled dataset and the outer loop tries to optimize the resulting distilled dataset itself.

Due to the heavy computation involved in the bi-level optimization process, one line of work (Zhao & Bilen, 2021a; Wang et al., 2022; Zhao et al., 2023; Sajedi et al., 2023; Zhang et al., 2024a) tries to directly match the latent feature space. While for other works (Zhou et al., 2022; Loo et al., 2022), they draw inspiration from Kernel Ridge Regression (KRR) (Nguyen et al., 2020) and try to approximate the surrogate model using Kernel methods to tackle the high computation cost. Orthogonally, some methods (Kim et al., 2022; Deng & Russakovsky, 2022; Wei et al., 2024; Shin et al., 2024; Liu & Wang, 2023) propose to compress the dataset into even more compact spaces through parametrization. In order to boost the applicability of the resulting dataset, recent works (Yin et al., 2024; Sun et al., 2023) try to scale DD to large scale datasets such as ImageNet-1K using different approaches such as a 3-stage process or an efficient paradigm that enables both diversity and realism. Pre-trained generative models are also used in recent works (Cazenavette et al., 2023; Zhang et al., 2023; Wang et al., 2023) to enhance the distilled datasets.

Because of the highly compressed dataset size and competitive performance, the distilled dataset can be used for many downstream tasks such as Continual Learning (Yang et al., 2023; Gu et al., 2023), Federated Learning (Xiong et al., 2023; Huang et al., 2023), Graph Neural Networks (Zhang et al., 2024b; Liu et al., 2024), Neural Architecture Search (Such et al., 2020; Medvedev & D'yakonov, 2021), etc. Since most of the parameterization methods can be used as an add-on module to non-parameterization methods, we focus on non-parameterization methods to study the effect of bias in this paper.

2.2. Dataset and Model Bias

Deep Neural Networks (DNNs) have achieved remarkable performance in discovering correlations present within datasets. However, when applied to datasets where simple and spurious correlations coexist with complex and intrinsic correlations, DNNs may inadvertently lean towards the shortcuts. The generalization ability of DNNs will be greatly hindered when these spurious relations are learned instead of the intrinsic ones. Following previous works (Hwang et al., 2022), we refer to samples strongly correlates with bias feature as bias-aligned and samples that don't align with bias features as bias-conflicting. The most commonly

studied bias types include color, background, noise, texture etc (Nam et al., 2020; Lee et al., 2021; Hwang et al., 2022). In this paper, we study how the bias in the original data affects the small synthetic set through the dataset distillation process and try to mitigate this phenomena. The evaluations are done by testing whether models trained on synthetic datasets can effectively generalize to unbiased test datasets.

2.3. De-biasing Methods

While prior research has explored de-biasing methods on entire datasets, our work is the first, to the best of our knowledge, to investigate the impact of biases on dataset distillation results in this emerging field. Below, we discuss several state-of-the-art de-biasing methods designed to train de-biased models on entire datasets.

LfF (Nam et al., 2020) debiases by training a biased and debiased model together. It uses generalized cross entropy (GCE) (Zhang & Sabuncu, 2018) loss to amplify bias and computes a difficulty score for debiased model loss weighting. DFA (Lee et al., 2021) disentangles bias and intrinsic attributes, creating unbiased samples by swapping bias embeddings among training samples to train a debiased model. SelectMix (Hwang et al., 2022) shares the same idea of creating more unbiased samples by using mixup augmentation on contradicting pairs selected by a biased model.

Note that, although some prior de-biasing methods involve reweighting (Sagawa et al., 2019; Li & Vasconcelos, 2019; Nam et al., 2020), our approach differs for the following reasons: 1) Previous methods (Li & Vasconcelos, 2019; Nam et al., 2020) integrate the reweighting scheme into the final de-biased model's training, either as a loss or using an auxiliary model, optimizing it alongside the de-biased model. This process does not apply to dataset condensation methods since a well-trained model is not required in the process (e.g., DM uses randomly initialized models, and MTT only matches part of the training trajectories). 2) Methods like (Sagawa et al., 2019) require explicit bias supervision, which may be challenging or infeasible, while our method has no such requirement.

3. The Impact of Bias in Dataset Condensation

In this section, we conduct comprehensive experiments to answer the following question: *how does a biased training set influence the distilled data? Will the bias be amplified or suppressed through the condensation process?*

In line with prior studies (Nam et al., 2020; Lee et al., 2021; Hwang et al., 2022), we explore three datasets: Colored MNIST (CMNIST), Background Fashion-MNIST (BG FMNIST), and Corrupted CIFAR-10. CMNIST introduces a color bias, causing classes to share specific colors, potentially confusing training. BG FMNIST combines Mini-Places (Zhou et al., 2017) with Fashion-MNIST, creating background biases, e.g. T-shirts in bamboo forests. Corrupted CIFAR-10 introduces perturbations and image distortions like Gaussian noise, blur, brightness, contrast changes, and occlusions. See Figure 7 for examples and Section 5.1 for detailed descriptions.

For each dataset, we use D to denote the unbiased set (e.g., randomly distributed colors in CMNIST), while D_b represents the bias-injected dataset. In D_b, 95% of the samples are aligned with the bias, meaning, for example, 95% of the digit '0' may be red, while the remaining 5% of the digit '0' possess random colors. Let $\mathcal{F}$ be a dataset distillation algorithm that maps the original dataset into a distilled synthetic set, and let $\mathcal{M}$ denote the model training procedure that maps a dataset to a model. By comparing $\mathcal{M}(\mathcal{F}(D))$ (model trained with distilled unbiased dataset) and $\mathcal{M}(\mathcal{F}(D_b))$ (model trained with distilled biased dataset), we evaluate these two models' performance on unbiased test samples and compute the following measurement to reveal how the bias in the source dataset degrades the performance of the model trained on distilled synthetic samples:

$$\text{Acc}(\mathcal{M}(\mathcal{F}(D))) - \text{Acc}(\mathcal{M}(\mathcal{F}(D_b))).$$

We conduct experiments on three representitive dataset distillation algorithms: DSA (Zhao & Bilen, 2021b), DM (Zhao & Bilen, 2021a) and MTT (Cazenavette et al., 2022) as lots of SOTA methods can be used as an add-on module to these methods (Wang et al., 2022; Kim et al., 2022; Liu et al., 2022; Lee et al., 2022). Additionally, we include a baseline approach without dataset condensation, which is equivalent to the case when $\mathcal{F}$ represents an identity transformation. The visualization of bias impacts in dataset condensation can be found in Figure 2.

For CMNIST, the results presented on the left panel of Figure 2 reveal a strong **bias amplification** effect – while the bias injected into the original dataset leads to a mere 4% performance drop in regular training, it results in over 50% performance decline when employing any of the three dataset distillation methods. This observation can be attributed to the following factors. Since the original set comprises 95% biased samples, with a selection of IPC 10, it is highly possible that all of the chosen images are biased. As a result, the unbiased signal totally diminishes through the distillation process. Another critical factor is that since color is a discriminative feature that can be easily learned by neural networks, dataset distillation algorithms will focus on distilling this color feature into the synthetic images in order to achieve good performance, leading to bias amplification. Similar impacts are also seen on the BG FMNIST dataset.

Interestingly, results from Corrupted CIFAR-10 show a reverse trend. The right panel of Figure 2 shows that the

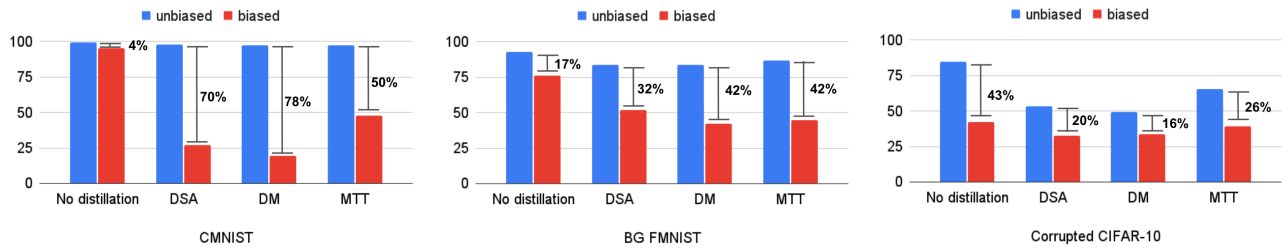

Figure 2. The left most 2 bars indicate the model performance on full dataset with no distillation. For DSA/DM/MTT, the blue bar shows the model performance on the unbiased dataset and the red bar shows the performance of the corresponding dataset distillation method on that biased dataset with 5% bias-conflicting samples. The distillation performances are measured under IPC 10.

performance degradation is actually more substantial in the traditional training pipeline compared to the ones incorporating dataset distillation. This observation indicates the **bias suppression** effect for corruption biases, and dataset distillation is helpful for mitigating bias in this setting. Since corruption biases include several different perturbation effects such as Gaussian noise and blurring, we assume that the distillation process naturally blends information from multiple images, and the resulting images are already blurred in nature where noisy effects tend to cancel out, so it is harder to capture the corruption bias. Sampled distilled images are visualized in Figure 7.

For datasets exhibiting amplified biases (CMNIST and BG FMNIST), we found that even state-of-the-art de-biasing training methods such as SelectMix (Hwang et al., 2022) and DFA (Lee et al., 2021) are not able to obtain an unbiased model from the biased synthetic set. See more details in Section 6. This finding also highlights the urgency of developing a bias-mitigating DC algorithm to obtain unbiased synthetic sets.

4. Ameliorate Bias in Dataset Condensation

4.1. Bias Amelioration Through Re-weighting

In dataset condensation methods such as DM (Zhao & Bilen, 2021a; Zhao et al., 2023), the objective is to match the embeddings generated by the synthetic dataset ($\mathcal{S}$) with the ones generated by the real images ($\mathcal{T}$). The objective function is formulated as below

$$\min_{\mathcal{S}} \mathop{\mathbb{E}}_{\substack{v\sim P_v \\ \omega\sim\Omega}} \parallel \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\psi_v(\mathcal{A}(x_i,\omega)) - \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\psi_v(\mathcal{A}(s_j,\omega)) \parallel^2, \tag{1}$$

where ψ is usually a surrogate model that maps the data into an embedding space and $\mathcal{A}$ is a differentiable augmentation function. By solving (1), DM learns the synthetic set to match the mean embedding of the original set. If the dataset is highly biased, the first term $\frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\psi_v(\mathcal{A}(x_i,\omega))$ will be dominated by bias-aligned samples, thus causing distribution matching based methods to synthesize more bias-aligned images.

To mitigate bias, we propose to compute a weighted sum of the real image embeddings instead of simply using the mean of all data points. For data points that exhibit a strong correlation with the spurious (bias) feature, they should be assigned lower weights. Conversely, data points that have limited association with the bias feature should be afforded higher weights. This adjustment ensures that the distillation process effectively captures the intrinsic features. Let $W(\mathcal{T}) = [w_0, w_1, ..., w_n]$ with n equals $|\mathcal{T}|$ in Equation (1) be the normalized weight of each training sample, the weighted loss can be written as

$$\min_{\mathcal{S}} \mathop{\mathbb{E}}_{\substack{v\sim P_v \\ \omega\sim\Omega}} \parallel W(\mathcal{T})\cdot\psi_v(\mathcal{A}(\mathcal{T},\omega)) - \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\psi_v(\mathcal{A}(s_j,\omega)) \parallel^2, \tag{2}$$

where $W(\mathcal{T})\cdot\psi_v(\mathcal{A}(\mathcal{T},\omega))$ is the re-weighted embeddings where the bias feature has been balanced.

