Privacy for Free: How does Dataset Condensation Help Privacy?

Tian Dong [1 *] **Bo Zhao** [2] **Lingjuan Lyu** [3]

Abstract

To prevent unintentional data leakage, research community has resorted to data generators that can produce differentially private data for model training. However, for the sake of the data privacy, existing solutions suffer from either expensive training cost or poor generalization performance. Therefore, we raise the question whether training efficiency and privacy can be achieved simultaneously. In this work, we for the first time identify that dataset condensation (DC) which is originally designed for improving training efficiency is also a better solution to replace the traditional data generators for private data generation, thus providing privacy for free. To demonstrate the privacy benefit of DC, we build a connection between DC and differential privacy, and theoretically prove on linear feature extractors (and then extended to non-linear feature extractors) that the existence of one sample has limited impact ($O(m/n)$) on the parameter distribution of networks trained on m samples synthesized from $n(n \gg m)$ raw samples by DC. We also empirically validate the visual privacy and membership privacy of DC-synthesized data by launching both the loss-based and the state-of-the-art likelihood-based membership inference attacks. We envision this work as a milestone for data-efficient and privacy-preserving machine learning.

1. Introduction

Machine learning models are notoriously known to suffer from a wide range of privacy attacks (Lyu et al., 2020), such as model inversion attack (Fredrikson et al., 2015), membership inference attack (MIA) (Shokri et al., 2017), property inference attack (Melis et al., 2019), etc. The numerous concerns on data privacy make it impractical for data curators to directly distribute their private data for purpose of interest. Previously, generative models, e.g., generative adversarial networks (GANs) (Goodfellow et al., 2014), was supposed to be an alternative of data sharing. Unfortunately, the aforementioned privacy risks exist not only in training with raw data but also in training with synthetic data produced by generative models (Chen et al., 2020b). For example, it is easy to match the fake facial images synthesized by GANs with the real training samples from the same identity (Webster et al., 2021). To counter this issue, existing efforts (Xie et al., 2018; Wang et al., 2021; Cao et al., 2021; Harder et al., 2021) applied differential privacy (DP) (Dwork et al., 2006) to develop differentially private data generators (called DP-generators), because DP is the *de facto* privacy standard which provides theoretical guarantees of privacy leakage. Data produced by DP-generators can then be applied to various downstream tasks, e.g., data analysis, visualization, training privacy-preserving classifier, etc.

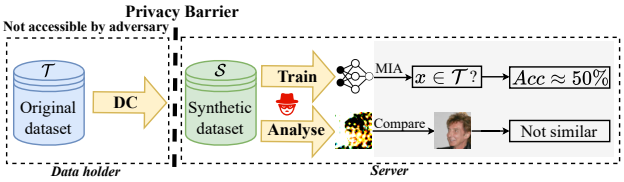

Figure 1. DC-synthesized data can be used for privacy-preserving model training and cannot be recovered through MIA and visual comparison analysis.

However, due to the noise introduced by DP, the data produced by DP-generators are of low quality, which impedes the utility as training data, i.e., accuracy of the models trained on these data. Thus, more data generated by DP-generators are needed to obtain good generalization performance, which inevitably decreases the training efficiency.

Recently, the research of dataset condensation (DC) (Wang et al., 2018; Sucholutsky & Schonlau, 2019; Such et al., 2020; Bohdal et al., 2020; Zhao et al., 2021; Zhao & Bilen, 2021b;a; Nguyen et al., 2021a;b; Jin et al., 2022; Cazenavette et al., 2022; Wang et al., 2022) emerges, which aims to condense a large training set into a small synthetic set that is comparable to the original one in terms of training deep neural networks (DNNs). Different from traditional

[*]Work done during internship at Sony AI. [1]Department of Computer Science and Engineering, Shanghai Jiao Tong University [2]School of Informatics, The University of Edinburgh [3]Sony AI. Correspondence to: Lingjuan Lyu <Lingjuan.Lv@sony.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

generative models that are trained to generate real-looking samples with high fidelity, these DC methods generate informative training samples for data-efficient learning. In this work, we for the first time investigate the feasibility of protecting data privacy using DC techniques. We find that DC can not only accelerate model training but also offer privacy for free. Figure 1 illustrates how DC methods can be applied to protect membership privacy and visual privacy. Specifically, we first analyse the relationship between DC-synthesized data and original ones (Proposition 4.3 and 4.4), and theoretically prove on linear DC extractors that the change caused by removing or adding one element in n raw samples to the parameter distribution of models trained on $m(m \ll n)$ DC-synthesized samples (i.e., privacy loss) is bounded by $O(m/n)$ (Proposition 4.10), which satisfies that one element does not greatly change the model parameter distribution (the concept of DP). The conclusions are further analytically and empirically generalized to non-linear feature extractors. Then, we empirically validate that models trained on DC-synthesized data are robust to both vanilla loss-based MIA and the state-of-the-art likelihood-based MIA (Carlini et al., 2022). Finally, we study the visual privacy of DC-synthesized data in case of adversary's direct matching attack. All the results show that DC-synthesized data are not perceptually similar to the original data as our Proposition 4.4 indicates, and cannot be reversed to the original data through similarity metrics (e.g., LPIPS).

Through empirical evaluations on image datasets, we validate that DC-synthesized data can preserve both *data efficiency* and *membership privacy* when being used for model training. For example, on FashionMNIST, DC-synthesized data enable models to achieve a test accuracy of at least 33.4% higher than that achieved by DP-generators under the same empirical privacy budget. Meanwhile, to achieve a test accuracy of the same level, DC only needs to synthesize at most 50% data of the size required by GAN-based methods, which speeds up the training by at least 2 times.

In summary, our contributions are three-fold:

- To the best of our knowledge, we are the first to introduce the emerging dataset condensation techniques into privacy community and provide systematical audit on state-of-the-art DC methods.
- We build the connection between dataset condensation and differential privacy, and contribute theoretical analysis with both linear and non-linear feature extractors.
- Extensive experiments on image datasets empirically validate that DC methods reduce the adversary advantage of membership privacy to zero, and DC-synthesized data are perceptually irreversible to original data in terms of similarity metrics of L_2 and LPIPS.

2. Background and Related Work

In this section, we briefly present dataset condensation and the membership privacy issues in machine learning models.

2.1. Dataset Condensation

Orthogonal to model knowledge distillation (Hinton et al., 2015), Wang et al. firstly proposed dataset distillation (DD) which aims to distill knowledge from a large training set into a small synthetic set. The synthetic set can be used to efficiently train deep neural networks with a moderate decrease of testing accuracy or anonymize sensitive images (Li et al., 2020). Recent works significantly advanced this research area by proposing Dataset Condensation (DC) with gradient matching (Zhao et al., 2021; Zhao & Bilen, 2021b), Distribution Matching (DM) (Zhao & Bilen, 2021a) and introducing Kernel Inducing Points (KIP) (Nguyen et al., 2021a;b). For example, the synthetic sets (50 images per class) generated by DM can be used to train a 3-layer convolutional neural networks from scratch and obtain over 60% testing accuracies on CIFAR10 (Krizhevsky et al., 2009) and over 98% testing accuracies on MNIST (LeCun et al., 1998). In this work, we mainly focus on synthetic sets generated by DSA (Zhao & Bilen, 2021b), DM (Zhao & Bilen, 2021a) and KIP (Nguyen et al., 2021a), because 1) DSA and DM are improved DC and KIP is improved DD, and 2) the performance of DD and DC are significantly lower than DSA, DM and KIP.

We formulate dataset condensation problem using the symbols presented in (Zhao & Bilen, 2021a). Given a large-scale dataset (target dataset) $\mathcal{T} = \{(\boldsymbol{x}_i, y_i)\}$ which consists of $|\mathcal{T}|$ samples from C classes, the objective of dataset condensation (or distillation) is to learn a synthetic set $\mathcal{S} = \{(\boldsymbol{s}_i, y_i)\}$ with $|\mathcal{S}|$ synthetic samples so that the deep neural networks can be trained on $\mathcal{S}$ and achieve comparable testing performance to those trained on $\mathcal{T}$:

$$\mathbb{E}_{\boldsymbol{x}\sim P_{\mathcal{D}}}[\mathcal{L}(\phi_{\boldsymbol{\theta}^{\mathcal{T}}}(\boldsymbol{x}), y)] \simeq \mathbb{E}_{\boldsymbol{x}\sim P_{\mathcal{D}}}[\mathcal{L}(\phi_{\boldsymbol{\theta}^{\mathcal{S}}}(\boldsymbol{x}), y)], \quad (1)$$

where $P_{\mathcal{D}}$ is the real data distribution, $\phi_{\boldsymbol{\theta}^{\mathcal{T}}}(\cdot)$ and $\phi_{\boldsymbol{\theta}^{\mathcal{S}}}(\cdot)$ are models trained on $\mathcal{T}$ and $\mathcal{S}$ respectively. $\mathcal{L}(\cdot, \cdot)$ is the loss function, e.g. cross-entropy loss.

To achieve this goal, Wang et al. proposed a meta-learning based method which parameterizes the model updated on synthetic set as $\boldsymbol{\theta}^{\mathcal{S}}(\mathcal{S})$ and then learns the synthetic data by minimizing the validation loss on original training data $\mathcal{T}$:

$$\arg\min_{\mathcal{S}} \mathcal{L}^{\mathcal{T}}(\boldsymbol{\theta}^{\mathcal{S}}(\mathcal{S})), \quad (2)$$

where $\boldsymbol{\theta}^{\mathcal{S}}(\mathcal{S}) = \arg\min_{\boldsymbol{\theta}} \mathcal{L}^{\mathcal{S}}(\boldsymbol{\theta})$. The meta-learning algorithm has to recurrently unroll the computation graph $\boldsymbol{\theta}^{\mathcal{S}}$ with respect to $\mathcal{S}$, which is expensive and unscalable. (Nguyen et al., 2021a) proposed Kernel Inducing Points

(KIP) which leverages the neural tangent kernel (NTK) (Jacot et al., 2018) to replace the expensive network parameter updating. With NTK, $\boldsymbol{\theta}^{\mathcal{S}}$ has a closed-form solution. Thus, KIP learns synthetic data by minimizing the kernel ridge regression loss:

$$\arg\min_{X_s} \frac{1}{2} \left\| y_t - K_{X_t X_s}(K_{X_s X_s} + \lambda I)^{-1} y_s \right\|^2, \quad (3)$$

where X_s and X_t are the synthetic and real images from $\mathcal{S}$ and $\mathcal{T}$, y_s and y_t are corresponding labels. K_{UV} represents the NTK matrix $(K(u,v))_{(u,v)\in U,V}$ for two sets U and V. For a neural network $\phi_{\boldsymbol{\theta}}$, the definition of $K(u,v)$ on elements u and v is $K(u,v) = \nabla_{\boldsymbol{\theta}}\phi_{\boldsymbol{\theta}}(u) \cdot \nabla_{\boldsymbol{\theta}}\phi_{\boldsymbol{\theta}}(v)$.

Zhao et al. proposed a novel DC framework to condense the real dataset into a small synthetic set by matching the gradients when inputting real and synthetic batches into the same model, which can be expressed as follows:

$$\arg\min_{\mathcal{S}} \mathrm{E}_{\boldsymbol{\theta}_0 \sim P_{\boldsymbol{\theta}_0}} [\sum_{t=0}^{T-1} D(\nabla_{\boldsymbol{\theta}}\mathcal{L}^{\mathcal{S}}(\boldsymbol{\theta}_t), \nabla_{\boldsymbol{\theta}}\mathcal{L}^{\mathcal{T}}(\boldsymbol{\theta}_t))], \quad (4)$$

where model $\boldsymbol{\theta}_t$ is updated by minimizing the loss $\mathcal{L}^{\mathcal{S}}(\boldsymbol{\theta}_t)$ alternatively, D computes distance between gradients. (Zhao & Bilen, 2021b) enabled the learned synthetic images to be effectively used to train neural networks with data augmentation by introducing the differentiable Siamese augmentation (DSA) $\mathcal{A}_\omega(\cdot)$ and improved the matching loss in (4) as follows:

$$D(\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_t, \mathcal{A}_\omega(\mathcal{S})), \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_t, \mathcal{A}_\omega(\mathcal{T}))). \quad (5)$$

Although (Zhao et al., 2021) successfully avoided unrolling the recurrent computation graph in (Wang et al., 2018), it still needs to compute the expensive bi-level optimization and second-order derivative. To further simplify the learning of synthetic data, (Zhao & Bilen, 2021a) proposed a simple yet effective dataset condensation method with distribution matching (DM). Specifically, the learned synthetic data $\mathcal{S}$ should have data distribution close to that of real data $\mathcal{T}$ in randomly sampled embedding spaces:

$$\begin{aligned} \min_{\mathcal{S}} \mathbb{E}_{\boldsymbol{\vartheta}\sim P_{\boldsymbol{\vartheta}}, \omega\sim\Omega} \| \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \psi_{\boldsymbol{\vartheta}}(\mathcal{A}(\boldsymbol{x}_i, \omega)) \\ - \frac{1}{|\mathcal{S}|} \sum_{j=1}^{|\mathcal{S}|} \psi_{\boldsymbol{\vartheta}}(\mathcal{A}(\boldsymbol{s}_j, \omega)) \|^2, \end{aligned} \quad (6)$$

where $\psi_{\boldsymbol{\vartheta}}$ represents randomly sampled embedding functions (namely feature extractors), e.g. randomly initialized neural networks and $\mathcal{A}(\cdot, \omega)$ is the differentiable Siamese augmentation. Experimental results show that this simple objective can lead to effective synthetic data that are comparable even better than those generated by existing methods.