4.2. Bias Estimation Using Kernel Density Estimation

The key problem of this reweighting scheme is how to compute $W(X)$ which is unknown. In order to compute it, we propose to use Kernel Density Estimation (KDE), which is a non-parametric technique used to estimate the probability density function (PDF) of a random variable based on observed data points. Mathematically, given a set of n data points $x_1, x_2, ..., x_n$, the kernel density estimate $\hat{f}(x)$ at any point x is given by:

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} K\left(\|\Phi(x) - \Phi(x_i)\|\right), \tag{3}$$

where $K(\cdot)$ is the kernel function that determines the shape and width of the kernel placed on each data point. The most commonly used kernel function is the Gaussian kernel: $K(u) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\cdot(\frac{u}{\sigma})^2}$. Φ is a function that maps the data points into an embedding space for distance computation.

By summing the contributions of the kernel functions centered at each data point, KDE provides an estimate of the PDF at any given point x. The estimate $\hat{f}(x)$ represents the density of the underlying distribution at that point. Since a biased dataset is usually dominated by bias-aligned samples which should be given a lower weight, we propose to use the normalized inverse of the kernel density function $\mathbb{N}(\frac{1}{\hat{f}(x)})$ as the new weights, and $\mathbb{N}$ is a normalization function such as softmax so that $\sum_{i=1}^{|\mathcal{T}|}\mathbb{N}(\frac{1}{\hat{f}(x_i)}) = 1$. Eventually we have $W(\mathcal{T}) = [\mathbb{N}(\frac{1}{\hat{f}(x_1)}), ..\mathbb{N}(\frac{1}{\hat{f}(x_n)})]$.

4.3. Distance Computation in KDE

The aforementioned KDE reweighting scheme requires a feature mapping $\Phi(\cdot)$ to map images from raw pixel space to a more meaningful hidden space. Moreover, an optimal mapping for $\Phi(\cdot)$ would be one that transforms images into biased features, enabling KDE to accurately represent the density of bias. Consequently, our reweighting scheme can effectively mitigate bias. To obtain bias features without external knowledge, DFA (Lee et al., 2021) and LfF (Nam et al., 2020) both utilize the generalized cross entropy (GCE) loss to train an auxiliary bias model. However, neither of them directly work with latent spaces that can be utilized in KDE. Following the recent SOTA de-biasing method SelecMix (Hwang et al., 2022), we utilize a supervised contrastive learning (Chen et al., 2020; Khosla et al., 2020) model, which is trained with generalized supervised contrastive (GSC) loss to produce image embeddings that can be used to measure distances between data points. The model is proven to produce high quality similarity matrix regarding bias features (Hwang et al., 2022). See Figure 1 for the visualization of our overall workflow.

4.4. Apply to Other Dataset Condensation Methods

In addition to DM and its follow up works such as DREAM (Liu et al., 2023) and IDM (Zhao et al., 2023), our method can also be easily applied to other dataset condensation methods. Here we use DSA (Zhao & Bilen, 2021b; 2023; Kim et al., 2022) as an example which synthesizes data by matching the gradients between surrogate models trained with real data and synthetic data. Generally, gradient matching based methods can be formulated as:

$$\min_{\mathcal{S}} \; D(\nabla_\theta \mathcal{L}_c^{\mathcal{T}}(\mathcal{A}(\mathcal{T}, \omega^{\mathcal{T}}), \theta_t), \nabla_\theta \mathcal{L}_c^{\mathcal{S}}(\mathcal{A}(\mathcal{S}, \omega^{\mathcal{S}}), \theta_t)), \tag{4}$$

where $\mathcal{L}_c^{\mathcal{T}} = \frac{1}{|\mathcal{T}|}\sum_{x,y}\ell(\phi_{\theta_t}(\mathcal{A}(\mathcal{T}, \omega^{\mathcal{T}}))$. We denote the re-computed weights as $W(\mathcal{T})$, then the first term in Equation (4) can be replaced with $W(\mathcal{T}) \cdot \nabla_\theta \mathcal{L}_c^{\mathcal{T}}(\mathcal{A}(\mathcal{T}, \omega^{\mathcal{T}}), \theta_t)$. We demonstrate that the proposed method works well with both DSA and DM in the following sections.

5. Experiments

5.1. Evaluation Datasets

Following previous works (Lee et al., 2021; Hwang et al., 2022), we first evaluate our method on 3 canonical biased datasets including two synthesized datasets Colored MNIST (CMNIST) and Corrupted CIFAR-10 and one real-world dataset BFFHQ. CMNIST originates from MNIST (Deng, 2012) dataset with added color correlation such as most digit 0 being red and digit 4 being green. Corrupted CIFAR10 applies different corruptions to the images in CIFAR-10 so that one class are associated with one type of corruption such as GaussianNoise or MotionBlur. BFFHQ (Lee et al., 2021) is constructed from FFHQ (Karras et al., 2019) which contains human face images paired with their corresponding facial attribute annotations. Age is selected as the label and gender is selected as the biased feature. We also contribute one novel dataset Background Fashion-MNIST (BG FMNIST) to evaluate background biases which is constructed by using Fashion-MNIST (Xiao et al., 2017) as foregrounds and MiniPlaces (Zhou et al., 2017) as backgrounds. Similar to SelecMix (Hwang et al., 2022) and DFA (Lee et al., 2021), we evaluate our methods under 3 settings with 1%,2%,5% bias conflicting samples[1] for CMNIST, Corrupted CIFAR-10 and BG FMNIST and 0.5% bias conflicting ratios for BFFHQ[2]. More detailed description regarding datasets can be found in Appendix A.5.

5.2. Experimental Results

We evaluate the performance of 4 representative dataset condensation methods: DM (Zhao & Bilen, 2021a) , DSA (Zhao & Bilen, 2021b), DREAM (Liu et al., 2023) and IDM (Zhao et al., 2023). We then use the proposed algorithm to improve DM and DSA algorithms in the experiments. Note that our algorithm can also be applied to other existing condensation methods, as discussed in Appendix A.1. Also, see Appendix A.7 for a qualitative analysis and Appendix A.2 for specific hyper-parameter settings.

5.2.1. Performance boost on DM

We first investigate if the proposed method can mitigate biases in distribution matching-based methods such as DM. The evaluation results are shown in Table 1. It can be seen that as the number of IPCs increases, the vanilla DM has little to no performance gain on bias amplifying datasets CMNIST and BG FMNIST. When IPC increases from 1 to 10 on CMNIST with 1, 2 and 5 percent bias-conflicting

[1]Bias-conflicting samples are samples in a class that have different bias properties from the majority of the samples in that class. For example, a yellow 0 is a bias conflicting sample when the majority of 0s are red.

[2]BFFHQ only comes with 0.5% bias conflicting ratio (Lee et al., 2021).

Table 1. Test accuracy of baseline methods and our methods on distilled datasets with varying bias-conflict ratios.

Dataset	Method	Bias-conflict Ratio (1.0%)			Bias-conflict Ratio (2.0%)			Bias-conflict Ratio (5.0%)		
		1	10	50	1	10	50	1	10	50
CMNIST	Random	22.6±1.0	13.5±0.3	16.0±0.1	22.8±1.0	16.2±0.1	20.7±0.1	19.9±0.5	17.4±0.4	27.2±0.2
	MTT	24.2±0.3	27.6±0.4	18.1±0.6	**28.3±0.3**	42.0±0.4	26.0±0.5	29.2±0.9	47.7±0.8	33.9±1.2
	IDM	20.1±0.9	18.8±1.2	17.6±1.6	20.2±1.0	20.1±0.5	19.1±1.3	20.8±0.9	22.3±1.1	26.5±1.5
	DREAM	21.3±0.4	15.9±1.2	17.9±0.8	26.2±0.5	17.1±1.3	17.5±1.5	23.7±1.2	30.8±1.5	50.4±1.0
	DM	25.4±0.1	18.6±0.2	22.6±0.5	24.8±0.4	18.5±0.6	23.6±0.8	25.3±0.3	19.6±0.9	23.8±1.3
	DM+Ours	**28.0±0.5**	64.9±0.3	75.4±1.1	26.4±0.7	50.6±1.2	75.7±1.0	32.2±1.0	86.5±1.2	91.5±0.9
	DSA	26.1±0.3	16.5±0.2	14.5±0.2	25.2±0.3	16.8±0.3	30.9±0.4	25.9±0.5	27.3±0.4	68.5±1.2
	DSA+Ours	27.9±0.4	**76.7±1.1**	**81.4±0.8**	26.4±0.2	**75.3±0.3**	**83.0±1.2**	**32.6±0.1**	**91.9±0.7**	**94.0±0.8**
BG FMNIST	Random	40.0±0.2	40.4±1.6	35.2±0.4	30.2±0.6	43.2±0.2	33.5±1.0	36.2±1.3	44.6±1.2	41.2±1.1
	MTT	39.0±1.2	48.0±1.5	45.3±0.9	38.9±1.4	59.2±1.1	59.3±0.8	48.1±1.4	45.2±1.3	62.3±0.8
	IDM	40.7±1.0	42.3±0.9	38.4±1.2	41.1±0.8	37.2±1.3	40.5±0.9	43.4±0.4	46.6±0.8	42.0±1.2
	DREAM	39.7±0.9	46.4±1.2	46.1±1.5	45.0±1.0	46.0±0.8	44.5±0.8	43.5±0.9	52.4±1.5	53.2±1.2
	DM	41.0±0.3	42.2±0.8	43.9±0.4	40.1±0.6	40.1±0.9	44.4±0.5	41.7±0.5	42.0±1.2	44.6±0.9
	DM+Ours	**44.6±0.5**	50.6±0.2	57.2±0.6	**51.4±0.7**	62.3±0.4	63.0±1.0	**49.4±0.2**	61.8±0.6	65.0±0.8
	DSA	43.4±0.4	45.8±0.5	40.7±0.9	43.7±0.5	47.6±0.3	48.4±0.8	44.7±0.6	52.8±0.5	59.3±0.6
	DSA+Ours	44.4±0.6	**57.0±1.0**	**58.3±0.8**	48.5±1.2	**64.4±0.9**	**65.1±0.8**	46.2±0.6	**66.4±0.6**	**71.2±1.1**
Corrupted CIFAR-10	Random	16.4±0.6	26.9±0.2	32.7±0.3	19.1±0.1	23.2±0.2	33.5±0.1	11.8±0.1	26.4±0.3	34.2±0.4
	MTT	23.5±0.4	25.4±1.5	33.3±0.5	24.1±0.3	**36.3±0.4**	35.7±0.2	24.2±0.8	**39.0±0.3**	39.5±0.4
	IDM	**26.3±1.0**	30.2±0.4	36.6±0.8	**26.2±0.8**	29.5±0.2	35.0±0.7	26.0±0.8	31.4±1.1	36.5±0.9
	DREAM	23.7±0.9	25.0±1.2	33.6±0.9	25.9±0.8	25.2±0.6	32.1±1.3	25.6±0.9	24.5±0.8	33.7±1.2
	DM	25.1±0.4	32.9±0.3	37.6±0.8	25.0±0.1	32.9±0.1	37.7±0.2	24.6±0.4	33.5±0.8	38.7±0.4
	DM+Ours	24.2±1.2	**33.4±0.9**	**39.4±0.8**	25.3±0.5	34.2±0.5	**39.7±0.4**	**26.6±0.5**	33.5±0.6	**40.2±0.4**
	DSA	25.5±0.3	31.9±0.8	34.1±0.5	25.1±0.2	32.0±0.1	34.2±0.3	25.7±0.5	32.8±0.6	35.6±0.5
	DSA+Ours	26.0±0.1	32.6±0.8	35.0±0.6	25.2±0.8	33.2±0.2	35.8±0.6	26.0±0.3	32.5±0.7	36.6±0.3

1. **Results** in bold show the best performance. Results with underline show the second best performance.
2. See Appendix A.1 for extending our method to DREAM (Liu et al., 2023) and IDM (Zhao et al., 2023) which achieves SOTA results based on DSA and DM.
3. See Appendix A.3 for the results under extremely low bias-conflict ratio (0.5%) where our method also achieves strong performances.

samples, there is even a performance degradation. After reweighting the samples according to Equation (2), we are able to mitigate the biases effectively. Under the settings of IPC 50, we are able to boost the performance from 22.6% to 75.4% on CMNIST dataset with 1% bias conflicting samples. With 2% and 5% bias conflicting samples, the accuracy also increased from 23.6% to 75.7% and 23.8% to 91.5% respectively. The complete results can be found in Tables 1 and 2. Similar performance boosts are also observed on BG FMNIST, e.g. with IPC 10, the performance gains are 8.4%, 22.2% and 19.8% with 1, 2 and 5 percent bias conflicting samples. As shown in previous works (Lee et al., 2021; Hwang et al., 2022), de-biasing becomes more challenging on real-world dataset BFFHQ which only comes with 0.5% bias conflict samples. However, we still see an overall improvement of up to 6.2%. On Corrupted CIFAR-10, we only observe a slight performance boost which aligns with the intuition described in Section 3.

5.2.2. Performance boost on DSA

Next, we investigate whether the proposed method can improve gradient matching based methods and choose DSA as the representative algorithm. Similar to DM, we also observe a strong performance boost on CMNIST. Under the settings of IPC 50, with 1, 2 and 5 percent bias conflicting samples, the performance increases from 14.5% to 81.4%, 30.9% to 83.0% and 68.5% to 94.0%. On BG FMNIST, the performances increase from 40.7% to 58.3%, 48.4% to 65.1% and 59.3% to 71.2% for 1, 2 and 5% bias conflicting samples with IPC 50. On BFFHQ, we see a performance increase of up to 2.8% even with the extremely low bias-conflict ratio. Complete results are shown in Tables 1 and 2.

Table 2. Test accuracy on real-world dataset BFFHQ with varying IPCs. BFFHQ dataset only comes with 0.5% bias-conflict samples which makes bias mitigation challenging. Our method is still able to boost the performance by up to 6.2%.

IPC	Random	MTT	DREAM	DM	DM+Ours	DSA	DSA+Ours
1	50.8±1.5	52.5±1.2	54.0±1.2	58.3±0.3	**64.5±1.5**	61.1±1.0	62.3±0.2
10	50.2±0.8	58.6±1.3	61.2±1.9	63.6±1.0	**65.2 ±1.7**	62.6±1.6	64.0±1.5
50	50.5 ±1.2	-	64.7±1.2	63.9±1.2	**65.6± 0.9**	58.7±1.4	61.5±0.7

* We aren't able to get IDM's performance because it's OOM on all settings for BFFHQ dataset.

5.2.3. Compare to other SOTA methods

The performance of 3 other representative methods, DREAM, IDM and MTT, are also included in Tables 1 and 2. We observe that MTT outperforms vanilla DM and DSA, achieving better results on CMNIST and BG FMNIST. For instance, on BG FMNIST with 5% bias-conflicting samples and IPC 50, MTT achieves 62.3% accuracy compared to DM's 44.6% under the same settings. We think the reason is that MTT doesn't use real images during distillation phase

but the trajectories from teacher models. Thus its performance is determined by both the teacher model trajectories (directly through trajectory matching) and biased original dataset (indirectly through teacher models trained using the biased original dataset). However, it also struggles with biases on general, whereas our proposed method is able to mitigate biases effectively on CMNIST, BG FMNIST and BFFHQ. DREAM and IDM, based on DSA and DM, tend to perform worse on biased datasets. This is partly due to their attempt to extract more information from the original dataset, potentially introducing additional biases. For instance, DREAM's use of Kmeans for selecting representative samples can lead to exclusion of unbiased samples in a biased dataset. See Appendix A.1 for results of applying our method to mitigate biases in DREAM and IDM.

5.3. Runtime Analysis

Our method has a similar runtime overhead to the state-of-the-art de-biasing method SelecMix (Hwang et al., 2022). It consists of two parts: 1) training a ResNet18 (He et al., 2016a) with a 128-dimensional MLP projection head, typically converging within 30 minutes which is around 5% overhead on DSA which usually requires more than 10 hours to finish with IPC 50 on the hardware, and 2) computing pairwise distances and KDE within each training batch, with a time complexity of $\mathcal{O}(B^2d)$, where B is the batch size (usually 256) and d is the projection dimension (128 in our case). With DSA and IPC 50, it takes 55 seconds for 10 iterations without KDE and 65 seconds with KDE, resulting in an 18.2% overhead. Note that, unlike SelecMix which updates the SupCon model during training and repeatedly recomputes pairwise distances, our pretrained SupCon model allows us to cache and reuse pairwise distances throughout the entire condensation process to mitigate this overhead. More details can be found in Appendix A.12.

6. Ablation Study

1. Apply de-biasing methods to synthetic dataset. Do we really need to mitigate bias in the DD process? Even though vanilla dataset distillation methods result in the biased synthetic set, can we still obtain an unbiased model from such a biased set using existing de-biasing training algorithms? In order to answer this question, we apply two SOTA de-biasing methods, SelectMix (Hwang et al., 2022) and DFA (Lee et al., 2021) on the synthetic dataset generated by DM and present the results in Table 3. The experiments are conducted on CMNIST and BG FMNIST with 5% bias-conflicting samples and IPC 10 and 50. We have the following observations from the experiments: 1) DFA can slightly improve the performance on CMNIST but suffers from severe performance degradation on BG FMNIST. We think the reason is that DFA relies on a bias model to split embeddings, which suffers from performance degradation when the dataset becomes more complex, thus causing the overal performance drop. This aligns with the observations in (Hwang et al., 2022). 2) While SelectMix consistently mitigates bias to some extent, it cannot fully rectify biases in synthetic datasets. For instance, SelectMix improves performance from 23.8% to 53.7% on CMNIST IPC 50. However, under the same bias conditions, standard training without dataset distillation achieves approximately >85% accuracy. This indicates that even state-of-the-art methods struggle to recover from biases amplified synthetic sets. This observation aligns with Figure 4, which illustrates the severe bias amplification to the point where there isn't a single unbiased sample (in this case, a digit with a different color) in the synthetic set. Consequently, de-biasing becomes impossible in such a scenario. In contrast, our method achieves 91.5% accuracy in this case, demonstrating the critical importance of debiasing during the dataset distillation procedure.

Table 3. Ablation study test accuracy (%) on applying de-biasing method to train with synthetic datasets from DM, assessed under 5% bias-conflicting samples and IPC 10 and 50.

	CMNIST		BG FMNIST	
IPC	10	50	10	50
DM	19.6±0.9	23.8±1.3	42.0±1.2	44.6±0.9
DFA	25.8±1.0	31.2±1.3	11.0±2.1	17.6±1.9
SelecMix*	43.3±1.3	53.7±1.5	57.2±1.1	58.7±0.9
DM+Ours	**86.5±1.2**	**91.5±0.9**	**61.8±0.6**	**65.0±0.8**

* SelecMix has two versions; we choose the LfF-based (Nam et al., 2020) version for its superior performance

Remark 1 *While applying a state-of-the-art debiasing method typically enhances performance with a biased synthetic dataset, it still doesn't match up to the effectiveness of our KDE-based approach.*

2. Applying de-biasing methods to surrogate models. Many dataset distillation methods use surrogate models to distill information into synthetic sets. To mitigate bias, applying existing de-biasing methods to these surrogate models is another straightforward idea. Here, we explore the impact of applying de-biasing methods to surrogate models in DM and MTT. The experiments are conducted on CMNIST with 5% bias conflicting sample. Since DM tries to match the distribution in the embedding space, we first apply DFA (Lee et al., 2021) to separate the embedding into intrinsic (shape of the digits) and bias (color of the digits) parts. Then we have DM match only the intrinsic part. For MTT, we first generate the de-biased expert training trajectories using SelecMix (Hwang et al., 2022). Then we have MTT match the de-biased training trajectories. The results are shown in Table Table 4. We observe that there is a slight performance increase for DM+DFA compared to the

Table 4. Ablation study test accuracy (%) on applying de-biasing methods to surrogate models on CMNIST with 5% bias-conflict samples and IPC 1, 10 and 50.

Method	IPC 1	IPC 10	IPC 50
DM	25.3±0.3	19.6±0.9	23.8±1.3
DM+DFA	26.1±0.3	20.5±0.5	25.4±0.4
MTT	29.2±0.9	47.7±0.8	33.9±1.2
MTT+SelecMix	18.1±0.5	29.9±0.8	52.1±0.3
DM+Ours	**32.2±1.0**	**86.5±1.2**	**91.5±0.9**

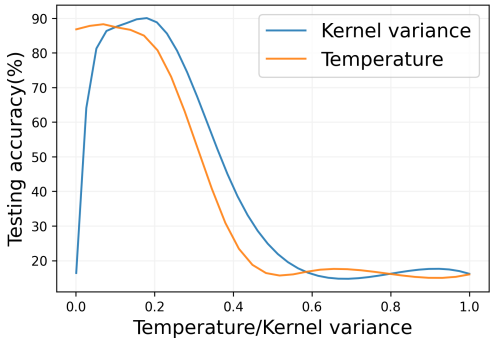

Figure 3. Ablation study on Kernel variance and temperature on CMNIST with 5% bias-conflicting samples and IPC 10.

vanilla DM which validates that the embeddings matched includes less biases. However, as SelecMix points out, fully separating intrinsic and bias parts is challenging. Thus biases will still be distilled into the synthetic dataset even if we only perform DM on the intrinsic embeddings. For MTT+SelecMix, we see mixed results such as a 17.8% drop on IPC 10 and an 18.2% increase on IPC 50. We think the reason is that the de-biased expert training trajectories are not stable due to the use of auxiliary models. Although the final de-biased teacher model performs well, the intermediate training trajectories are hard for MTT to match.

Remark 2 *Our experiments reveal that biased features will still dominate even when using de-biased surrogate models during dataset condensation. This may be due to sub-optimal bias feature disentanglement and less stable matching trajectories.*

3. Ablation study on hyper-parameters We assess two hyper-parameters in our method, the kernel variance and temperature of $\mathbb{N}$, on CMNIST with 5% bias-conflicting samples and IPC 10 and present the results in Figure 3. We use DM as the base method which achieves similar results but runs much faster than DSA. In our observation, a very small variance introduces noise to the estimation, while a large value prevents the algorithm from assigning more weight to bias samples, leading to degraded performances. In general, choosing σ^2 to be 0.1 works well, so we fix it in all of our experiments.

Table 5. Ablation study test accuracy (%) on creating de-biased dataset first using SelecMix and then applying dataset distillation method on it. The experiment is run on CMNIST with IPC 10.

Method	Bias-conflict Ratio 1%	2%	5%
DM	18.6±0.2	18.5±0.6	19.6±0.9
SelecMix+DM	44.6±1.0	25.6±0.9	65.6±0.8
DM+Ours	**64.9±0.3**	**50.6±1.2**	**86.5±1.2**

We also study the impact of softmax temperature when normalizing scores in Equation (3). When the temperature goes up, the weights are more evenly distributed among all samples. When the temperature goes down, more weights are given to samples that are further away from the rest of the samples. As shown in the figure, best performances are achieved around 0.1 which is the default setting.

Remark 3 *It can be seen that our method is robust to different hyper-parameter settings.*

4. Ablation study on learning from de-biased dataset Recent state-of-the-art de-biasing methods such as DFA (Lee et al., 2021) and SelecMix (Hwang et al., 2022) try to train the de-biased model by creating a less biased data representation. This leads us to wonder: Can we follow the same procedure and have DD methods train on de-biased representations generated by these methods? In this ablation study, we run DM on de-biased dataset generated by one of the recent SOTA methods SelecMix and show the results in Table 5. The experiment is conducted on CMNIST with IPC 10 and varying bias-conflict ratios. As we can see from the table that the performances are greatly boosted after creating a less biased representation and have DM learn from it. This aligns with our earlier observation that dataset distillation methods can effectively learn from unbiased datasets. However, its performance still falls short compared to our KDE based method. We think the reason is that KDE estimates a smoothed distribution of the biased and unbiased samples while SelecMix combines two data points with lowest similarity (e.g. intra-class) which depends heavily on the auxiliary model quality.