2.2. Membership Privacy

For privacy analysis, we mainly focus on membership privacy as it directly relates to personal privacy. For example, inferring that an individual's facial image was in a shop's training dataset reveals the individual had visited the shop. Shokri et al. have shown that DNNs' output can leak the membership privacy of the input (i.e., whether the input belongs to the training dataset) under *membership inference attack* (MIA). In general, MIA only needs black-box access to model parameters (Sablayrolles et al., 2019) and can be successful with logit (Yu et al., 2021) or hard label prediction (Li & Zhang, 2021; Choquette-Choo et al., 2021).

Loss-based MIA. The loss-based MIA infers membership by the predicted loss: if the loss is lower than a threshold τ, then the input is a member of the training data. Formally, the membership $M(\mathbf{x})$ of an input $\mathbf{x}$ can be expressed as:

$$M(\mathbf{x}) = \mathbb{1}(l(\mathbf{x}) \leq \tau), \quad (7)$$

where $M(\mathbf{x}) = 1$ means $\mathbf{x}$ is a training member, $\mathbb{1}(A) = 1$ if event A is true. The threshold τ can be either chosen by locally trained shadow models (Shokri et al., 2017) or via optimal bayesian strategy (Sablayrolles et al., 2019).

Likelihood-based MIA. Recent works (Carlini et al., 2022; Rezaei & Liu, 2021) pointed out that the evaluation of MIA should include False Postive Rate (FPR) instead of averaged metrics (e.g., attack accuracy, Area Under Curve (AUC) score of Receiver Operating Characteristic (ROC) curve), because MIA is a real threat only if the FPR is low (i.e., few data are inferred as member). Moreover, Carlini et al. also discovered that loss-based MIAs are hardly effective under constraint of low FPR (e.g., FPR $< 0.1\%$). Hence, they devised a more advanced MIA, i.e. Likelihood Ratio Attack (LiRA) based on the model output difference caused by membership of an input. We consider the *online LiRA attack* because of its high attack performance. Particularly, the adversary first prepares shadow models ahead of the attack by sampling N sub-datasets and training shadow models on each of the sampled dataset. Hence, for each data sample, there are $\frac{N}{2}$ shadow models that are trained on it (called IN models) and the rest $\frac{N}{2}$ that are not trained on it (called OUT models). The adversary then measures the means $\boldsymbol{\mu}_{in}, \boldsymbol{\mu}_{out}$ and the variances $\boldsymbol{\sigma}^2_{in}, \boldsymbol{\sigma}^2_{out}$ of model confidence for IN and OUT models, respectively. Here, the confidence of model f for $(\mathbf{x}, y)$ is $\phi(f(\mathbf{x})_y) = \phi(\exp(-l(f(\mathbf{x}), y)))$, where l is the cross-entropy loss and $\phi(p) = \log(\frac{p}{1-p})$. To attack, the adversary queries the victim model f with a target example $(\mathbf{x}, y)$ to estimate the likelihood Λ defined as:

$$\Lambda = \frac{p(\mathrm{conf}_{obs}|\mathcal{N}(\boldsymbol{\mu}_{in}, \boldsymbol{\sigma}^2_{in}))}{p(\mathrm{conf}_{obs}|\mathcal{N}(\boldsymbol{\mu}_{out}, \boldsymbol{\sigma}^2_{out}))}, \quad (8)$$

where $\mathrm{conf}_{obs} = \phi(f(\mathbf{x})_y)$ is the confidence of victim model f on target example $(\mathbf{x}, y)$. The adversary infers

membership by thresholding the likelihood Λ with threshold τ determined in advance.

3. Problem Statement

In practice, companies may utilize personal data for model training in order to provide better services. For example, data holders (e.g., smart retail stores, smart city facilities) may capture clients' data and send to cloud servers for model training. However, models trained on the raw data (i.e., $\mathcal{T}$) can be attacked by MIA. In addition, transmitting raw data to servers suffers from potential data leakage (e.g., to honest-but-curious operators). Therefore, a better protocol is to first learn knowledge from data by, for instance, generating synthetic dataset $\mathcal{S}$ from the raw data (i.e., $\mathcal{T}$), and then send $\mathcal{S}$ to the server for model training for downstream applications. Formally, we define the threat model as follows:

Adversary Goal. The adversary aims to examine the membership information of the target dataset $\mathcal{T}$. Specifically, for a sample of interest $\mathbf{x}$, the adversary infers whether $\mathbf{x} \in \mathcal{T}$.

Adversary Knowledge. We assume a strong adversary (e.g., honest-but-curious server), who although has no access to $\mathcal{T}$ but has the *white-box* access to both the synthetic dataset $\mathcal{S}$ synthesized from the target dataset $\mathcal{T}$ and the model $f_{\mathcal{S}}$ trained on the synthetic dataset. The adversary also knows the data distribution of $\mathcal{T}$.

Adversary Capacity. The adversary has unlimited computational power to generate shadow synthetic datasets on data of same distribution of $\mathcal{T}$ and train shadow models on them.

Note that white-box access to the model parameters does not help MIA (Sablayrolles et al., 2019), so we omit other advantages brought by the white-box access to $f_{\mathcal{S}}$.

4. Theoretical Analysis

In this section, we theoretically analyse the relationship between the target dataset $\mathcal{T}$ and the synthetic dataset $\mathcal{S}$ of DM (improved DC) (Zhao & Bilen, 2021a), and the privacy guarantees of $\mathcal{T}$ that are thereby provided. The reason of choosing DM is because of its high condensation efficiency and utility for model training. We also verify the difference between DM and other DC methods (see Appendix E.3) in terms of the synthetic data distribution, indicating our theoretical results can be generalized to other methods to some extent. Theoretical analysis of other DC methods (e.g., DSA) is left as the future work.

Overview. We briefly present an overview of the analysis that consists of three parts. First, we clarify the assumptions and notations in Section 4.1. Then, in Section 4.2, we analyse the connection between synthetic and original datasets for different DC initializations. Finally, in Section 4.3, with conclusion of Section 4.2, we study the privacy loss of models trained on DC-synthesized data in a DP manner: how does removing one sample in the original dataset impact models trained on synthetic dataset. Because of the randomness in model training, we base on the model parameter distribution assumption from (Sablayrolles et al., 2019) and compute the order of magnitude of the impact, which establishes the connection between DC and DP.

4.1. Assumptions & Notations

The DM loss (6) can be optimized for each class (Zhao & Bilen, 2021a). To simplify the notations, we consider only one class and omit the label vectors in the synthetic dataset $\mathcal{S} = \{\mathbf{s}_1, \cdots, \mathbf{s}_{|\mathcal{S}|}\}$ and the target dataset $\mathcal{T} = \{\mathbf{x}_1, \cdots, \mathbf{x}_{|\mathcal{T}|}\}$. We consider the following two assumptions of the target dataset and the convergence of DM.

Assumption 4.1. The linear span of the target dataset $\text{span}(\mathcal{T})$ satisfies $d_{\mathcal{T}} = \dim(\text{span}(\mathcal{T})) < d$, where d is the data dimension, $\dim(V)$ represents the dimension of vector space V, $\text{span}(\mathcal{T})$ is the vector subspace generated by all linear combinations of $\mathcal{T}$:

$$\text{span}(\mathcal{T}) := \{\sum_{i=1}^{|T|} w_i \mathbf{x}_i | 1 \leq i \leq |\mathcal{T}|, w_i \in \mathbb{R}, \mathbf{x}_i \in \mathcal{T}\}. \tag{9}$$

In practice, $d_{\mathcal{T}}$ can be computed as the rank of matrix form of $\mathcal{T}$. This assumption generally holds for high dimensional data and can be directly verified for common datasets (e.g., CIFAR-10). Without loss of generality, we consider an orthogonal basis (under inner product of $\mathbb{R}^d$) $\mathcal{E} = \{\mathbf{e}_1, \cdots, \mathbf{e}_d\}$ among which the first $d_{\mathcal{T}}$ basis vectors $\mathcal{E}_{\mathcal{T}} = \{\mathbf{e}_1, \cdots, \mathbf{e}_{d_{\mathcal{T}}}\}$ form the orthogonal basis of vector subspace $\text{span}(\mathcal{T})$.

Assumption 4.2. [Convergence of DM]. We assume there exists at least one synthetic dataset $\mathcal{S}^* = \{\mathbf{s}_1^*, \cdots, \mathbf{s}_{|\mathcal{S}|}^*\}$ that minimizes (6).

4.2. Analysis of Synthetic Data

We first analyse synthetic data by linear extractors and then discuss the generalization to non-linear case (Remark 4.5).

Proposition 4.3 (Minimizer of DM Loss)**.** *For a linear extractor $\psi_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^k$ such that $k < d$, $\boldsymbol{\theta} \in \mathbb{R}^{k \times d}$, under Assumption 4.1 and 4.2, the dataset $\mathcal{S}^*$ synthesized by DM from the target dataset $\mathcal{T}$ satisfies:*

1) The barycenters of $\mathcal{S}^*$ and $\mathcal{T}$ coincide:

$$\frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \mathbf{x}_i - \frac{1}{|\mathcal{S}^*|} \sum_{i=1}^{|\mathcal{S}^*|} \mathbf{s}_i^* = \mathbf{0}, \tag{10}$$

2) $\forall \mathbf{s}_i^* \in \mathcal{S}^*, \mathbf{s}_i^* = \mathbf{s}_{i,\mathcal{E}_{\mathcal{T}}}^* + \mathbf{s}_{i,\mathcal{E}_{\mathcal{T}}^\perp}^*$, where $\mathbf{s}_{i,\mathcal{E}_{\mathcal{T}}}^* \in$ span$(\mathcal{T})$,

$\mathbf{s}^*_{i,\mathcal{E}^\perp_\mathcal{T}} \in span(\mathcal{T})^\perp$ *that verifies*

$$\sum_{i=1}^{|\mathcal{S}|} \mathbf{s}^*_{i,\mathcal{E}^\perp_\mathcal{T}} = \mathbf{0}_{\mathcal{E}^\perp_\mathcal{T}}. \tag{11}$$

The proof of Proposition 4.3 can be found in Appendix A. Note that the minimizer is $\mathbf{s}^*_1 = \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\mathbf{x}_i$ when $|\mathcal{S}^*| = 1$, indicating that the synthetic data falls into vector subspace span$(\mathcal{T})$, confirming the Proposition 4.3.

The DC initialization of synthetic dataset can be either real data sampled from $\mathcal{T}$ or random noise. Next, we study the impact of DM initialization and obtain the following results (proof can be found in Appendix B).