Remark 4 *Creating an unbiased dataset first and then applying dataset condensation methods on it also greatly boost the model performance. However it still falls short by a large margin compared to our method.*

7. Conclusion

This paper conducts the first analysis of dataset bias in dataset distillation. Our findings show that bias type greatly influences distillation behavior (amplification vs. suppression). Then we introduce a debiasing method using reweight and kernel density estimation which substantially

reduces retained bias in distilled datasets. We assess our debiasing method on various benchmark datasets with different bias ratios and IPC values and empirically verify the effectiveness of our method. In summary, our study offers insights into bias in dataset condensation, presents a practical algorithm for better performance and paves the way for future research on bias mitigation.

Limitations. Although many dataset condensation methods rely on the matching of real and synthetic dataset through carefully designed objective functions, there are methods such as MTT where our method cannot be easily applied due to the use of expert training trajectories instead of real images directly. This will be the focus of our future research.

Impact Statement

Our paper will have the following positive social impacts: 1) We are the first revealing bias amplification effect in data condensation processes, which can prevent the misuse of such technique. 2) We propose a simple yet effective way to mitigate bias when data condensation is used, which can lead to unbiased decisions when employing those models.

Acknowledgements

The work is partially supported by NSF 2048280, 2331966, 2325121, 2244760 and ONR N00014-23-1-2300.

References

Cazenavette, G., Wang, T., Torralba, A., Efros, A. A., and Zhu, J.-Y. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4750–4759, 2022.

Cazenavette, G., Wang, T., Torralba, A., Efros, A. A., and Zhu, J.-Y. Generalizing dataset distillation via deep generative prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3739–3748, 2023.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Cui, J., Wang, R., Si, S., and Hsieh, C.-J. Scaling up dataset distillation to imagenet-1k with constant memory. In *International Conference on Machine Learning*, pp. 6565–6590. PMLR, 2023.

Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Deng, Z. and Russakovsky, O. Remember the past: Distilling datasets into addressable memories for neural networks. *Advances in Neural Information Processing Systems*, 35:34391–34404, 2022.

Du, J., Jiang, Y., Tan, V. Y., Zhou, J. T., and Li, H. Minimizing the accumulated trajectory error to improve dataset distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3749–3758, 2023.

Du, J., Shi, Q., and Zhou, J. T. Sequential subset matching for dataset distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

Feng, Q., Jiang, Z., Li, R., Wang, Y., Zou, N., Bian, J., and Hu, X. Fair graph distillation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a.

Feng, Y., Vedantam, S. R., and Kempe, J. Embarrassingly simple dataset distillation. In *The Twelfth International Conference on Learning Representations*, 2023b.

Gu, J., Wang, K., Jiang, W., and You, Y. Summarizing stream data for memory-restricted online continual learning. *arXiv preprint arXiv:2305.16645*, 2023.

Guo, Z., Wang, K., Cazenavette, G., Li, H., Zhang, K., and You, Y. Towards lossless dataset distillation via difficulty-aligned trajectory matching. *arXiv preprint arXiv:2310.05773*, 2023.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016b.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations, Mar 2019.

Huang, C.-Y., Jin, R., Zhao, C., Xu, D., and Li, X. Federated virtual learning on heterogeneous data with local-global distillation. *arXiv preprint arXiv:2303.02278*, 2023.

Hwang, I., Lee, S., Kwak, Y., Oh, S. J., Teney, D., Kim, J.-H., and Zhang, B.-T. Selecmix: Debiased learning by contradicting-pair sampling. *Advances in Neural Information Processing Systems*, 35:14345–14357, 2022.

Jin, W., Zhao, L., Zhang, S., Liu, Y., Tang, J., and Shah, N. Graph condensation for graph neural networks. *arXiv preprint arXiv:2110.07580*, 2021.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Kim, J.-H., Kim, J., Oh, S. J., Yun, S., Song, H., Jeong, J., Ha, J.-W., and Song, H. O. Dataset condensation via efficient synthetic-data parameterization. In *International Conference on Machine Learning*, pp. 11102–11118. PMLR, 2022.

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

Lee, H. B., Lee, D. B., and Hwang, S. J. Dataset condensation with latent space knowledge factorization and sharing. *arXiv preprint arXiv:2208.10494*, 2022.

Lee, J., Kim, E., Lee, J., Lee, J., and Choo, J. Learning debiased representation via disentangled feature augmentation. *Advances in Neural Information Processing Systems*, 34:25123–25133, 2021.

Li, G., Zhao, B., and Wang, T. Awesome dataset distillation. https://github.com/Guang000/Awesome-Dataset-Distillation, 2022.

Li, Y. and Vasconcelos, N. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9572–9581, 2019.

Liu, S. and Wang, X. Few-shot dataset distillation via translative pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18654–18664, 2023.

Liu, S., Wang, K., Yang, X., Ye, J., and Wang, X. Dataset distillation via factorization. *Advances in Neural Information Processing Systems*, 35:1100–1113, 2022.

Liu, Y., Gu, J., Wang, K., Zhu, Z., Jiang, W., and You, Y. Dream: Efficient dataset distillation by representative matching. *arXiv preprint arXiv:2302.14416*, 2023.

Liu, Z., Zeng, C., and Zheng, G. Graph data condensation via self-expressive graph structure reconstruction. *arXiv preprint arXiv:2403.07294*, 2024.

Lloyd, S. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

Loo, N., Hasani, R., Amini, A., and Rus, D. Efficient dataset distillation using random feature approximation. *Advances in Neural Information Processing Systems*, 35: 13877–13891, 2022.

Loo, N., Hasani, R., Lechner, M., and Rus, D. Dataset distillation with convexified implicit gradients. *arXiv preprint arXiv:2302.06755*, 2023.

Medvedev, D. and D'yakonov, A. Learning to generate synthetic training data using gradient matching and implicit differentiation. In *International Conference on Analysis of Images, Social Networks and Texts*, pp. 138–150. Springer, 2021.

Nam, J., Cha, H., Ahn, S., Lee, J., and Shin, J. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33: 20673–20684, 2020.

Nguyen, T., Chen, Z., and Lee, J. Dataset meta-learning from kernel ridge-regression. In *International Conference on Learning Representations*, 2020.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Sajedi, A., Khaki, S., Amjadian, E., Liu, L. Z., Lawryshyn, Y. A., and Plataniotis, K. N. Datadam: Efficient dataset distillation with attention matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17097–17107, 2023.

Shin, D., Shin, S., and Moon, I.-C. Frequency domain-based dataset distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

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 *International Conference on Machine Learning*, pp. 9206–9216. PMLR, 2020.

Sun, P., Shi, B., Yu, D., and Lin, T. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. *arXiv preprint arXiv:2312.03526*, 2023.

Tommasi, T., Patricia, N., Caputo, B., and Tuytelaars, T. A deeper look at dataset bias. *Domain adaptation in computer vision applications*, pp. 37–55, 2017.

Wang, K., Zhao, B., Peng, X., Zhu, Z., Yang, S., Wang, S., Huang, G., Bilen, H., Wang, X., and You, Y. Cafe: Learning to condense dataset by aligning features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12196–12205, 2022.

Wang, K., Gu, J., Zhou, D., Zhu, Z., Jiang, W., and You, Y. Dim: Distilling dataset into generative model. *arXiv preprint arXiv:2303.04707*, 2023.

Wang, R., Cheng, M., Chen, X., Tang, X., and Hsieh, C.-J. Rethinking architecture selection in differentiable nas. In *International Conference on Learning Representation*, 2021.

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

Wei, X., Cao, A., Yang, F., and Ma, Z. Sparse parameterization for epitomic dataset distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

Wu, X., Deng, Z., and Russakovsky, O. Multimodal dataset distillation for image-text retrieval. *arXiv preprint arXiv:2308.07545*, 2023.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Xiong, Y., Wang, R., Cheng, M., Yu, F., and Hsieh, C.-J. Feddm: Iterative distribution matching for communication-efficient federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16323–16332, 2023.

Yang, E., Shen, L., Wang, Z., Liu, T., and Guo, G. An efficient dataset condensation plugin and its application to continual learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Yang, M. and Kim, B. Benchmarking Attribution Methods with Relative Feature Importance. *CoRR*, abs/1907.09701, 2019.

Yin, Z., Xing, E., and Shen, Z. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. *Advances in Neural Information Processing Systems*, 36, 2024.

Zhang, D. J., Wang, H., Xue, C., Yan, R., Zhang, W., Bai, S., and Shou, M. Z. Dataset condensation via generative model. *arXiv preprint arXiv:2309.07698*, 2023.

Zhang, H., Li, S., Wang, P., Zeng, D., and Ge, S. M3d: Dataset condensation by minimizing maximum mean discrepancy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 9314–9322, 2024a.

Zhang, Y., Zhang, T., Wang, K., Guo, Z., Liang, Y., Bresson, X., Jin, W., and You, Y. Navigating complexity: Toward lossless graph condensation via expanding window matching. *arXiv preprint arXiv:2402.05011*, 2024b.

Zhang, Z. and Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

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 *International Conference on Machine Learning*, pp. 12674–12685. PMLR, 2021b.

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

Zhao, G., Li, G., Qin, Y., and Yu, Y. Improved distribution matching for dataset condensation. 2023.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

Zhou, Y., Nezhadarya, E., and Ba, J. Dataset distillation using neural feature regression. *Advances in Neural Information Processing Systems*, 35:9813–9827, 2022.

A. Appendix

A.1. Extension to Other SOTA methods

Here we show that our method can be extended to other SOTA methods such as DREAM (Liu et al., 2023) and IDM (Zhao et al., 2023) that's based on either DM or DSA. In DREAM, instead of sampling and matching random batches of real images, it applies Kmeans (Lloyd, 1982) algorithm to select representative samples. However, if a dataset is dominated by biased samples, this will cause the bias to be amplified, thus making de-biasing harder. Same for IDM which applies extra operations on top of DM. Both DREAM and IDM shares similar loss with DSA and DM with extra loss to capture the information shared between the condensed dataset and the real dataset. Therefore, our method can naturally extend to these methods as well. See Table 6 for the detailed performance boost.

Table 6. Extend our method to recent SOTA methods DREAM and IDM. Results are run on CMNIST dataset with various bias-conflict ratios.

Method	Bias-conflict Ratio (1.0%)			Bias-conflict Ratio (2.0%)			Bias-conflict Ratio (5.0%)		
	1	10	50	1	10	50	1	10	50
IDM	20.1±0.9	18.8±1.2	17.6±1.6	20.2±1.0	20.1±0.5	19.1±1.3	20.8±0.9	22.3±1.1	26.5±1.5
IDM+Ours	26.1±1.0	54.1±0.3	72.2±0.1	27.8±1.0	35.2±0.1	55.7±0.1	30.9±0.5	40.4±0.4	77.2±0.2
DREAM	21.3±0.4	15.9±1.2	17.9±0.8	26.2±0.5	17.1±1.3	17.5±1.5	23.7±1.2	30.8±1.5	50.4±1.0
DREAM+Ours	27.0±1.2	56.6±0.8	73.8±1.1	27.4±1.1	38.3±1.1	58.5±0.1	40.1±0.5	50.0±1.2	80.1±0.6

A.2. Experimental Setup

Following previous dataset distillation methods (Zhao & Bilen, 2021a; Cazenavette et al., 2022), we use ConvNet as the model architecture. It has 128 filters with kernel size of 3×3. Then it's followed by instance normalization, RELU activation, and an average pooling layer. We use SGD as the optimizer with 0.01 learning rate. For the supervised contrastive model, we use ResNet18 (He et al., 2016b) following (Hwang et al., 2022) with a projection head of 128 dimensions. Same as previous de-biasing works (Nam et al., 2020; Lee et al., 2021; Hwang et al., 2022), we evaluate our results by training a DNN model (same as distillation) on the synthetic dataset and measure its accuracy using the unbiased test set. The results are evaluated with IPC 1, 10 and 50. See Appendix A.2 in Appendix for more experiment details.

To avoid introducing biases during initialization, all synthetic images are initialized with random noise instead of randomly selected real images. For KDE, we fix kernel variance and temperature to be 0.1 across all datasets. All experiments are run on a single 48GB NVIDIA RTX A6000 GPU.

A.3. Extremely low bias-conflict setting

It's possible that in some datasets that bias-conflict samples are extremely rare such as 0.5% used in SelecMix (Hwang et al., 2022). Here we show that our method performs well under this setting as well. See Table 7.

Table 7. Test accuracy under extremely low bias-conflict ratios (0.5%).

Dataset	IPC	MTT	DM	DM+Ours	DSA	DSA+Ours
CMNIST	1	22.9±1.2	25.1±0.9	27.4±0.7	25.9±0.7	26.0±1.0
	10	26.8±1.8	18.8±0.9	40.0±1.2	15.7±0.8	42.0±1.2
	50	20.7±1.0	23.8±0.9	49.0±1.0	11.6±2.0	50.0±1.5

A.4. Experiment results on the original unbiased dataset

Here we show the experiment results of our method on the original unbiased dataset in Table 8.

Table 8. Performance Comparison on original Datasets

Dataset	Method	IPC 1	IPC 10	IPC 50
CMNIST	DM	87.3 ± 0.4	95.6 ± 0.5	97.2 ± 0.7
	DM+Ours	86.0 ± 0.5	95.8 ± 0.8	97.2 ± 1.0
FashionMNIST	DM	72.5 ± 0.5	83.1 ± 0.4	86.2 ± 1.0
	DM+Ours	70.4 ± 0.9	82.4 ± 0.7	84.5 ± 1.2
CIFAR-10	DM	27.2 ± 0.3	49.5 ± 0.6	60.2 ± 0.8
	DM+Ours	27.1 ± 0.2	49.0 ± 0.8	60.8 ± 0.9

A.5. Datasets

We conduct experiments on 3 datasets, two of which are widely used in SOTA de-biasing methods to assess color and noise biases, while also introducing a novel dataset to evaluate background biases.

Colored MNIST (CMNIST): (Nam et al., 2020) introduces the Colored MNIST dataset by injecting color with random perturbation into the MNIST dataset (Deng, 2012). Each digit will be associated with a specific color as its bias such as digit 0 being red and digit 4 being green. We evaluate our method under 3 bias conflicting ratios with 1% (54,509 bias aligned, 491 bias conflicting), 2% (54,014 bias aligned, 986 bias conflicting) and 5% (52,551 bias aligned, 2,449 bias conflicting).

Corrupted CIFAR-10: Generated from the regular CIFAR-10 dataset (Krizhevsky et al., 2009), Corrupted CIFAR-10 applies different corruptions (Hendrycks & Dietterich, 2019) to the images in CIFAR-10 so that images from one class are associated with one type of corruption such as GaussianNoise or MotionBlur. We also evaluate our method under 3 bias conflicting ratios with 1% (44,527 bias aligned, 442 bias conflicting), 2% (44,145 bias aligned, 887 bias conflicting) and 5% (42,820 bias aligned, 2,242 bias conflicting).

Background Fashion-MNIST (BG FMNIST): Background bias, which results in an over-reliance on the background for predicting foreground objects, has been employed to assess interpretability methods in various prior studies (Yang & Kim, 2019; Zhou et al., 2017). Following this idea, we construct a new dataset biased in backgrounds by using Fashion-MNIST (Xiao et al., 2017) as foregrounds which include a training set of 60,000 examples and a test set of 10,000 examples. And MiniPlaces (Zhou et al., 2017) is used as backgrounds to introduce background biases such as T-shirt is associated with bamboo forest background, trouser is associated with livingroom background, etc. Similar to other biased datasets, we also conduct experiments in 3 settings including 1%, 2% and 5% bias conflicting samples.

BFFHQ: It is introduced in (Lee et al., 2021) by curating a biased dataset from the FFHQ dataset (Karras et al., 2019). In this dataset, they focus on age and gender as key facial attributes, with age being the intrinsic attribute and gender as the bias attribute. The dataset is curated to exhibit a strong correlation between these two attributes. The dataset predominantly features females categorized as 'young' (aged between 10 to 29 years) and males as 'old' (aged between 40 to 59 years). Consequently, the majority of the samples in the dataset are composed of young women and old men, aligning with the intended bias. The dataset only comes with 0.5% bias-conflict samples. For more detailed info regarding the dataset, please refer to (Lee et al., 2021).

A.6. Matching Training Trajectories

The objective function of MTT can be described as below

$$\mathcal{L} = \|\hat{\theta}_{t+T} - \theta^*_{t+M}\|_2^2 / \|\theta^*_t - \theta^*_{t+M}\|_2^2. \tag{5}$$

Where T is the number of synthetic data training steps and M is the expert model training steps using the whole dataset. As MTT tries to match the training trajectory of expert models and models trained using synthetic data, the real dataset is not needed during matching phase. Therefore, our reweighting scheme doesn't work directly in MTT. However, as shown in Table 4, we should still be able to mitigate bias for MTT through fixing the expert models. We leave this to future work.

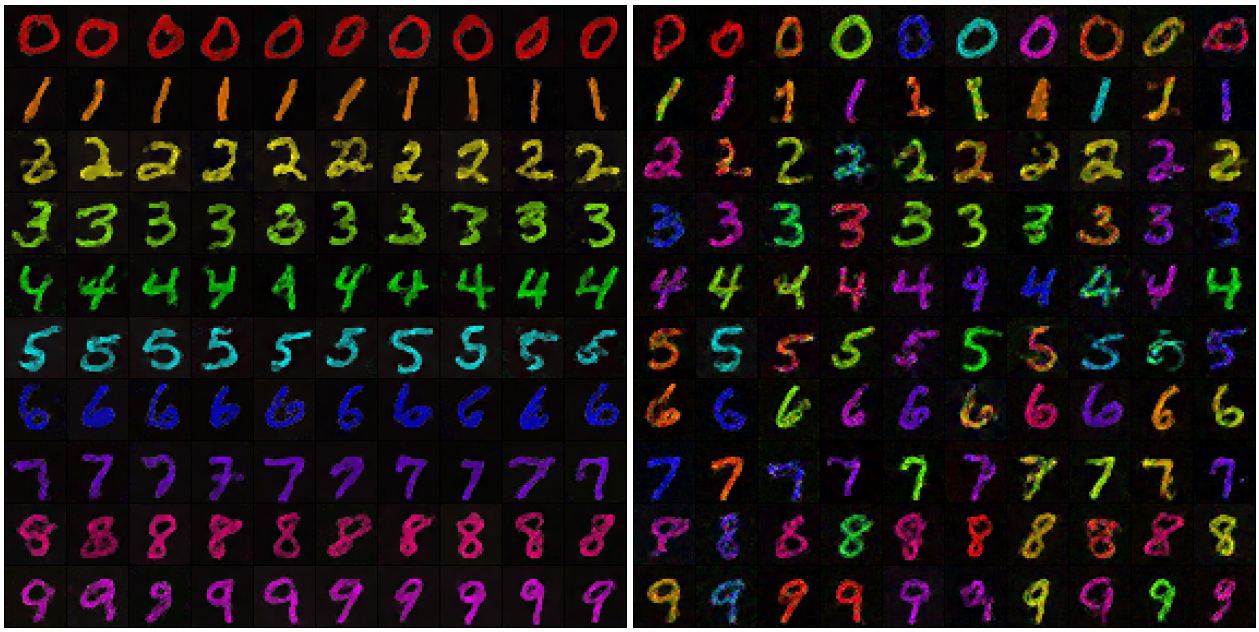

Figure 4. Synthetic images from vanilla DM (left) vs Ours (right) distilled from CMNIST with 5% bias-conflict samples. The one synthesized by vanilla DM is dominated by the bias feature while ours includes a rich set of features from both biased and bias-conflict samples.

A.7. Qualitative Analysis

Here we visualize the synthetic datasets produced by vanilla DM and DM+Ours on CMNIST with 5% bias-conflicting samples and IPC 10 in Figure 4. As shown, images synthesized by the vanilla DM completely ignores the unbiased samples due to the reasons explained in Section 4.1, causing the bias to be even more amplified than the original dataset (the original dataset has 5% unbiased samples such as green 0s or yellow 1s, the distilled dataset has 0%). When combined with our method, DM is able to identify and synthesize unbiased samples into the final synthetic dataset. Similar results can also be seen with the vanilla DSA and DSA+Ours in Appendix Appendix A.14.

A.8. Visualization of KDE

See Figure Figure 5 for a visualization of KDE applied on a normal distribution. The dotted line is a true normal distribution. The histogram represents the observed data points and the red curve shows the density function estimated using KDE.

A.9. Ablation study on cutting-off score

Here we conduct an ablation study on the cutoff score used when computing the sample weights, the results are shown in Figure 6. It can be seen that choosing a cutoff score that's too large or too small will hinder the de-biasing performance. Also our algorithm is not very sensitive to cutoff scores and the optimal performances can be achived with a wide range of cutoff scores.

A.10. Implementation details of applying de-biasing methods to synthetic dataset

As mentioned before, one natural idea is to directly apply de-biasing methods on distilled synthetic dataset. Here we describe the implementation details. For the results of DM, it's acquired by directly running the vanilla DM on CMNIST dataset. For DM+DFA, we apply the DFA de-biasing methods on the distilled synthetic dataset from the original DM. For DM+SelecMix, we test the two methods proposed in (Hwang et al., 2022) which corresponds to directly applying SelecMix methods and appling it on top of LfF (Nam et al., 2020).

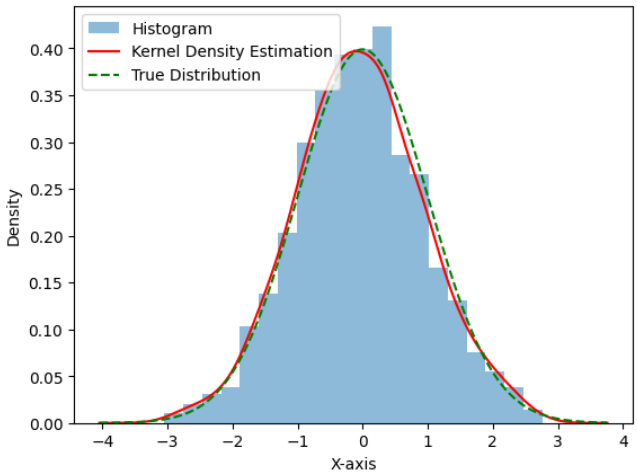

Figure 5. KDE applied on a normal distribution.

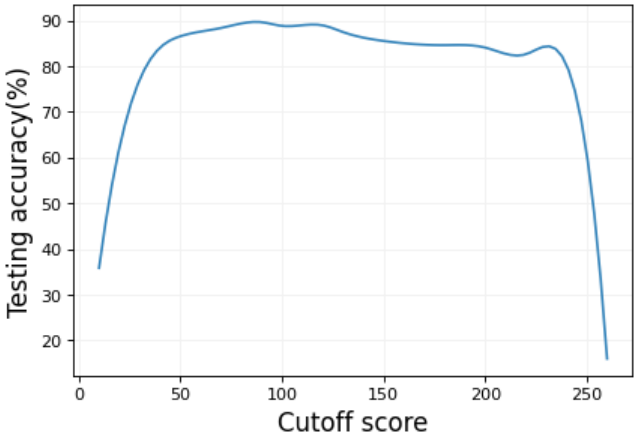

Figure 6. Ablation study on cutoff score. The experiment is conducted on CMNIST with 5% bias-conflicting samples and IPC 10.