Proposition 4.4 (Connection between $\mathcal{S}^*$ and $\mathcal{T}$)**.** *Based on Proposition 4.3, the minimizer synthetic dataset* $\mathcal{S}^* = \{\mathbf{s}^*_1, \cdots, \mathbf{s}^*_{|\mathcal{S}|}\}$ *has the following properties for different initialization strategies:*

1) Real data initialization. Assume that $\mathcal{S}$ *is initialized with first* $|\mathcal{S}|$ *samples of* $\mathcal{T}$*, i.e.,* $\mathbf{s}_i = \mathbf{x}_i$*, then we have*

$$\mathbf{s}^*_i = \mathbf{x}_i + \frac{1}{|\mathcal{T}|}\sum_{j=1}^{|\mathcal{T}|}\mathbf{x}_j - \frac{1}{|\mathcal{S}|}\sum_{j=1}^{|\mathcal{S}|}\mathbf{s}_j \in span(\mathcal{T}). \tag{12}$$

2) Random initialization. The synthetic data are initialized with noise of normal distribution, i.e., $\forall \mathbf{s}_i \in \mathcal{S}, \mathbf{s}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$*, and we assume the empirical mean is zeroed, i.e.,* $\frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\mathbf{s}_i = \mathbf{0}$*, then we have*

$$\mathbf{s}^*_i = \mathbf{s}^*_{i,\mathcal{E}_\mathcal{T}} + \mathbf{s}^*_{i,\mathcal{E}^\perp_\mathcal{T}}, \tag{13}$$

where $\mathbf{s}^*_{i,\mathcal{E}_\mathcal{T}} = \mathbf{s}_{i,\mathcal{E}_\mathcal{T}} + \frac{1}{|\mathcal{T}|}\sum_{j=1}^{|\mathcal{T}|}\mathbf{x}_j \in span(\mathcal{T})$*, and* $\mathbf{s}^*_{i,\mathcal{E}^\perp_\mathcal{T}} = \mathbf{s}_{i,\mathcal{E}^\perp_\mathcal{T}} \in (span(\mathcal{T}))^\perp$.

Remark 4.5 (Non-linear Extractor). Our results can be generalized to the non-linear extractors. Giryes et al. proved that multi-layer random neural networks generate distance-preserving embedding of input data, so (6) is minimized if and only if the distance between real and synthetic data is minimized. Take 2-layer random networks as an example (Estrach et al., 2014), there exist two constants $0 < A \leq B$ such that $\forall(\mathbf{x}, \mathbf{y}) \in (\mathbb{R}^d)^2$, $A\|\mathbf{x}-\mathbf{y}\|_2 \leq \|\rho(\boldsymbol{\theta}\mathbf{x}) - \rho(\boldsymbol{\theta}\mathbf{y})\|_2 \leq B\|\mathbf{x}-\mathbf{y}\|_2$, where ρ is ReLU. We also analyse the case of 2-layer extractors (activated by ReLU) and found that the (pseudo)-barycenters of $\mathcal{S}^*$ and $\mathcal{T}$ still coincide. Moreover, on convolutional extractors and the 2-layer extractors, we empirically verify Proposition 4.3 in Appendix D (see Figure 7).

Remark 4.6 (Impact of initialization on privacy). Note that in case of real data initialization, a higher $|\mathcal{S}|$ results in lower distance between barycenters of initialized $\mathcal{S}$ and $\mathcal{T}$, thus the changes brought to $\mathcal{S}$ become smaller when $\mathcal{S}$ becomes larger. This explains the phenomenon that DM-generated images are more visually similar to the real images for higher ipc (images per class) (Zhao & Bilen, 2021a). However, as we demonstrate in our experiments (Section 5.2), the membership of data used for DC initialization can still be inferred by vanilla loss-based MIA. One countermeasure is to choose hard-to-infer samples (Carlini et al., 2022), i.e., samples whose model outputs are not affected by the membership, as initialization data.

On the other hand, data not used for initialization generate little effect (i.e., their weights in synthetic data are $O(\frac{1}{|\mathcal{T}|})$) on synthetic data, just as the case of random initialization, where data in $\mathcal{T}$ also generate little effect on $\mathcal{S}^*$. We demonstrate that the membership of those data cannot be inferred under both loss-based MIA and the state-of-the-art likelihood MIA (see Section 5.2). Moreover, projection component of $(\text{span}(\mathcal{T}))^\perp$ can further protect the privacy (e.g., visual privacy in Section 5.4).

Remark 4.7 (Comparison between DC and GAN). The generator of GAN is trained to minimize the distance between the real and the generated data distributions, which is similar to the objective of DC. However, GAN-generated data share the same constraints (i.e., bounded between 0 and 1) as the real data. DC-generated data do not need to satisfy these constraints. This enables the DC-generated data to contain more features and explains the higher accuracy of models trained on DC-generated data (Zhao et al., 2021). We also empirically compare the accuracy of model trained on GAN-generated data and DC-synthesized data (see Section 5.3) , and found that DC-synthesized data outperform GAN-generated data for training better models with smaller amount of training data.

4.3. Privacy Bound of Models Trained on Synthetic Data

To understand how synthetic dataset protects membership privacy of $\mathcal{T}$ when being used for training model $f_\mathcal{S}$, we estimate how model parameters change when removing one sample from $\mathcal{T}$ by adopting below assumption. With a little abuse of notation, we denote the minimizer set $\mathcal{S}^*$ by $\mathcal{S}$ when the context is clear.

Assumption 4.8 (Distribution of model parameter (Sablayrolles et al., 2019))**.** The distribution of model parameter $\boldsymbol{\theta}$ given training dataset $\mathcal{T} = \{\mathbf{x}_1, \cdots, \mathbf{x}_{|\mathcal{T}|}\}$ and loss function l is:

$$\mathbb{P}(\boldsymbol{\theta}|\mathcal{T}) = \frac{1}{K_\mathcal{T}}\exp\left(-\sum_{i=1}^{|\mathcal{T}|} l(\boldsymbol{\theta}, \mathbf{x}_i)\right), \tag{14}$$

where $K_\mathcal{T}$ is the constant normalizing the distribution.

Unlike widely known DP mechanisms (e.g., Gaussian mechanism) that transform the deterministic query function into a randomized one, randomness brought by optimization al-

gorithm (i.e., SGD) or hardware defaults leads to different parameters each time of training, which justifies Assumption 4.8 and ensures the "uncertainty" in DP. In addition, we need the following assumption on the datasets $\mathcal{S}$, $\mathcal{T}$ and the loss function l introduced in the Assumption 4.8. The assumption is valid for finite datasets and common loss functions (e.g., cross-entropy) and is used to quantify the data bound and loss variation.

Assumption 4.9. We assume the data of $\mathcal{T}$ and $\mathcal{S}$ are bounded, i.e.,

$$\exists B > 0, \forall \mathbf{x} \in \mathcal{T} \cup \mathcal{S}, \|\mathbf{x}\|_2 \leq B. \tag{15}$$

The loss function $l(\boldsymbol{\theta}, \cdot) : \mathbb{R}^d \to \mathbb{R}^+$ is L-Lipschitz according to the L_2 norm, i.e.,

$$\forall (\mathbf{x}, \mathbf{y}) \in \mathcal{B}(B)^2, \boldsymbol{\theta}, |l(\boldsymbol{\theta}, \mathbf{x}) - l(\boldsymbol{\theta}, \mathbf{y})| \leq L \|\mathbf{x} - \mathbf{y}\|_2, \tag{16}$$

where $\mathcal{B}(B) = \{\mathbf{x} | \|\mathbf{x}\| \leq B\}$ is the close ball of space $\mathbb{R}^d$.

With all previous assumptions, we have the following result.

Proposition 4.10. *Suppose a target dataset $\mathcal{T} = \{\mathbf{x}_1, \cdots, \mathbf{x}_{|\mathcal{T}|}\}$ and the leave-one-out dataset $\mathcal{T}' = \mathcal{T} \backslash \{\mathbf{x}\}$ such that $\mathbf{x}$ is not used for initialization. The synthetic datasets are $\mathcal{S}$ and $\mathcal{S}'$ and $|\mathcal{S}| = |\mathcal{S}'| \ll |\mathcal{T}|$. Denote the model parameter distributions of $\mathcal{S}$ and $\mathcal{S}'$ by $p(\boldsymbol{\theta}) = \mathbb{P}(\boldsymbol{\theta}|\mathcal{S})$ and $q(\boldsymbol{\theta}) = \mathbb{P}(\boldsymbol{\theta}|\mathcal{S}')$ respectively. Then, the membership privacy leakage caused by removing $\mathbf{x}$ is*

$$D_{KL}(p||q) = O(\frac{|\mathcal{S}|}{|\mathcal{T}|}). \tag{17}$$

Proposition 4.10 indicates that the adversary can only obtain limited information (i.e., $O(\frac{|\mathcal{S}|}{|\mathcal{T}|})$) by MIA when the synthetic data is much fewer than the original data ($|\mathcal{S}| \ll |\mathcal{T}|$), which explains why synthetic data $\mathcal{S}$ protects membership privacy of model $f_{\mathcal{S}}$. The proof is in Appendix C.

Connection to DP. Our privacy analysis is based on the impact of model parameter distribution by removing one element from the origin training dataset, which is similar to the definition of DP (Dwork et al., 2006). Formally, a ϵ-differential privacy mechanism $\mathcal{M}$ satisfies:

$$\ln \frac{\mathbb{P}(\mathcal{M}(D) \in \mathcal{S}_{\mathcal{M}})}{\mathbb{P}(\mathcal{M}(D') \in \mathcal{S}_{\mathcal{M}})} \leq \epsilon \tag{18}$$

for all neighbor dataset pair (D, D') and all subset $\mathcal{S}_{\mathcal{M}}$ of the range of $\mathcal{M}$. Without knowledge of explicit form of model parameter distribution, we can only claim that the privacy budget ϵ varies at the order of $O(\frac{|\mathcal{S}|}{|\mathcal{T}|})$.

In practice, we use an empirical budget $\hat{\epsilon}$ through MIA (Kairouz et al., 2015) to measure the privacy guarantee against MIA. Typically, for a MIA that achieves FPR and TPR (True Positive Rate) against a model, the empirical budget is $\hat{\epsilon} \geq \ln(TPR/FPR)$. In other words, the model behaves $\hat{\epsilon}$-differentially private to an adversary that applies MIA (i.e., threat model in Section 3).

Note that the empirical budget $\hat{\epsilon}$ is *not* equivalent to the real budget ϵ because of different threat models (Nasr et al., 2021). Nonetheless, we consider black-box MIA as the only privacy threat to the model, thus we can regard the DP budget ϵ as a model privacy metric against MIA. In this way, we can compare $\hat{\epsilon}$ and ϵ by the definition of $\hat{\epsilon}$. In Section 5.3, we show that models trained on data synthesized by DC achieve $\hat{\epsilon} \approx 2$ against threat from the state-of-the-art MIA (LiRA), and obtain accuracy much higher than differentially private generators (Chen et al., 2020a; Harder et al., 2021), indicating DC is a better option for efficient and privacy-preserving model training.

5. Evaluation

In this section, we evaluate the membership privacy of $f_{\mathcal{S}}$ for real data and random initialization. Then, we compare DC with previous DP-generators and GAN to demonstrate DC's better trade-off between privacy and utility. Finally, we investigate the visual privacy of DC-synthesized data.

5.1. Experimental Setup

Datasets & Architectures. We use three datasets: FashionMNIST (Xiao et al., 2017), CIFAR-10 (Krizhevsky et al., 2009) and CelebA (Liu et al., 2015) for gender classification. The CelebA images are center cropped to dimension 64×64, and we randomly sample $5,000$ images for each class, which is same as CIFAR-10, while FashionMNIST contains $6,000$ images for each class. We adopt the same 3-layer convolutional neural networks used in (Zhao & Bilen, 2021a) and (Nguyen et al., 2021b) as the feature extractor.

DC Settings. One important hyperparameter of DSA, DM and KIP is the ratio of image per class $r_{ipc} = \frac{|\mathcal{S}|}{|\mathcal{T}|}$. We evaluate $r_{ipc} = 0.002, 0.01$ for all methods, and for DM we add an extra evaluation $r_{ipc} = 0.02$ due to its high efficiency on producing large synthetic set. Note that r_{ipc} influences the model training efficiency: the lower r_{ipc}, the faster model training. We also consider ZCA preprocessing for KIP as it is reported to be effective for KIP performance improvement. Appendix E.1 contains more DC implementation details.

Baselines. As for non-private baseline, we adopt subset sampled from $\mathcal{T}$ (this baseline is termed real data) and data generated by conditional GAN (cGAN or GAN for short) (Mirza & Osindero, 2014) which is trained on $\mathcal{T}$. For private baseline, we choose DP-generators including GS-WGAN (Chen et al., 2020a), DP-MERF (Harder et al., 2021) and DP-Sinkhorn (Cao et al., 2021). We compare the DC methods with baselines in terms of privacy and efficiency.

MIA Settings & Attack Metrics. For each dataset, we randomly split it into two subsets of equal amount of samples and choose one subset as $\mathcal{T}$ (member data). We then synthesize dataset $\mathcal{S}$ on $\mathcal{T}$, and train a model $f_{\mathcal{S}}$ (victim model) on $\mathcal{S}$. The other subset becomes the non-member data used for testing the MIA performance. The above process is called *preparation of synthetic dataset*.

For loss-based MIA, we repeat the preparation of synthetic dataset 10 times with different random seeds. This gives us 10 groups of $\mathcal{T}$, $\mathcal{S}$ and $f_{\mathcal{S}}$. For each $f_{\mathcal{S}}$, we first select N member samples from $\mathcal{T}$ and N non-member samples, and choose an optimal threshold that maximizes the advantage score on the previously chosen $2N$ samples (Sablayrolles et al., 2019). The threshold is then tested on another disjoint $2N$ samples composed by N member samples and N non-member samples to compute the advantage score of loss-based MIA. We report the advantage (in percentage) defined as $2 \times (acc - 50\%)$ where acc is the test accuracy of membership in percentage.

For LiRA (Carlini et al., 2022), we repeat the preparation of synthetic dataset N_m times with different random seeds, and obtain N_m shadow $\mathcal{T}$, $\mathcal{S}$ and $f_{\mathcal{S}}$. We set $N_m = 256$ for DM and $N_m = 64$ for KIP because of its lower training efficiency. We omit DSA for LiRA due to longer training time. To attack a victim model, we compute the likelihoods of each sample with N_m shadow $f_{\mathcal{S}}$ and determine the threshold of likelihood according to the requirements of false positive. We use the Receiver Operating Characteristic (ROC) curve and Area Under Curve (AUC) score to evaluate the attack performance. Remark that we adopt the strongest (and unrealistic) attack assumption (i.e., the attacker knows the membership), so that we investigate the privacy of DC-synthesized data under the *worst* case.

5.2. Membership Privacy of $f_{\mathcal{S}}$

Table 1. Advantage (%) of loss-based MIA against models trained on real data (baseline) and data synthesized by DSA, DM and KIP with *real data* initialization.

Method	r_{ipc}	FashionMNST	CIFAR-10	CelebA
Real (baseline, non-private)	0.002	46.67 ± 16.33	72.00 ± 24.00	100.00 ± 0.00
	0.01	21.00 ± 3.67	92.80 ± 5.31	84.00 ± 5.06
	0.02	17.33 ± 2.91	82.60 ± 5.59	77.00 ± 6.71
DM	0.002	78.17 ± 3.20	49.80 ± 5.83	37.00 ± 12.69
	0.01	83.67 ± 2.77	64.20 ± 4.77	47.00 ± 19.52
	0.02	83.00 ± 2.56	68.20 ± 7.35	53.00 ± 14.18
DSA	0.002	74.40 ± 2.65	55.40 ± 8.20	30.50 ± 8.16
	0.01	81.60 ± 2.27	56.60 ± 2.95	28.00 ± 3.74
KIP (w/o ZCA)	0.002	67.83 ± 4.54	42.40 ± 4.80	23.00 ± 11.87
	0.01	70.00 ± 2.47	51.40 ± 5.73	25.00 ± 15.65
KIP (w/ ZCA)	0.002	67.67 ± 4.42	50.40 ± 5.35	23.00 ± 15.52
	0.01	64.00 ± 4.23	48.40 ± 6.62	17.00 ± 18.47

Real Data Initialization Leaks Membership Privacy. We begin with membership privacy leaked by $f_{\mathcal{S}}$ as mentioned in Remark 4.6 of Proposition 4.4. We aim to verify that DC with real data initialization still leaks membership privacy of the data used for initialization. Here, the data used for initialization are sampled from $\mathcal{T}$ during each time of preparation of synthetic dataset. We launch loss-based MIA against $f_{\mathcal{S}}$ and adopt the real data baseline.

Table 1 shows the advantage score of loss-based MIA. Here, we vary a little bit the attack setting: the advantage scores are computed with the data used for real initialization and the same amount of member data not used for initialization but involved in DC. We observe that, on CIFAR-10 and CelebA, the synthetic dataset with real data initialization achieves lower advantage scores comparing to directly using real data for training (baseline). This can be explained by Proposition 4.4, which tells us that the synthesized data deviates slightly from the real data used for initialization. However, on FashionMNIST, the baseline has lower advantage scores. We suspect this is because FashionMNIST images are greyscale and synthetic data contain more features that prone to be memorized by networks. Loss distribution in Figure 8 in Appendix E.2 also demonstrates that *synthetic data with real data initialization can still leak membership privacy*.

Next, we show that models trained on data synthesized by DC with random initialization are robust against both loss-based MIA and LiRA (the state-of-the-art MIA).

MIA Robustness of Random Initialization. Because of random initialization, each member sample contributes equally to the synthetic dataset. Thus, in this case, we follows the loss-base attack setting and set $N = 1000$. Table 2 provides the average and the standard variance of advantages for models trained on synthetic datasets by cGAN, DSA, DM and KIP. The advantages are around 0 for all r_{ipc}, signifying that the adversary cannot infer membership of $\mathcal{T}$. Nevertheless, as long as the adversary has access to the generated images (which is included in our threat model), the membership of the GAN generator's training data (i.e., $\mathcal{T}$) can still be leaked (see Appendix E.4). Meanwhile, as we show later in Figure 3, models trained on DC-synthesized data achieve higher accuracy scores than baseline (i.e., cGAN-generated data), demonstrating the higher utility of DC-synthesized data.

LiRA is a more powerful MIA because it can achieve higher TPR at low FPR (Carlini et al., 2022), while the adversary's computational cost is higher. Figure 2 provides the ROC curves of LiRA against $f_{\mathcal{S}}$. We can observe that the ROC curves are close to the diagonal (red line) for all datasets and r_{ipc}. The AUC scores of ROC curves are around 0.5, indicating *there is negligible attack benefit (low TPR) for the attacker compared with random guess*. Recall that LiRA is evaluated on the whole dataset (half as member and other half as non-member), the minimum FPR value is $\frac{1}{5000} = 2\times$

Table 2. Advantage (%) of loss-based MIA against models trained on data synthesized by cGAN (baseline), DSA, DM and KIP with *random* initialization.

Methods	r_{ipc}	FashionMNST	CIFAR-10	CelebA
cGAN (baseline, non-private)	0.002	0.29 ± 0.89	-0.44 ± 1.88	-0.57 ± 0.97
	0.01	0.18 ± 1.21	-0.58 ± 2.09	-0.81 ± 0.95
	0.02	0.04 ± 0.70	-0.77 ± 1.59	-0.47 ± 1.22
DM	0.002	-0.34 ± 0.42	0.31 ± 1.93	-0.66 ± 1.44
	0.01	-0.29 ± 0.48	1.06 ± 1.20	-0.56 ± 1.52
	0.02	0.18 ± 0.53	0.72 ± 0.70	-0.67 ± 1.18
DSA	0.002	0.09 ± 0.51	0.39 ± 1.04	-0.39 ± 1.90
	0.01	0.52 ± 0.55	1.27 ± 1.71	-1.16 ± 0.90
KIP (w/o ZCA)	0.002	-1.13 ± 1.84	0.25 ± 1.20	-0.56 ± 1.07
	0.01	-0.95 ± 0.96	0.25 ± 1.80	-1.51 ± 0.69
KIP (w/ ZCA)	0.002	-0.56 ± 2.02	-0.64 ± 1.86	-1.06 ± 1.10
	0.01	-1.69 ± 1.96	-0.22 ± 1.27	-1.80 ± 1.91

10^{-4} for CelebA, 4×10^{-5} for CIFAR-10 and 3.33×10^{-5} for FashionMNIST. Therefore, when FPR is close to 0, the ROC curves have different shapes for different datasets. However, we also notice that the TPR is close to FPR when FPR is around the minimum (e.g., FPR$\sim 10^{-4}$ for CelebA), demonstrating that *models trained on data synthesized by DC with random initialization is robust to LiRA at low FPR.*

5.3. Comparison with Different Generators

Table 3. Utility comparison of dataset synthesized by DP-generators, DM and KIP. The utility is measured by the accuracy (%) of models trained on the synthetic dataset. The results are estimated on FashionMNIST.

Method	DP Budget	r_{ipc} 0.002	r_{ipc} 0.01	r_{ipc} 0.02
GS-WGAN	$\epsilon = 10$	53.53 ± 0.42	51.85 ± 0.54	50.10 ± 0.32
DP-MERF	$\epsilon = 10$	52.18 ± 0.37	52.88 ± 0.75	50.73 ± 0.66
	$\epsilon = 2$	60.41 ± 0.78	55.14 ± 0.61	56.39 ± 0.45
DP-Sinkhorn	$\epsilon = 10$	-	-	70.9*
KIP (w/o ZCA)	$\hat{\epsilon} = 1.25$	73.70 ± 1.13	68.11 ± 1.33	-
KIP (w/ ZCA)	$\hat{\epsilon} = 2.07$	74.37 ± 0.96	70.03 ± 0.84	-
DM	$\hat{\epsilon} = 2.30$	80.59 ± 0.62	85.10 ± 0.51	86.13 ± 0.34

* Results reported in the paper (Cao et al., 2021) ($r_{ipc} = 1$).

GAN Generator. Figure 3 compares the accuracy scores of models trained on synthetic datasets (DC-synthesized with random initialization under $r_{ipc} = 0.01$). We can find that under the same constraint of training efficiency (i.e., $r_{ipc} = 0.01$), the DM and DSA outperform the other methods. Note that models trained on KIP-synthesized data achieve lower accuracy than baseline because the loss is hard to converge for large r_{ipc}. Nevertheless, for small r_{ipc}, the KIP significantly outperforms baselines on CIFAR-10 and CelebA (see Figure 10 in Appendix E).

Then, we aim to know how DC improves model training efficiency compared to cGAN. In other words, to achieve the same accuracy of f_S, the difference between the r_{ipc} that DC requires and the r_{ipc} that cGAN requires can be seen as the model training efficiency that DC improves. For different r_{ipc} (the x-axis), Figure 4 shows the accuracy of models trained on cGAN-generated dataset whose ratio is r_{ipc} (the blue solid curve). The red and green horizontal lines represent the accuracy of f_S trained on dataset synthesized by DSA and DM for $r_{ipc} = 0.01$, respectively. We omit KIP here because of its lower utility than baselines. Therefore, the r_{ipc} of the intersection point of the red (resp. green) line and the blue curve is the r_{ipc} of cGANs-generated dataset on which the models can be trained to achieve the same accuracy as DSA (resp. DM). We can see that, cGAN needs to generate more data to train a model that achieves the same accuracy as models trained on data synthesized by DM and DSA, because the r_{ipc} values indicated in the x-axis are all higher than 0.01. It is worth noting that DC improves the training efficiency (measured by r_{ipc}) by at least 2 times than cGAN for $r_{ipc} = 0.01$, because on FashionMNIST (the leftmost sub-figure in Figure 4)), cGAN requires to generate synthetic dataset of $r_{ipc} = 0.02$ to achieve the same accuracy (0.85) as the DM-synthesized dataset ($r_{ipc} = 0.01$).

DP-generators. We estimate an empirical $\hat{\epsilon}$ based on the ratio of TPR and FPR computed by LiRA. In Table 3, we compare the accuracy of models trained on DC-synthesized data and on data generated by recent DP-generators. We reproduce DP-MERF and GS-WGAN according to the official implementation and adopt the reported results of DP-Sinkhorn. We observe that the accuracy of models trained on data generated by the state-of-the-art DP-generator (DP-Sinkhorn) is still lower than DM-synthesized images, even the ratio for DP-Sinkhorn is $r_{ipc} = 1$. The reason is that DP is designed to defend against the strongest adversary who has access to the training process of generator. Hence, data generated by DP-generators are of lower utility for model training because of the too strong defense requirement.