Figure 7. Examples of original images (top) and synthetic images (bottom) from biased dataset.

A.11. Implementation details of applying de-biasing method to surrogate model

For applying de-biasing methods to surrogate models, we mainly test two combinations. The first one is DM plus DFA. The reason this can potentially work is that DM tries to match the distribution of real data and synthetic data in the embedding space. DFA is a perfect fit for DM because DFA tries to separate the embeddings into the intrinsic parts and the bias parts. DM can choose to match only the intrinsic parts, thus getting rid of biases. For MTT, since it doesn't rely on real data during matching phase but the expert training trajectories. Therefore, the best way to mitigate bias in MTT synthesized datasets is to debias the expert training trajectories. Thus we choose to apply the most recent SOTA model de-biasing method to acquire the expert trajectories first. Then we have MTT match these de-biased trajectories.

A.12. Computation time for KDE

Similar to (Hwang et al., 2022), computing the embedding distance between samples adds extra computational cost. This applies to our method as well. For distribution based method such as DM, since it doesn't compute second order gradients, the whole distillation process is very fast. The time used to compute 10 matching iterations on CMNIST for the vanilla DM is 1.0 second while the time used to compute 10 matching iterations of DM+KDE is 1.9 seconds. However, note that DM usually takes less than 10,000 iterations to converge which is around 16.7 minutes. Compared to other dataset distillation based methods such as DSA that takes hours to finish on the same dataset, DM+KDE can converge quickly even with the added computation. For DSA, sicne it computes second order gradients, it takes around 55 seconds for each 10 iterations. With KDE, it takes 65 seconds. Therefore, computing KDE adds around 18.2% computation time. We usually observe significant performance improvement at the early training epochs when combining our method with the vanilla method. Note that generating the Supervised Contrastive model is fast, e.g. it takes less than 30 minutes to train it on CMNIST. All the data are measured on the same hardware as our experiments.

The computation of KDE can be greatly reduced by caching or grouping the data points. Based on our findings mentioned in the main paper, we observe that most of the biased samples tend to be close to each other in the SupCon embedding space and most of the unbiased samples are far away from this majority group. Therefore, there is no need to compute their distance within these biased samples and their distance with unbiased samples repeatedly. Thus, by pre-computing an estimated grouping information and reuse the intra-group and inter-group distance, the whole KDE process can be greatly reduced. We leave this to our future work.

A.13. More qualitative results

First of all, we show the results under IPC 1 in Appendix A.13. It can be seen that images synthesized by the vanilla DM is dominated by the biased samples such as red 0s and green 4s. On the contrary, the ones produced by our method is a fusion of both biased samples and unbiased samples, thus mitigating the biases in synthetic datasets. Similar results can also be seen from gradient based method in Appendix A.13 where the single image synthesized by DSA is also a fusion of both biased and unbiased samples.

A.14. Example Synthetic Images

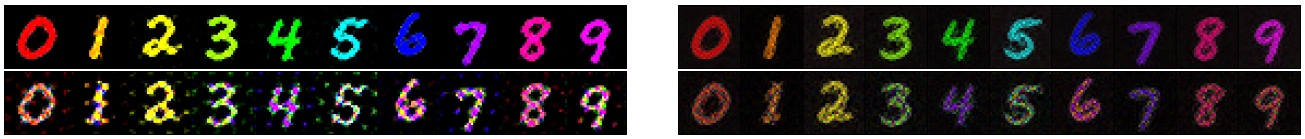

Figure 8. Synthetic images from original DM (top left) vs DM+Ours (bottom left) and original DSA (top right) vs DSA+Ours (bottom right). Results are generated from CMNIST with 5% bias-conflicting samples and IPC 1.

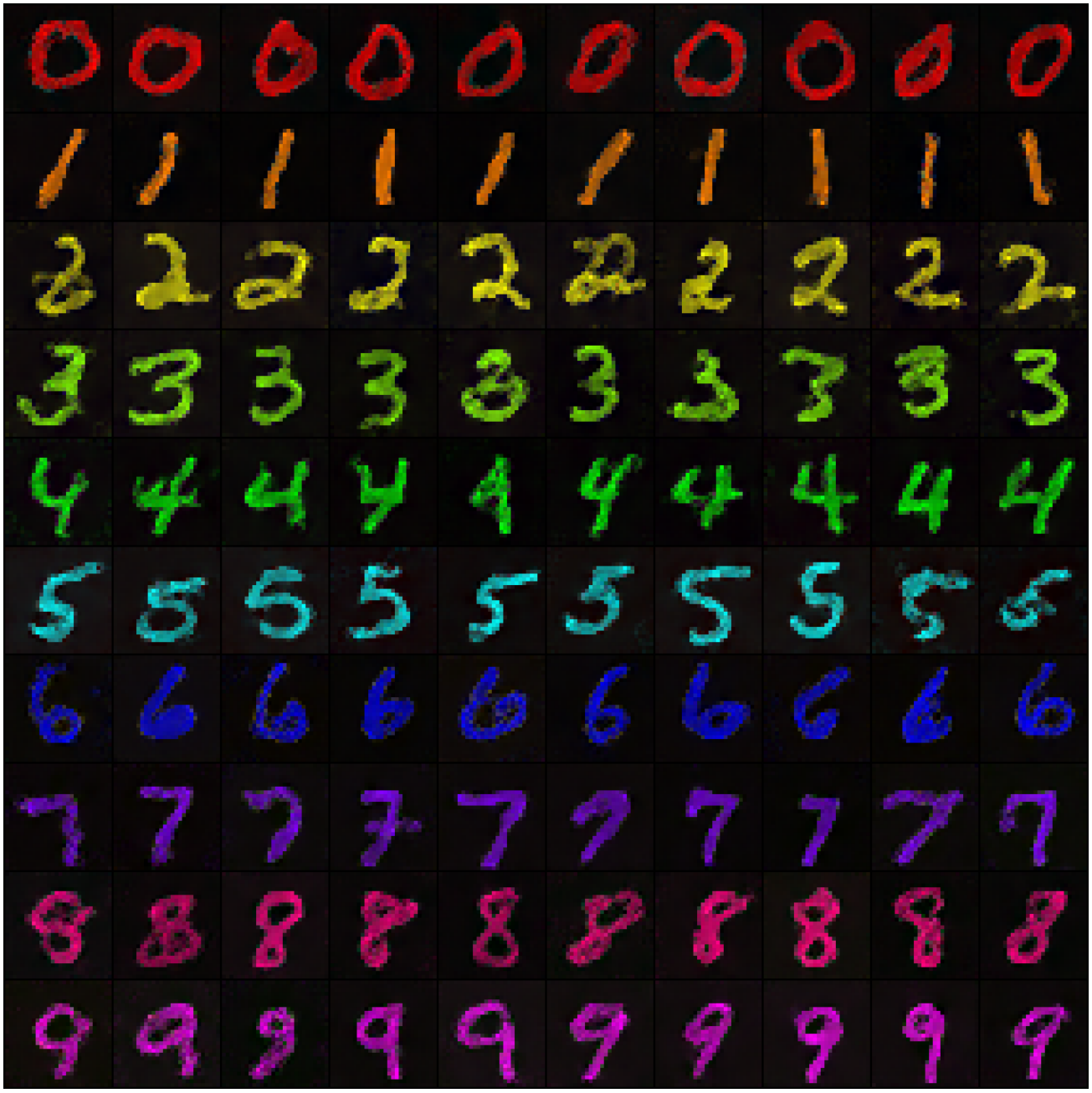

Figure 9. Synthesized dataset by DM on CMNIST with 5% bias-conflicting samples.

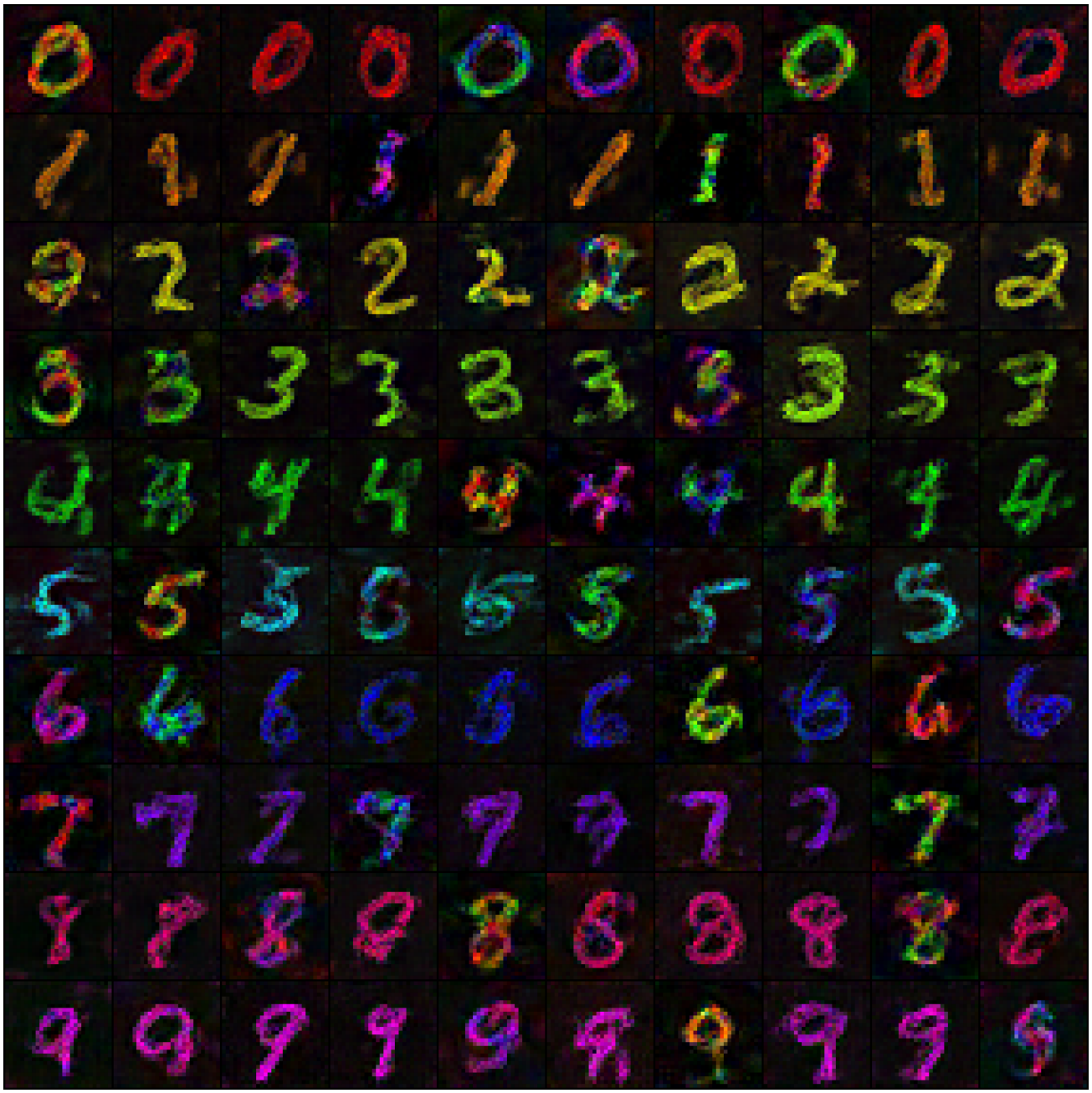

Figure 10. Synthesized dataset by DSA on CMNIST with 5% bias-conflicting samples.

Figure 11. Synthesized dataset by MTT on CMNIST with 5% bias-conflicting samples.