5.4. Visual Privacy

The adversary can directly visualize synthetic data and compare with the target sample to infer the membership. We visualize the synthetic images and use L_2 distance as well as the perceptual metric LPIPS (Zhang et al., 2018) with VGG backbone to measure the similarity between synthetic and real images. Figure 5 shows examples of DM-generated images and the their (top 3) most similar real images, i.e., images of lowest L_2 and LPIPS distance with the synthetic image on the top of the column. We observe that the real data share similar facial contour patterns with the synthetic images, but more fine-grained features, e.g., eye shape, are different, which explains why models trained on synthetic dataset protect the membership privacy of original data. This can also explain why current MIAs fail on models trained on synthetic datasets: the generated synthetic training data

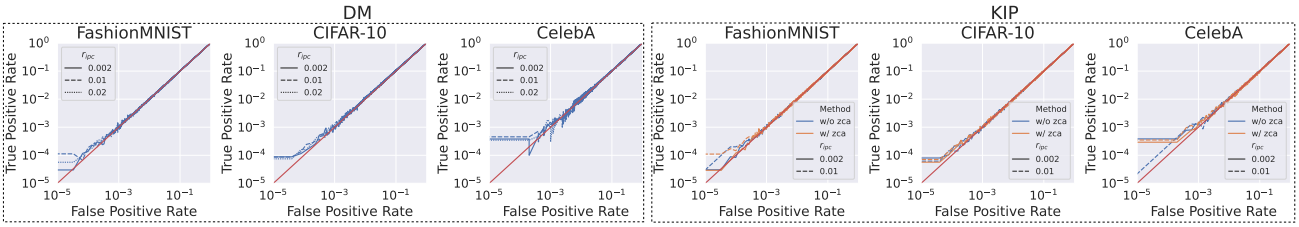

Figure 2. ROC curves of LiRA against models trained on data synthesized by DM (left three figures) and KIP (right three figures). The solid, dashed and dotted lines stand for results of $r_{ipc} = 0.002, 0.01$ and 0.02, respectively. In KIP figures, the orange and blue lines represent the results of KIP with and without ZCA preprocessing, respectively. The red diagonal represents random guess and the AUC scores of ROC curves are all under 0.51.

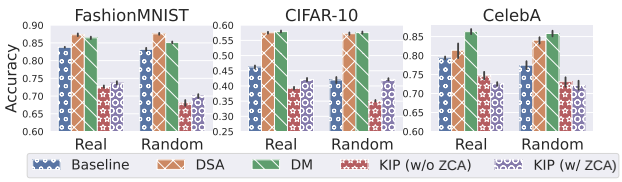

Figure 3. Accuracy of models trained on data synthesized by DSA, DM, KIP and on data generated by baselines for $r_{ipc} = 0.01$. The x-axis represents initialization strategy. For real data and random initialization, the baselines are real data and cGAN-generated data, respectively.

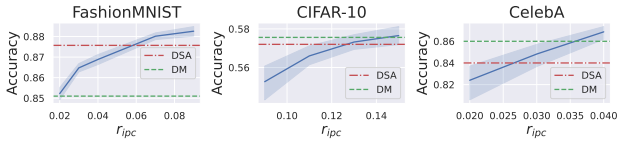

Figure 4. Accuracy of models trained on cGAN-generated data for different r_{ipc}. The horizontal lines are accuracy of models trained on data synthesized by DM (green, dashed) and DSA (red, dash-dotted) for $r_{ipc} = 0.01$.

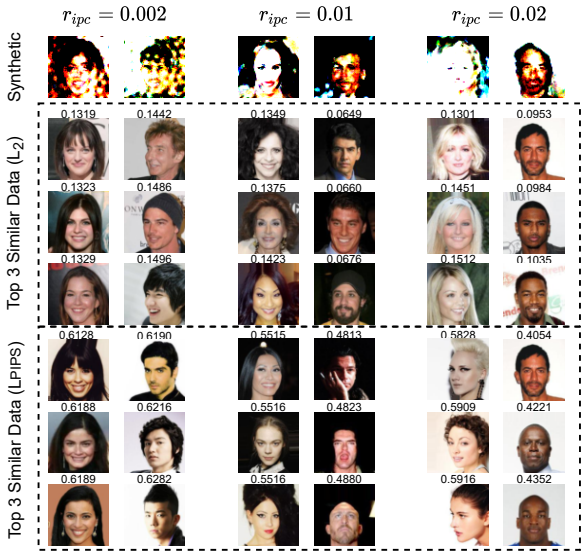

Figure 5. Examples of facial images that are most similar to synthetic data generated by DM with *random* initialization. The value above each image is the distance (L_2 and LPIPS) between the image and the synthetic data (first row). Lower distance indicates higher similarity. Even though these real images have similar face contour, blurred facial details (e.g., eyes, nose) make it difficult for the adversary to infer the membership.

have lost the private properties of real data and thus the adversary are not able to infer the privacy from models trained on such synthetic data.

6. Discussion and Conclusion

In this work, we make the first effort to introduce the emerging dataset condensation techniques into the privacy community and provide systematical audit including theoretical analysis of the privacy property and empirical evaluation, i.e. visual privacy examination and robustness against loss-based MIA and LiRA on FashionMNIST, CIFAR-10 and CelebA datasets.

Our future work will attempt to generalize the theoretical findings to other DC methods. This can be studied from the perspective of information loss (e.g. data compression ratio). Moreover, DC methods that satisfy formal DP formulation, e.g., (α, ϵ)-Rényi DP (Mironov, 2017), are worth exploring.

The current efforts of DC mainly focus on image classification, thus another interesting direction is the extension of the privacy benefit brought by DC to more complicated vision tasks (e.g., object detection) and non-vision tasks (e.g. text and graph related applications). In essence, DC methods can generalize to other machine learning tasks, as they learn the synthetic data by summarizing the distribution or discriminative information of real training data. Hence, their privacy advantage should also generalize to other tasks.

Acknowledgements

We would like to thank He Tong for the help in analyzing non-linear models and the anonymous reviewers for constructive feedback.

References

Bohdal, O., Yang, Y., and Hospedales, T. Flexible dataset distillation: Learn labels instead of images. *NeurIPS Workshop*, 2020.

Cao, T., Bie, A., Vahdat, A., Fidler, S., and Kreis, K. Don't generate me: Training differentially private generative models with sinkhorn divergence. *Advances in Neural Information Processing Systems*, 34, 2021.

Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramer, F. Membership inference attacks from first principles. In *43rd IEEE Symposium on Security and Privacy, SP 2022*. IEEE, 2022.

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*, 2022.

Chen, D., Orekondy, T., and Fritz, M. Gs-wgan: A gradient-sanitized approach for learning differentially private generators. In *Neural Information Processing Systems (NeurIPS)*, 2020a.

Chen, D., Yu, N., Zhang, Y., and Fritz, M. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 343–362, 2020b.

Choquette-Choo, C. A., Tramer, F., Carlini, N., and Papernot, N. Label-only membership inference attacks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1964–1974. PMLR, 2021.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284, 2006.

Estrach, J. B., Szlam, A., and LeCun, Y. Signal recovery from pooling representations. In *International conference on machine learning*, pp. 307–315. PMLR, 2014.

Fredrikson, M., Jha, S., and Ristenpart, T. Model inversion attacks that exploit confidence information and basic countermeasures. In *CCS*, pp. 1322–1333, 2015.

Giryes, R., Sapiro, G., and Bronstein, A. M. Deep neural networks with random gaussian weights: A universal classification strategy? *IEEE Transactions on Signal Processing*, 64(13):3444–3457, 2016.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Harder, F., Adamczewski, K., and Park, M. Dp-merf: Differentially private mean embeddings with randomfeatures for practical privacy-preserving data generation. In *International Conference on Artificial Intelligence and Statistics*, pp. 1819–1827, 2021.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Jacot, A., Hongler, C., and Gabriel, F. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, pp. 8580–8589, 2018.

Jin, W., Zhao, L., Zhang, S., Liu, Y., Tang, J., and Shah, N. Graph condensation for graph neural networks. *ICLR*, 2022.

Kairouz, P., Oh, S., and Viswanath, P. The composition theorem for differential privacy. In *International conference on machine learning*, pp. 1376–1385. PMLR, 2015.

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

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, G., Togo, R., Ogawa, T., and Haseyama, M. Soft-label anonymous gastric x-ray image distillation. In *IEEE International Conference on Image Processing, ICIP 2020, Abu Dhabi, United Arab Emirates, October 25-28, 2020*, pp. 305–309. IEEE, 2020.

Li, Z. and Zhang, Y. Membership leakage in label-only exposures. In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 880–895, 2021.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Lyu, L., Yu, H., Zhao, J., and Yang, Q. Threats to federated learning. In *Federated Learning*, pp. 3–16. Springer, 2020.

Melis, L., Song, C., De Cristofaro, E., and Shmatikov, V. Exploiting unintended feature leakage in collaborative learning. In *SP*, pp. 691–706, 2019.

Mironov, I. Rényi differential privacy. In *30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017*, pp. 263–275. IEEE Computer Society, 2017.

Mirza, M. and Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Nasr, M., Songi, S., Thakurta, A., Papemoti, N., and Carlin, N. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 866–882. IEEE, 2021.

Nguyen, T., Chen, Z., and Lee, J. Dataset meta-learning from kernel ridge-regression. In *International Conference on Learning Representations*, 2021a.

Nguyen, T., Novak, R., Xiao, L., and Lee, J. Dataset distillation with infinitely wide convolutional networks. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021b.

Powell, M. J. D. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Comput. J.*, 7(2):155–162, 1964.

Rezaei, S. and Liu, X. On the difficulty of membership inference attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7892–7900, 2021.

Sablayrolles, A., Douze, M., Schmid, C., Ollivier, Y., and Jégou, H. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pp. 5558–5567, 2019.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2017.

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

Sucholutsky, I. and Schonlau, M. Soft-label dataset distillation and text dataset distillation. *arXiv preprint arXiv:1910.02551*, 2019.

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. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Wang, T., Zhu, J., Torralba, A., and Efros, A. A. Dataset distillation. *CoRR*, abs/1811.10959, 2018. URL http://arxiv.org/abs/1811.10959.

Wang, Y., Ding, Z., Xiao, Y., Kifer, D., and Zhang, D. Dpgen: Automated program synthesis for differential privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 393–411, 2021.

Webster, R., Rabin, J., Simon, L., and Jurie, F. This person (probably) exists. identity membership attacks against gan generated faces. *arXiv preprint arXiv:2107.06018*, 2021.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Xie, L., Lin, K., Wang, S., Wang, F., and Zhou, J. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

Yu, D., Zhang, H., Chen, W., Yin, J., and Liu, T.-Y. How does data augmentation affect privacy in machine learning? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10746–10753, 2021.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Zhao, B. and Bilen, H. Dataset condensation with distribution matching. *CoRR*, abs/2110.04181, 2021a.

Zhao, B. and Bilen, H. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, 2021b.

Zhao, B., Mopuri, K. R., and Bilen, H. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021.

A. Proof of Proposition 4.3

We begin our analysis of DM with the linear extractor $\psi_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^k$ such that $k < d$, $\boldsymbol{\theta} = [\theta_{i,j}] \in \mathbb{R}^{k\times d}$ and for an input $\mathbf{x}$, $\psi_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}\mathbf{x}$. We also omit the differentiable Siamese augmentation to simplify the analysis. As the representation extractors $\psi_{\boldsymbol{\theta}}$ in DM are randomly initialized, we assume that the extractor parameters follow the standard normal distribution and are identically and independently distributed (i.i.d), i.e., $\theta_{i,j} \overset{iid}{\sim} \mathcal{N}(0,1)$. Thus, Equation (6) becomes the expectation over $\|\mathbf{d}_{DM}\|^2$ where $\mathbf{d}_{DM}$ is defined as:

$$\mathbf{d}_{DM} \coloneqq \boldsymbol{\theta}(\frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\mathbf{x}_i - \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\mathbf{s}_i). \tag{19}$$

Hence, $L_{DM} = \mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{N}(0,1)}\|\mathbf{d}_{DM}\|^2$. The optimization of $\mathcal{S}$ with SGD relies on the gradient of (6). Given a sampled model parameter $\boldsymbol{\theta}$, for some synthetic sample $\mathbf{s}_j$, we have:

$$\frac{\partial\|\mathbf{d}_{DM}\|^2}{\partial\mathbf{s}_i} = -\frac{2}{|\mathcal{S}|}(\mathbf{d}_{DM})^\top\boldsymbol{\theta} = -\frac{2}{|\mathcal{S}|}(\frac{1}{|\mathcal{T}|}\sum_{j=1}^{|\mathcal{T}|}\mathbf{x}_j - \frac{1}{|\mathcal{S}|}\sum_{j=1}^{|\mathcal{S}|}\mathbf{s}_j)^\top\cdot\boldsymbol{\theta}^\top\boldsymbol{\theta}. \tag{20}$$

Hence, we obtain:

$$\boxed{\frac{\partial L_{DM}}{\partial\mathbf{s}_i} = \frac{\partial\mathbb{E}_{\boldsymbol{\theta}}\|\mathbf{d}_{DM}\|^2}{\partial\mathbf{s}_i} = \mathbb{E}_{\boldsymbol{\theta}}\frac{\partial\|\mathbf{d}_{DM}\|^2}{\partial\mathbf{s}_i} = -\frac{2}{|\mathcal{S}|}(\frac{1}{|\mathcal{T}|}\sum_{j=1}^{|\mathcal{T}|}\mathbf{x}_j - \frac{1}{|\mathcal{S}|}\sum_{j=1}^{|\mathcal{S}|}\mathbf{s}_j)^\top\cdot\mathbb{E}[\boldsymbol{\theta}^\top\boldsymbol{\theta}],} \tag{21}$$

where $\mathbb{E}[\boldsymbol{\theta}^\top\boldsymbol{\theta}] = k\mathbf{I}_d$ by definition of $\boldsymbol{\theta}$, and $\mathbf{I}_d$ is the identity matrix of $\mathbb{R}^d$. Equation (21) indicates that the optimization direction of synthetic sample $\mathbf{s}_i$ is the direction of moving barycenter of $\mathcal{S}$ to the barycenter of $\mathcal{T}$. To conceptually interpret, the optimization of (6) will move the initialized $\mathcal{S}$ until the barycenter coincides with that of $\mathcal{T}$ because of the existence of minimizer (Assumption 4.2) where the left hand-side of (21) should be $\mathbf{0}$.

B. Proof of Proposition 4.4

Case of real data initialization. Suppose that each $\mathbf{s}_i \in \mathcal{S}$ is sampled from $\mathcal{T}$ and we can consider $\mathbf{s}_i = \mathbf{x}_i$ as initialization for simplicity. According to (21), all $\mathbf{s}_i$ are optimized until the barycenters of $\mathcal{S}$ and $\mathcal{T}$ coincide. Observe that each $\mathbf{s}_j^* \in \text{span}(\mathcal{T})$, because the projection components of $\text{span}(\mathcal{T})^\perp$ remain zero throughout the optimization process of DM. Thus, one solution of minimizer elements $\mathbf{s}_i^* \in \mathcal{S}^*$ with the real data initialization can be:

$$\boxed{\mathbf{s}_i^* = \mathbf{x}_i + \frac{1}{|\mathcal{T}|}\sum_{j=1}^{|\mathcal{T}|}\mathbf{x}_j - \frac{1}{|\mathcal{S}|}\sum_{j=1}^{|\mathcal{S}|}\mathbf{s}_j.} \tag{22}$$

When $|\mathcal{S}|$ and $|\mathcal{T}|$ are large (e.g., > 50), we can consider $\mathbf{s}_i^* \approx \mathbf{x}_i$, thus *initialization with real data in DM still risks of membership privacy leakage*.

Case of random initialization. The synthetic data are initialized as vectors of multivariate normal distribution, i.e.,

$$\forall\mathbf{s}_i \in \mathcal{S}, \mathbf{s}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d). \tag{23}$$

Each synthetic sample $\mathbf{s}_i$ can be written as a vector $[s_{i,1}, \cdots, s_{i,d}]$ under the basis $\mathcal{E}$ where $\forall j, s_{i,j} \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$, because $\mathbf{s}_i$'s covariance matrix remains identity matrix under any orthogonal transformation (i.e., orthogonal basis). Thus, we can decompose $\mathbf{d}_{DM}$ to the projections on subspace $\text{span}(\mathcal{T})$ and its orthogonal complement. Formally, we have

$$\begin{aligned}\|\mathbf{d}_{DM}\|^2 = \left\|\boldsymbol{\theta}_{1:d_\mathcal{T}}\text{Proj}_{\mathcal{E}_\mathcal{T}}(\Delta_{\mathcal{S},\mathcal{T}})\right\|^2 + \underbrace{\left\|\boldsymbol{\theta}_{d_\mathcal{T}:d}\text{Proj}_{\mathcal{E}_\mathcal{T}^\perp}(\Delta_{\mathcal{S},\mathcal{T}})\right\|^2}_{\|\mathbf{d}_{DM}\|^2_{\mathcal{E}_\mathcal{T}^\perp}} \\ + 2 < \boldsymbol{\theta}_{1:d_\mathcal{T}}\text{Proj}_{\mathcal{E}_\mathcal{T}}(\Delta_{\mathcal{S},\mathcal{T}}), \boldsymbol{\theta}_{d_\mathcal{T}:d}\text{Proj}_{\mathcal{E}_\mathcal{T}^\perp}(\Delta_{\mathcal{S},\mathcal{T}}) >,\end{aligned} \tag{24}$$

where $\boldsymbol{\theta}_{a:b}$ represents the submatrix composed by a-th column to the b-th column, Proj_V is the projection operator onto subspace V and

$$\Delta_{\mathcal{S},\mathcal{T}} \coloneqq \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\mathbf{x}_i - \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\mathbf{s}_i. \tag{25}$$

Note that

$$\text{Proj}_{\mathcal{E}_\mathcal{T}^\perp}(\Delta_{\mathcal{S},\mathcal{T}}) = \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\text{Proj}_{\mathcal{E}_\mathcal{T}^\perp}(\mathbf{s}_i), \tag{26}$$

because $\mathbf{x}_i \in \text{span}(\mathcal{T}) = \text{span}(\mathcal{E}_\mathcal{T})$ for each $\mathbf{x}_i$. Let $\mathbf{s}_{i,\mathcal{E}_\mathcal{T}^\perp} = \text{Proj}_{\mathcal{E}_\mathcal{T}^\perp}(\mathbf{s}_i)$, then we have

$$\mathbb{E}_{\boldsymbol{\theta}}\frac{\partial \left\|\mathbf{d}_{DM}\right\|^2_{\mathcal{E}_\mathcal{T}^\perp}}{\partial \mathbf{s}_{i,\mathcal{E}_\mathcal{T}^\perp}} = \frac{2}{|\mathcal{S}|^2}(\sum_{j=1}^{|\mathcal{S}|}\mathbf{s}_{j,\mathcal{E}_\mathcal{T}^\perp})^\top \mathbb{E}_{\boldsymbol{\theta}}[(\boldsymbol{\theta}_{d_\mathcal{T}:d})^\top \boldsymbol{\theta}_{d_\mathcal{T}:d}], \tag{27}$$

because $\mathbb{E}_{\boldsymbol{\theta}}[\boldsymbol{\theta}_{1:d_\mathcal{T}}^\top \boldsymbol{\theta}_{d_\mathcal{T}:d}] = \mathbf{0}$. Therefore, the expectation of the above equation is the optimization direction of the projection of $\mathbf{s}_j$ on the subspace $(\text{span}(\mathcal{T}))^\perp$:

$$\frac{\partial L_{DM}}{\partial \mathbf{s}_{i,\mathcal{E}_\mathcal{T}^\perp}} = \mathbb{E}_{\boldsymbol{\theta}}\frac{\partial \left\|\mathbf{d}_{DM}\right\|^2_{\mathcal{E}_\mathcal{T}^\perp}}{\partial \mathbf{s}_{i,\mathcal{E}_\mathcal{T}^\perp}} = \frac{2\mathbb{E}[(\boldsymbol{\theta}_{d_\mathcal{T}:d})^\top \boldsymbol{\theta}_{d_\mathcal{T}:d}]}{|\mathcal{S}|^2}(\sum_{j=1}^{|\mathcal{S}|}\mathbf{s}_{j,\mathcal{E}_\mathcal{T}^\perp})^\top. \tag{28}$$

Note that $\mathbb{E}[(\boldsymbol{\theta}_{d_\mathcal{T}:d})^\top \boldsymbol{\theta}_{d_\mathcal{T}:d}] = k\mathbf{I}_{d-d_\mathcal{T}}$, thus the optimization direction is aligned with the barycenter of all $\mathbf{s}_{i,\mathcal{E}_\mathcal{T}^\perp}$ and will converge to 0 when

$$\frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\mathbf{s}_{i,\mathcal{E}_\mathcal{T}^\perp} = \mathbf{0}_{\mathcal{E}_\mathcal{T}^\perp}. \tag{29}$$

Since the initialization of $\mathcal{S}$ is essentially noise of standard normal distribution, the empirical average of $\mathbf{s}_{i,\mathcal{E}_\mathcal{T}^\perp}$ is close to $\mathbf{0}$ (by law of large numbers), thus we can consider that the projection component of minimizer $\mathbf{s}^*_{i,\mathcal{E}_\mathcal{T}^\perp}$ is close to the initialized value, i.e.,

$$\boxed{\forall \mathbf{s}^*_{i,\mathcal{E}_\mathcal{T}^\perp} \in \mathcal{S}^*, \mathbf{s}^*_{i,\mathcal{E}_\mathcal{T}^\perp} \approx \mathbf{s}_{i,\mathcal{E}_\mathcal{T}^\perp}.} \tag{30}$$

Similar as the case of real data initialization, the projection components on $\text{span}(\mathcal{T})$ of $\mathbf{s}_i$ are optimized to verify the first property of Proposition 4.3, i.e., the projection component of i-th minimizer $\mathbf{s}^*_{i,\mathcal{E}_\mathcal{T}}$ becomes

$$\boxed{\mathbf{s}^*_{i,\mathcal{E}_\mathcal{T}} = \mathbf{s}_{i,\mathcal{E}_\mathcal{T}} + \frac{1}{|\mathcal{T}|}\sum_{j=1}^{|\mathcal{T}|}\mathbf{x}_j - \frac{1}{|\mathcal{S}|}\sum_{j=1}^{|\mathcal{S}|}\mathbf{s}_{j,\mathcal{E}_\mathcal{T}}.} \tag{31}$$

B.1. Empirical verification

We empirically verify our conclusions for random and real data initializations in Figure 6. The images are synthesized from CIFAR-10 by DM using linear extractor of embedding dimension 2048, and each line contains images from the same class. On the right side, we plot the images synthesized with random initialization and real data initialization. We can observe that images synthesized with random initialization resemble combination of noise and class-dependent background, which verifies our conclusion of random initialization: synthetic data with random initialization are composed of barycenter of original data in space span and initialized noise in space $\text{span}(\mathcal{T})^\perp$ (see (13)). Note that even in this case, models trained on synthetic data can still achieve validation accuracy around 27%.

On the other hand, real data initialization generates little changes on the images used for initialization, which verifies the conclusion of real data initialization: synthetic data with real data initialization are composed of images used for initialization and the barycenter distance vector (see (12)).

Besides linear extractor, we also investigate the impact of activation function. On the left of Figure 6, we show images synthesized by DM using ReLU-activated extractor (ReLU on top of linear extractor). We can see that the existence of

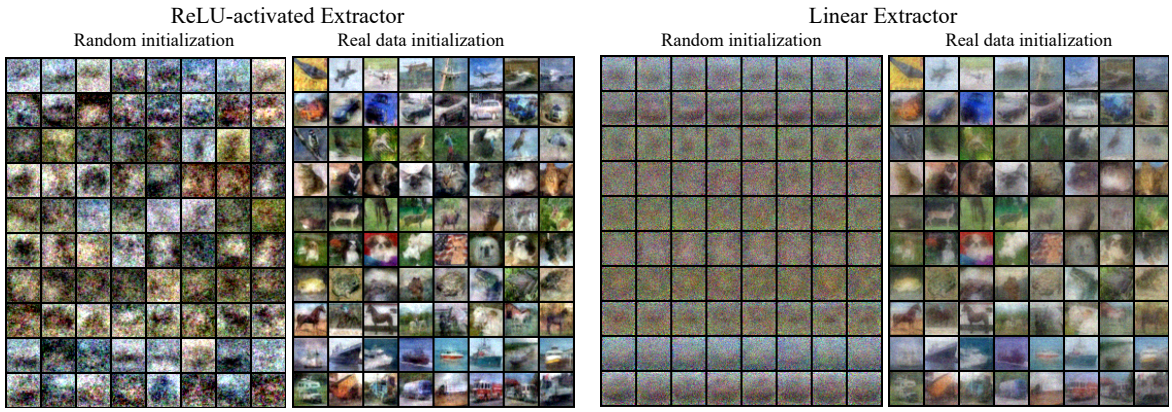

Figure 6. Synthetic images with random noise/real data initialization and with linear/ReLU-activated extractor of CIFAR-10.