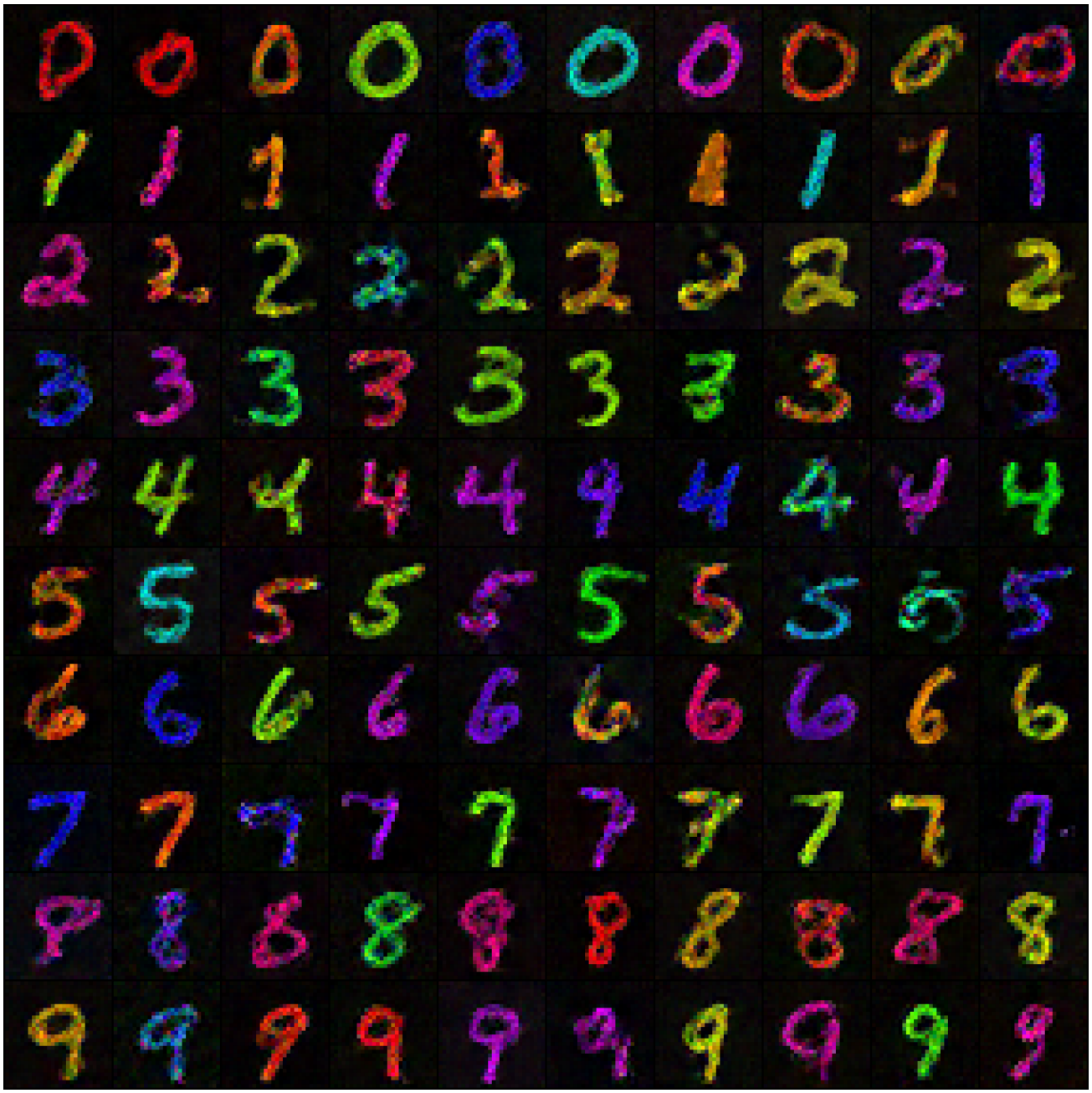

Figure 12. Synthesized dataset by DM+Ours on CMNIST with 5% bias-conflicting samples.

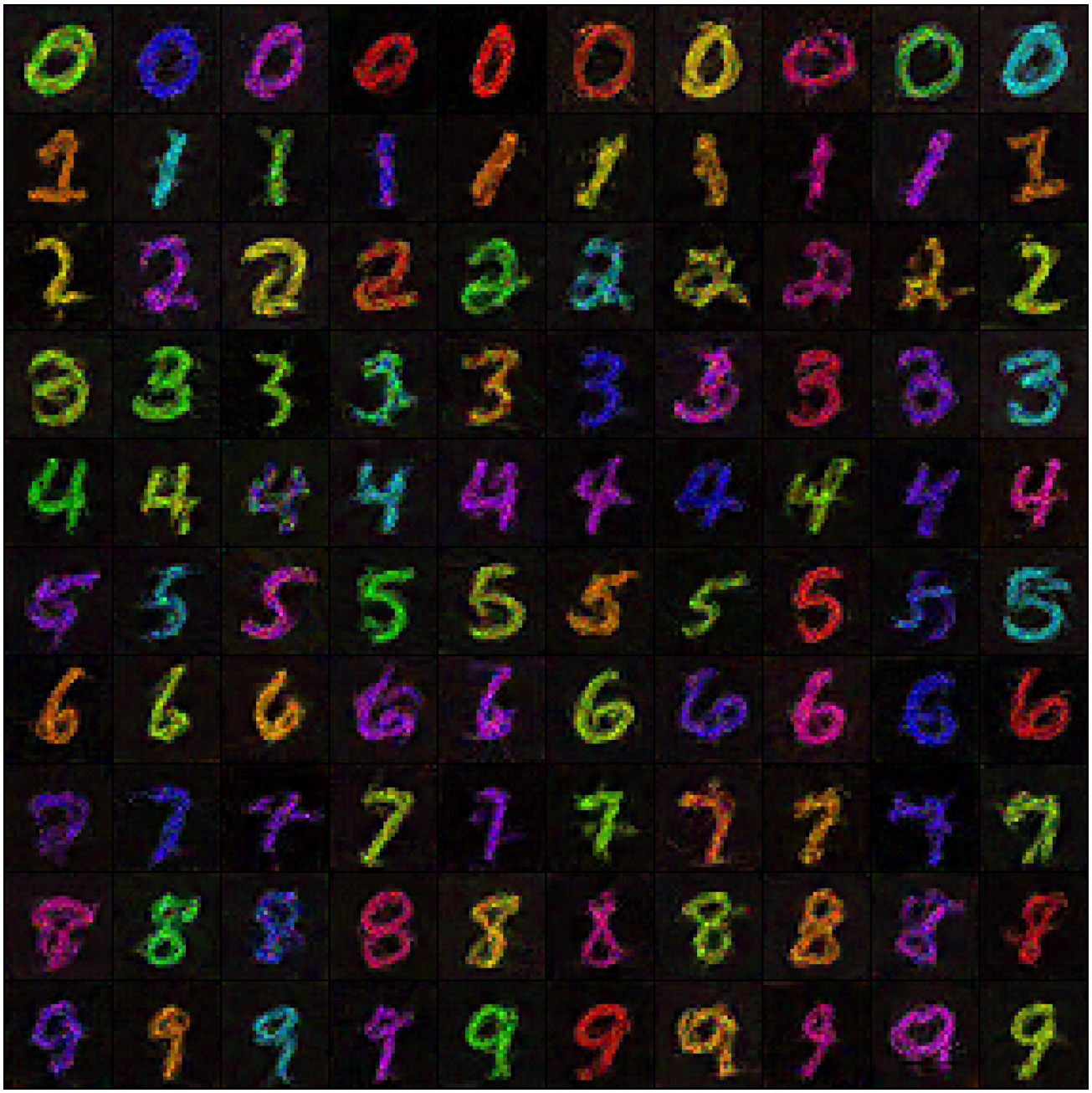

Figure 13. Synthesized dataset by DSA+Ours on CMNIST with 5% bias-conflicting samples.

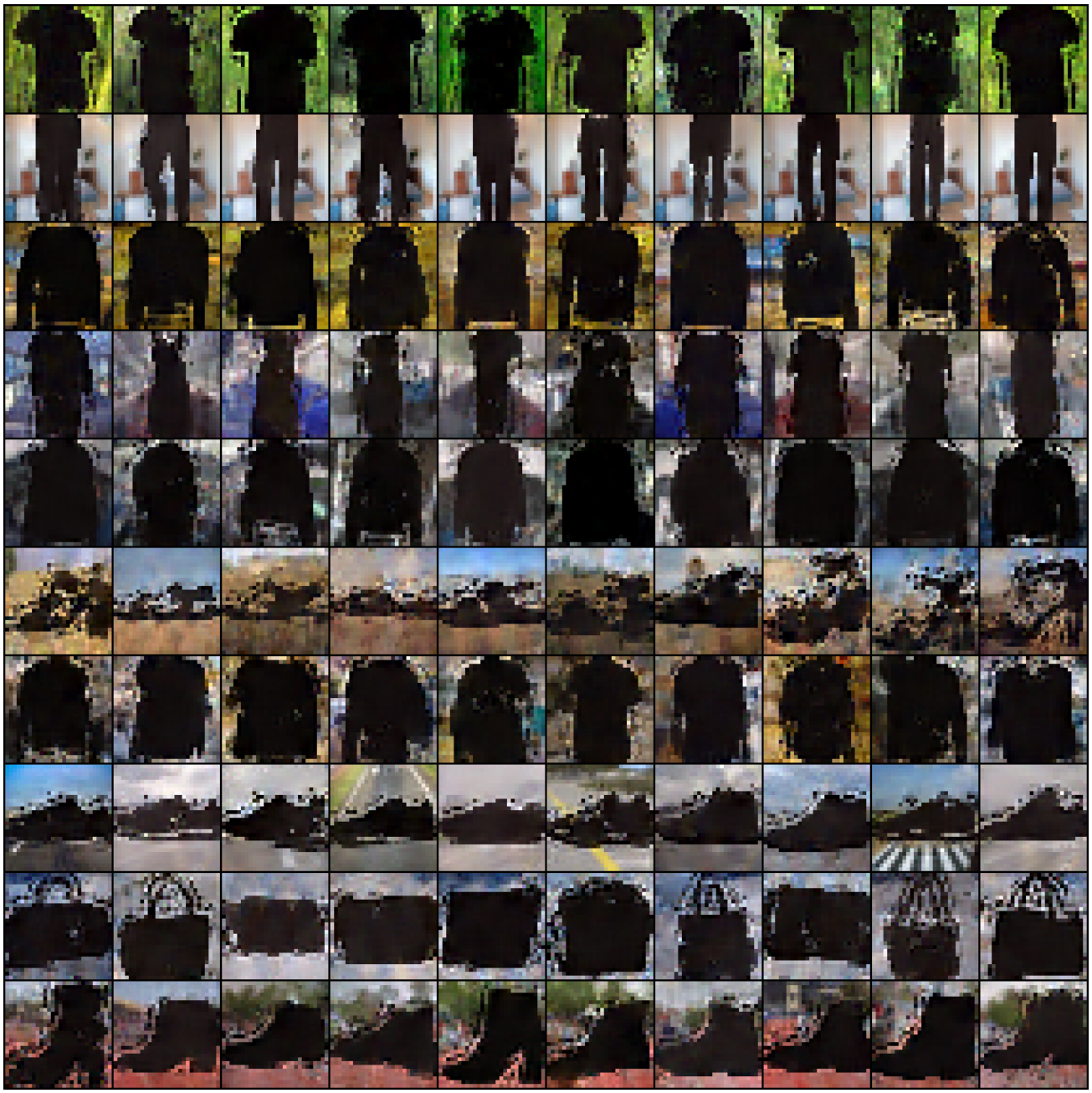

Figure 14. Synthesized dataset by DM on BG FMNIST with 5% bias-conflicting samples.

Figure 15. Synthesized dataset by DSA on BG FMNIST with 5% bias-conflicting samples.

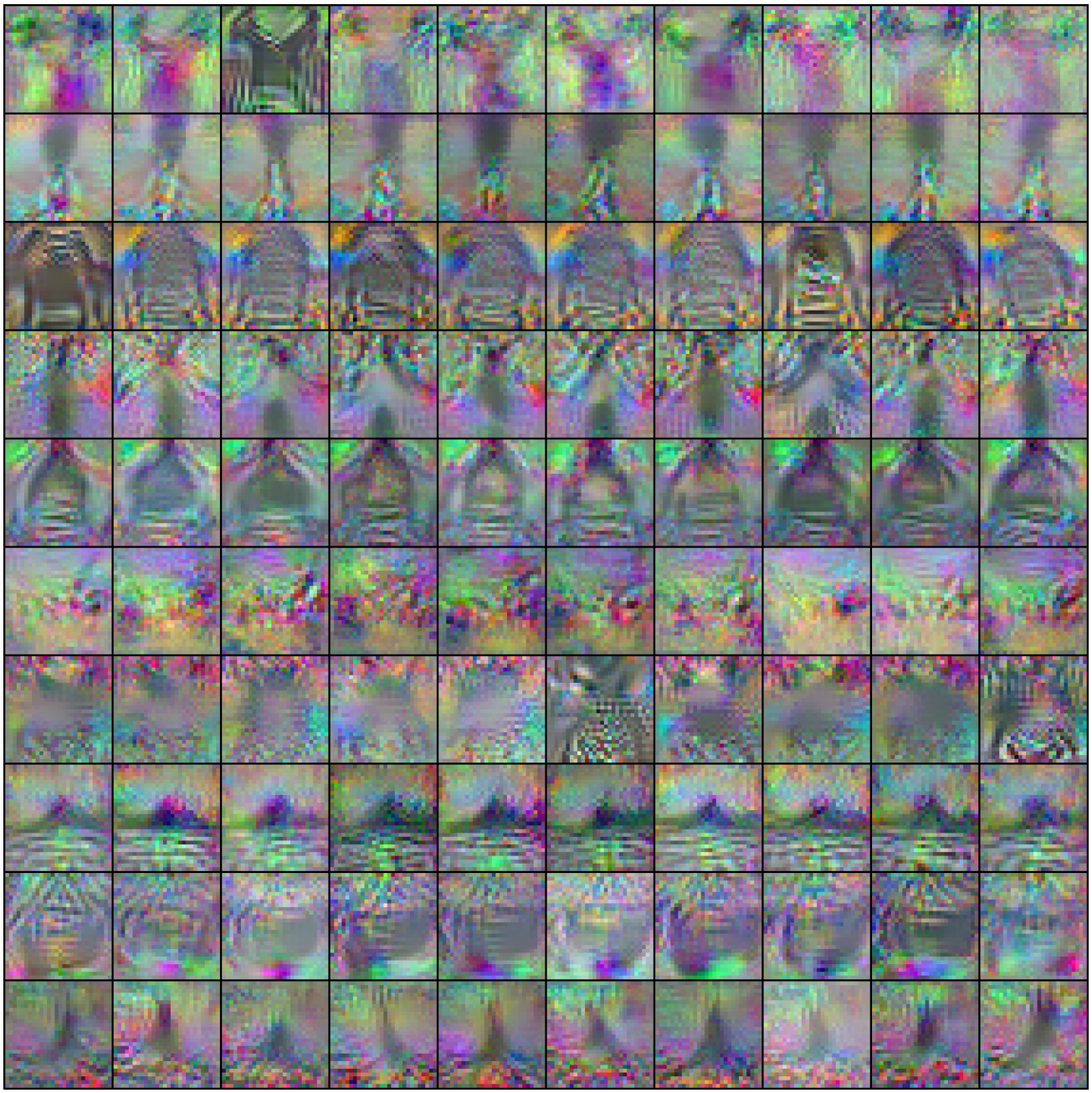

Figure 16. Synthesized dataset by MTT on BG FMNIST with 5% bias-conflicting samples.

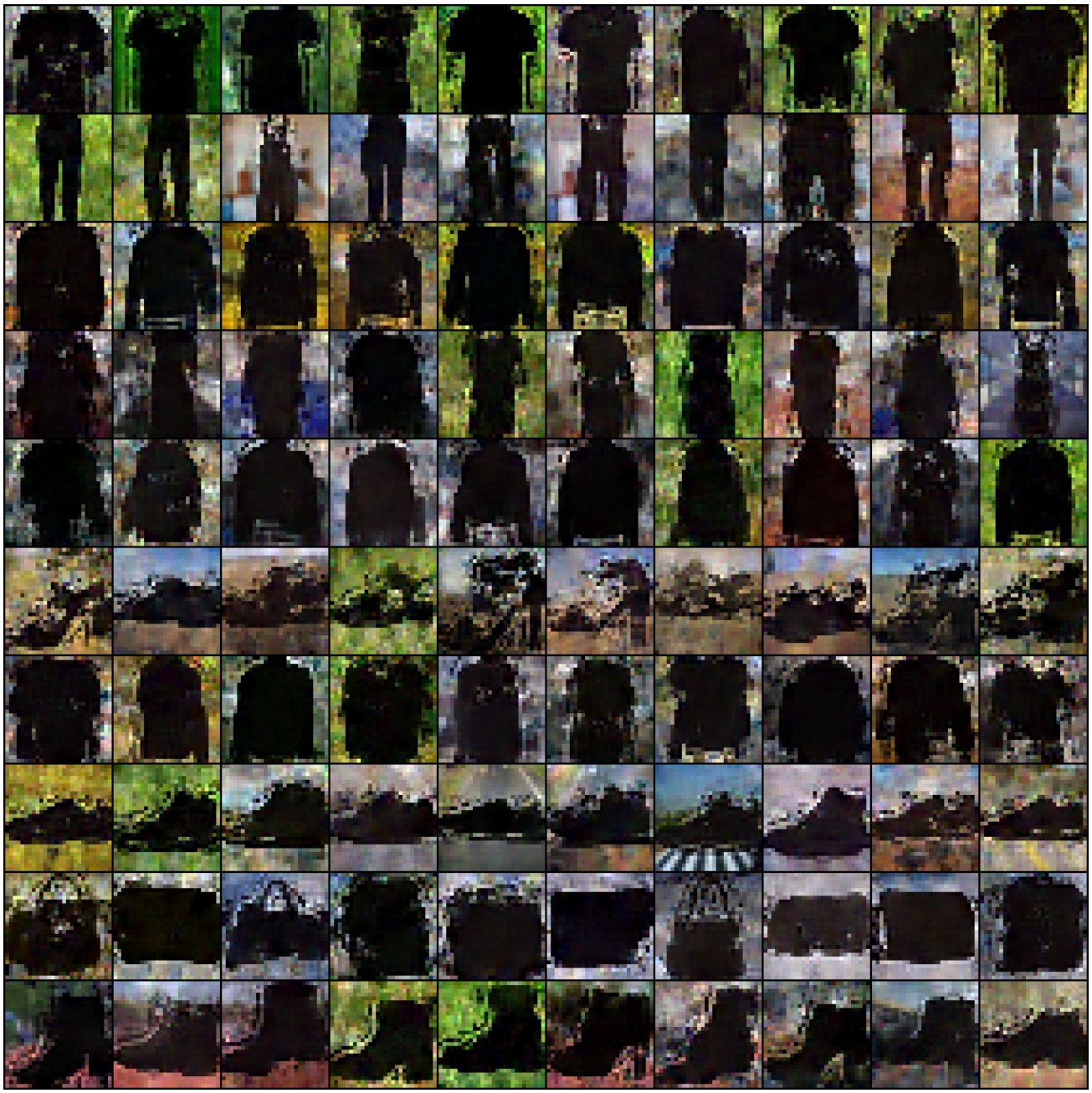

Figure 17. Synthesized dataset by DM+Ours on BG FMNIST with 5% bias-conflicting samples.

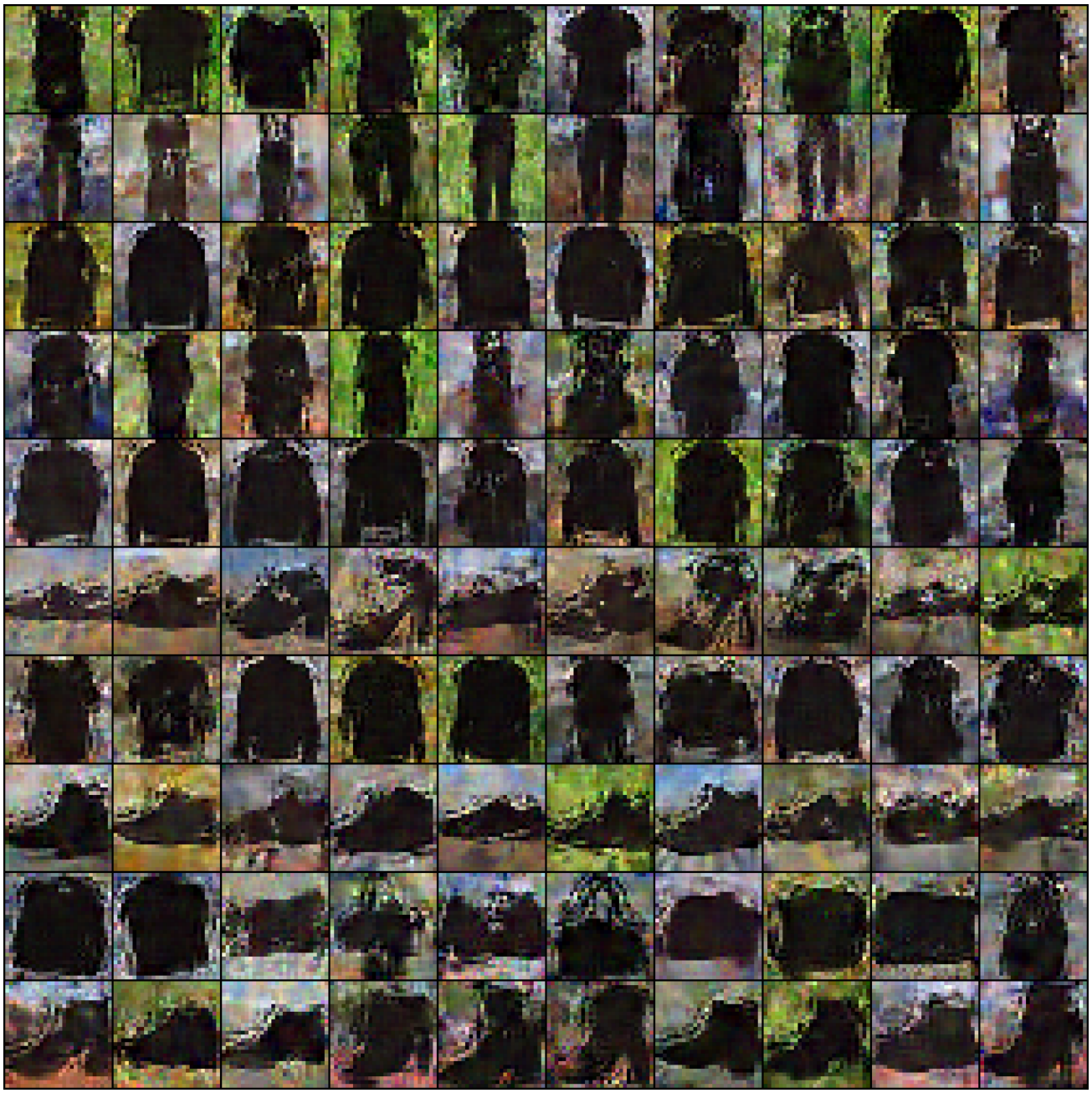

Figure 18. Synthesized dataset by DSA+Ours on BG FMNIST with 5% bias-conflicting samples.