ReLU results in better convergence of DC and thus better image quality for both random and real data initialization. A potential reason is that ReLU changes the DC optimization and can lead to different local minima other than that found by using linear extractor. That is, data synthesized in this case are possibly composed by barycenter of a certain group of similar images (e.g., images of brown horse heading towards right with grass background) within the same class and a orthogonal noise vector. For example, CIFAR-10 synthetic images of class "horse" (third last line of leftmost figure in Figure 6) are noisy but contain different backgrounds which should be the barycenter of different image group of class "horse": there are numerous horse images in CIFAR-10 where the horse head towards the right or left. This observation also confirms that ReLU improves generalization of neural networks. Appendix D encompasses more detailed analysis for non-linear extractor.

C. Proof of Proposition 4.10

We aim to quantify the membership privacy leakage of a member $\mathbf{x}$ with the Kullback-Leibler (KL) divergence of model parameter distributions. Without loss of generality, we study how the last element $\mathbf{x}_{|\mathcal{T}|}$ influences the model parameter distribution. Let $\mathcal{T}'$ denote $\mathcal{T} \setminus \{\mathbf{x}_{|\mathcal{T}|}\}$, where $\mathcal{T} = \{\mathbf{x}_1, \cdots, \mathbf{x}_{|\mathcal{T}|}\}$. The synthetic datasets by DM based on $\mathcal{T}$ and $\mathcal{T}'$ are noted as $\mathcal{S}$ and $\mathcal{S}'$, respectively, and $|\mathcal{S}| = |\mathcal{S}'|$. In addition, we denote $p(\boldsymbol{\theta}) = \mathbb{P}(\boldsymbol{\theta}|\mathcal{S})$ and $q(\boldsymbol{\theta}) = \mathbb{P}(\boldsymbol{\theta}|\mathcal{S}')$. The KL divergence between p and q is

$$D_{KL}(p||q) = \int_{\boldsymbol{\theta}} p(\boldsymbol{\theta}) \ln \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \mathrm{d}\boldsymbol{\theta} = \int_{\boldsymbol{\theta}} \frac{1}{K_{\mathcal{S}}} \exp\left(-\sum_{i=1}^{|\mathcal{S}|} l(\boldsymbol{\theta}, \mathbf{s}_1)\right) \ln \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \mathrm{d}\boldsymbol{\theta}, \tag{32}$$

where

$$\begin{aligned} \ln \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} &= \sum_{i=1}^{|\mathcal{S}'|} l(\boldsymbol{\theta}, \mathbf{s}'_i) - \sum_{i=1}^{|\mathcal{S}|} l(\boldsymbol{\theta}, \mathbf{s}_i) + K_{\mathcal{S}'} - K_{\mathcal{S}} \\ &= \sum_{i=1}^{|\mathcal{S}|} (l(\boldsymbol{\theta}, \mathbf{s}'_i) - l(\boldsymbol{\theta}, \mathbf{s}_i)) + K_{\mathcal{S}'} - K_{\mathcal{S}} \\ &\leq L \sum_{i=1}^{|\mathcal{S}|} \|\mathbf{s}'_i - \mathbf{s}_i\|_2 + |K_{\mathcal{S}'} - K_{\mathcal{S}}|. \end{aligned} \tag{33}$$

According to the assumption 4.8, $K_{\mathcal{S}}$ (similar for $K_{\mathcal{S}'}$) is:

$$K_{\mathcal{S}} \coloneqq \int_{\boldsymbol{\theta}} \exp\left(-\sum_{i=1}^{|\mathcal{S}|} l(\boldsymbol{\theta}, \mathbf{s}_i)\right) \mathrm{d}\boldsymbol{\theta}. \tag{34}$$

Since $\mathbf{x}_{|\mathcal{T}|}$ is not used for real data initialization, according to the Proposition 4.4, if $\mathcal{S}$ and $\mathcal{S}'$ share the same initialization

type and initialized values, we have for each i

$$\begin{aligned}\|\mathbf{s}_i' - \mathbf{s}_i\|_2 &= \left\| \frac{1}{|\mathcal{T}|-1} \sum_{j=1}^{|\mathcal{T}|-1} \mathbf{x}_j - \frac{1}{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{T}|} \mathbf{x}_j \right\|_2 \\ &= \frac{1}{|\mathcal{T}|} \left\| \frac{1}{|\mathcal{T}|-1} \sum_{j=1}^{|\mathcal{T}|-1} \mathbf{x}_j - \mathbf{x}_{|\mathcal{T}|} \right\|_2 \\ &\leq \frac{2B}{|\mathcal{T}|}.\end{aligned} \tag{35}$$

Thus, we have

$$L \sum_{i=1}^{|\mathcal{S}|} \|\mathbf{s}_i' - \mathbf{s}_i\|_2 \leq \frac{2LB\,|\mathcal{S}|}{|\mathcal{T}|}. \tag{36}$$

The second term on the right side of (33) can be processed similarly:

$$\begin{aligned}|K_{\mathcal{S}'} - K_{\mathcal{S}}| &= \left| \int_{\boldsymbol{\theta}} \exp\left(-\sum_{i=1}^{|\mathcal{S}'|} l(\boldsymbol{\theta}, \mathbf{s}_i')\right) - \exp\left(-\sum_{i=1}^{|\mathcal{S}|} l(\boldsymbol{\theta}, \mathbf{s}_i)\right) \mathrm{d}\boldsymbol{\theta} \right| \\ &= \left| \int_{\boldsymbol{\theta}} [\exp(\sum_{i=1}^{|\mathcal{S}|} (l(\boldsymbol{\theta}, \mathbf{s}_i) - l(\boldsymbol{\theta}, \mathbf{s}_i'))) - 1] \exp\left(-\sum_{i=1}^{|\mathcal{S}|} l(\boldsymbol{\theta}, \mathbf{s}_i)\right) \mathrm{d}\boldsymbol{\theta} \right|.\end{aligned} \tag{37}$$

From previous analysis, we know that

$$\sum_{i=1}^{|\mathcal{S}|} (l(\boldsymbol{\theta}, \mathbf{s}_i) - l(\boldsymbol{\theta}, \mathbf{s}_i')) \leq \frac{2LB\,|\mathcal{S}|}{|\mathcal{T}|}. \tag{38}$$

Since $\exp(x) - 1 = O(x)$ in the neighborhood of 0, we have

$$|K_{\mathcal{S}'} - K_{\mathcal{S}}| = O(\frac{2LB\,|\mathcal{S}|\,K_{\mathcal{S}}}{|\mathcal{T}|}) = O(\frac{|\mathcal{S}|}{|\mathcal{T}|}). \tag{39}$$

Note that $K_{\mathcal{S}}$ should decrease as $|\mathcal{S}|$ increases because an additional synthetic sample $\mathbf{s}$ introduces a factor $\exp(-l(\boldsymbol{\theta}, \mathbf{s})) \leq 1$ in the integral. We omit it here and assume $K_{\mathcal{S}}$ varies little when $|\mathcal{S}|$ changes. Together with (32) and (33), we obtain the privacy bound by KL divergence:

$$\boxed{D_{KL}(p||q) = O(\frac{|\mathcal{S}|}{|\mathcal{T}|}).} \tag{40}$$

D. Generalization to Non-Linear Extractor

We consider 2-layer network as the extractor, i.e., linear extractor with ReLU activation, and show that the (pseudo)-barycenters of $\mathcal{S}$ and $\mathcal{T}$ also coincide as claimed by Proposition 4.3. We then empirically validate the conclusion by plotting the L_2 distance between $\mathcal{S}$ and $\mathcal{T}$ during the condensation by DM for different r_{ipc} on CIFAR-10 (see Figure 7).

D.1. Analysis for 2-layer Network as Extractor

With activation function ReLU (noted as ρ), Equation (19) becomes:

$$\mathbf{d}_{DM}^{ReLU} := \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \rho(\boldsymbol{\theta} \cdot \mathbf{x}_i) - \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \rho(\boldsymbol{\theta} \cdot \mathbf{s}_i). \tag{41}$$

Since $\theta_{i,j} \overset{iid}{\sim} \mathcal{N}(0,1)$ for each element $\theta_{i,j}$ of $\boldsymbol{\theta}$, for an input $\mathbf{x} = [x_j]_{1\le j\le d} \in \mathbb{R}^d$, we have

$$\mathbf{y} = \boldsymbol{\theta}\cdot\mathbf{x} = [\sum_{j=1}^{d}\theta_{i,j}x_j]_{1\le i\le k} = [y_i]_{1\le i\le k} \in \mathbb{R}^k, \tag{42}$$

where $y_i \overset{iid}{\sim} \mathcal{N}(0, \sum_{j=1}^{d} x_j^2)$. Since $\rho(x) \coloneqq \max(0,x)$, we have $\rho(\mathbf{y}) = [\max(0,y_i)]_{1\le i\le k}$. Define $Y = \max(0,X)$ where the random variable $X \sim \mathcal{N}(0,\sigma^2)$. Then, Y follows the same distribution of $B\,|X|$, where $B \sim Bernoulli(\frac{1}{2})$ independent of X and $\mathbb{E}_X[Y] = \mathbb{E}_B[B]\mathbb{E}_X[|X|]$. Therefore, for each i, $\max(0,y_i) = B_i\,|y_i| = B_i\left|\sum_{j=1}^{d}\theta_{i,j}x_j\right|$, and we can obtain

$$\rho(\mathbf{y}) = \rho(\boldsymbol{\theta}\mathbf{x}) = \mathbf{B}\odot|\boldsymbol{\theta}\mathbf{x}| \tag{43}$$

where $\odot$ is element-wise multiplication, $\mathbf{B} = [B_i]_{1\le i\le k}$ and $B_i \overset{iid}{\sim} Bernoulli(\frac{1}{2})$. With this in mind, Equation (41) becomes:

$$\mathbf{d}_{DM}^{ReLU} = \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\rho(\boldsymbol{\theta}\cdot\mathbf{x}_i) - \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\rho(\boldsymbol{\theta}\cdot\mathbf{s}_i) = \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\mathbf{B}_i^x\odot|\boldsymbol{\theta}\mathbf{x}_i| - \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\mathbf{B}_i^s\odot|\boldsymbol{\theta}\mathbf{s}_i|\,, \tag{44}$$

where vectors of Bernoulli random variable for each data samples $\mathbf{B}_i^{\cdot}$ are independent. To simplify notation, we consider $k=1$. The vector of Bernoulli random variable reduces to single random variable $B_i^{\cdot}$, and the bold symbol $\mathbf{d}$ becomes d. Moreover, let $\text{sgn}(x)$ denote the sign of x, and we can see that $|x| = \text{sgn}(x)x$ for a real number x. Thus, with $k=1$, we can reduce d_{DM}^{ReLU} to the similar form of (19):

$$\begin{aligned} d_{DM}^{ReLU} &= \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}B_i^x\text{sgn}(\boldsymbol{\theta}\mathbf{x}_i)\boldsymbol{\theta}\mathbf{x}_i - \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}B_i^s\text{sgn}(\boldsymbol{\theta}\mathbf{s}_i)\boldsymbol{\theta}\mathbf{s}_i \\ &= \boldsymbol{\theta}(\frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}B_i^x\text{sgn}(\boldsymbol{\theta}\mathbf{x}_i)\mathbf{x}_i - \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}B_i^s\text{sgn}(\boldsymbol{\theta}\mathbf{s}_i)\mathbf{s}_i). \end{aligned} \tag{45}$$

Recall that for each j, $\partial L_{DM}/\partial\mathbf{s}_j = \mathbb{E}_{\boldsymbol{\theta}}[(d_{DM}^{ReLU})^2/\partial\mathbf{s}_j] = \mathbb{E}_{\boldsymbol{\theta}}[2(\partial d_{DM}^{ReLU}/\partial\mathbf{s}_j)d_{DM}^{ReLU}]$, and we have

$$\frac{\partial d_{DM}^{ReLU}}{\partial\mathbf{s}_j} = -\frac{1}{|\mathcal{S}|}B_j^s\text{sgn}(\boldsymbol{\theta}\mathbf{s}_j)\boldsymbol{\theta}^\top. \tag{46}$$

Thus, the gradient of L_{DM} on $\mathbf{s}_j$ becomes:

$$\begin{aligned} \frac{L_{DM}}{\mathbf{s}_j} &= \mathbb{E}_{B,\boldsymbol{\theta}}[-\frac{2}{|\mathcal{S}|}\boldsymbol{\theta}^\top\boldsymbol{\theta}(\frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}B_j^s\text{sgn}(\boldsymbol{\theta}\mathbf{s}_j)B_i^x\text{sgn}(\boldsymbol{\theta}\mathbf{x}_i)\mathbf{x}_i - \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}B_j^s\text{sgn}(\boldsymbol{\theta}\mathbf{s}_j)B_i^s\text{sgn}(\boldsymbol{\theta}\mathbf{s}_i)\mathbf{s}_i)] \\ &= -\frac{2}{|\mathcal{S}|}(\frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\mathbb{E}_B[B_j^sB_i^x]\mathbb{E}_{\boldsymbol{\theta}}[\text{sgn}(\boldsymbol{\theta}\mathbf{s}_j)\text{sgn}(\boldsymbol{\theta}\mathbf{x}_i)\boldsymbol{\theta}^\top\boldsymbol{\theta}]\mathbf{x}_i - \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\mathbb{E}_B[B_j^sB_i^s]\mathbb{E}_{\boldsymbol{\theta}}[\text{sgn}(\boldsymbol{\theta}\mathbf{s}_j)\text{sgn}(\boldsymbol{\theta}\mathbf{s}_i)\boldsymbol{\theta}^\top\boldsymbol{\theta}]\mathbf{s}_i). \end{aligned} \tag{47}$$

Let $M(\mathbf{x},\mathbf{y})$ denote $\mathbb{E}_{\boldsymbol{\theta}}[\text{sgn}(\boldsymbol{\theta}\mathbf{x})\text{sgn}(\boldsymbol{\theta}\mathbf{y})\boldsymbol{\theta}^\top\boldsymbol{\theta}] \in \mathbb{R}^{d\times d}$, then the above equation becomes:

$$\boxed{\frac{L_{DM}}{\mathbf{s}_j} = -\frac{2}{|\mathcal{S}|}(\frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\frac{1}{4}M(\mathbf{s}_j,\mathbf{x}_i)\mathbf{x}_i - \frac{1}{|\mathcal{S}|}\sum_{i=1,i\neq j}^{|\mathcal{S}|}\frac{1}{4}M(\mathbf{s}_j,\mathbf{s}_i)\mathbf{s}_i - \frac{1}{2\,|\mathcal{S}|}\mathbf{s}_j),} \tag{48}$$

because $M(\mathbf{x},\mathbf{y}) = \mathbf{I}_d$ if $\mathbf{x} = \mathbf{y}$. Note that if $\mathbf{x} = -\mathbf{y}$, then $M(\mathbf{x},\mathbf{y}) = -\mathbf{I}_d$. In fact, we can prove that

$$\mathbf{x}^\top M(\mathbf{x},\mathbf{y})\mathbf{y} = \mathbb{E}_{\boldsymbol{\theta}}[(\text{sgn}(\boldsymbol{\theta}\mathbf{x})\boldsymbol{\theta}\mathbf{x})^\top(\text{sgn}(\boldsymbol{\theta}\mathbf{y})\boldsymbol{\theta}\mathbf{y})] = \frac{\|\mathbf{x}\|_2\,\|\mathbf{y}\|_2}{\pi}[(\pi-2\phi)\cos(\phi)+2\sin(\phi)], \tag{49}$$

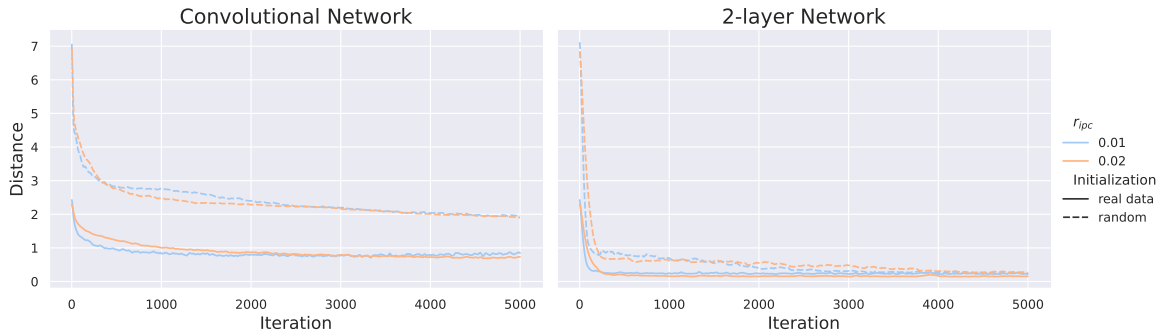

Figure 7. Distance ($\|\cdot\|_2$) between barycenters of $\mathcal{S}$ and $\mathcal{T}$ deceases with the iteration round, which verifies first property of Proposition 4.3. The solid and dashed lines represent real data and random initialization, respectively. The blue and orange lines represent the cases where r_{ipc} equals to 0.01 and 0.02, respectively.

which can be seen as a matrix depending on the angle ϕ between $\mathbf{x}$ and $\mathbf{y}$. Even though each original data $\mathbf{x}_i$ is varied by $M(\mathbf{s}_j, \mathbf{x}_i)$, their average can still be seen as a pseudo-barycenter, and the above equation signifies that each $\mathbf{s}_j$ is updated towards minimizing the distance between the pseudo-barycenters of $\mathcal{T}$ and $\mathcal{S}$, which verifies the first property of Proposition 4.3 on non-linear extractor. This further validates the privacy property of DM which is based on the connection between $\mathcal{S}$ and $\mathcal{T}$.

Next, we empirically verify the Proposition 4.3 with tests on CIFAR-10.

D.2. Empirical verification of Proposition 4.3 for non-linear extractor

Figure 7 shows the distance $\left\| \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\mathbf{s}_i - \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\mathbf{x}_i \right\|_2$ for each DM iteration on CIFAR-10. We can observe that the barycenter distance decreases with the iteration round, and achieves to the minimum. Note that the right subfigure of Figure 7 shows that the barycenters of $\mathcal{T}$ and $\mathcal{S}$ synthesized on 2-layer network (i.e., linear model activated by ReLU) have distance around 0, validating the theoretical analysis above. As for convolutional network (ConvNet), the distance decreases slower than 2-layer network. We suspect that the convolutional layers will lead the optimization to a local minimum. Figure 7 also validates the impact of r_{ipc} and initialization to the distance of barycenters of $\mathcal{S}$ and $\mathcal{T}$: 1) when the iteration round is around 0, the distance of 100 image per class is smaller than that of 50 image per class, 2) the real initialization has much lower distance than of random initialization at the beginning of DM optimization.

E. Additional Experimental Details and Results

All experiments are conducted with Pytorch 1.10 on a Ubuntu 20.04 server.

E.1. Details of Hyperparameters and Settings.

DC Settings. We reproduced DM (Zhao & Bilen, 2021a) and adopt large learning rates to accelerate the condensation (i.e., $10, 50, 100$ as learning rate for $r_{ipc} = 0.002, 0.01, 0.02$, respectively). For DSA (Zhao & Bilen, 2021b), we adopt the default setting [1] for all datasets. For KIP, we reproduced in Pytorch according to the official code of (Nguyen et al., 2021a), and set learning rate 0.04 and 0.1 for $r_{ipc} = 0.002$ and 0.01, respectively. Note that we omit $r_{ipc} = 0.02$ for KIP and DSA due to the low efficiency. We also apply differentiable siamese augmentations (Zhao & Bilen, 2021b) for both DM and KIP.

E.2. Loss distribution of data used for DC initialization and test data on $f_{\mathcal{S}}$

In Figure 8, we show the distribution of $f_{\mathcal{S}}$ losses evaluated on data used for DC initialization (Real Init) and data not used for initialization (Other). We can observe that the losses of data used for initialization are smaller than other data, showing that the membership of data used for DC initialization are easier to be inferred. The distribution difference also explains the high advantage scores in Table 1.

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

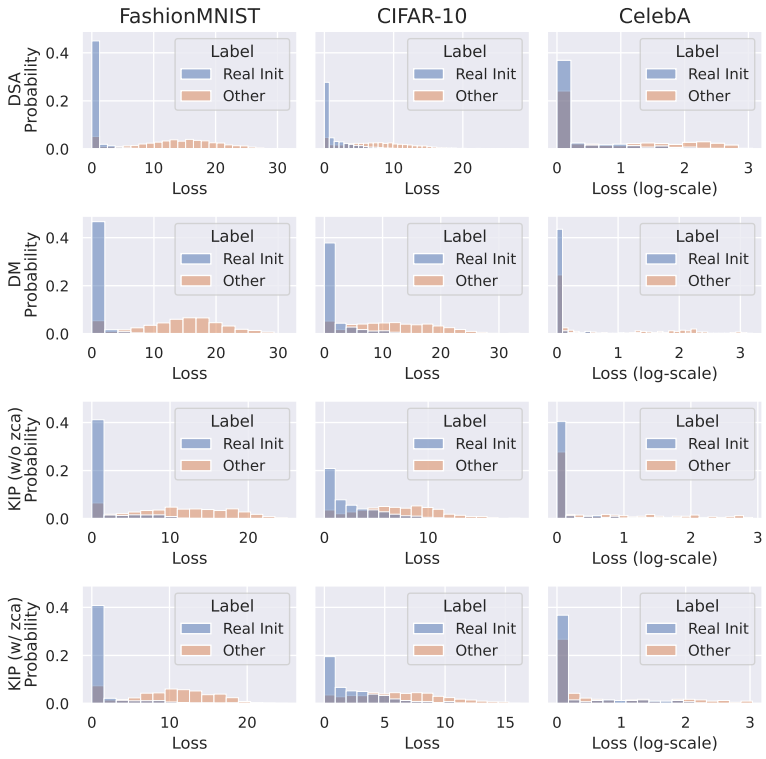

Figure 8. Loss distribution of data used for DC initialization (Real Init) are smaller than data not used for initialization (Other).

E.3. Visualization of DC-synthesized data distribution

Figure 9 shows the t-SNE visualization of CIFAR-10 and CelebA data synthesized by GAN and DC methods (DSA, DM and KIP without ZCA preprocessing). We clip the DC-synthesized into 0 and 1 for fair comparison with GAN-synthesized data. Note that the generated data distributions of DM and DSA are more similar than KIP and GAN, explaining why DM-synthesized data and DSA-synthesized data enable models to achieve higher accuracy under same r_{ipc}.

E.4. MIA against cGANs

Our threat model assumes that the adversary has white-box access to the synthetic dataset. We apply the MIA against GANs proposed by Chen et al. (called GAN-leak). The main intuition is that member data are easier to be reconstructed by GAN

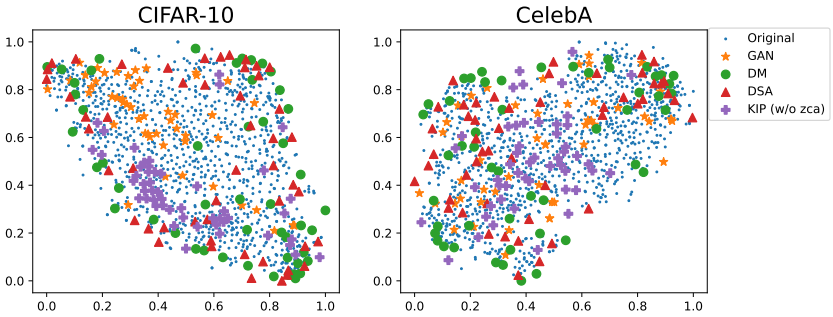

Figure 9. Distribution visualization of CIFAR-10 (left) and CelebA (right) synthesized by GAN, DSA, DM and KIP (without ZCA preprocessing).

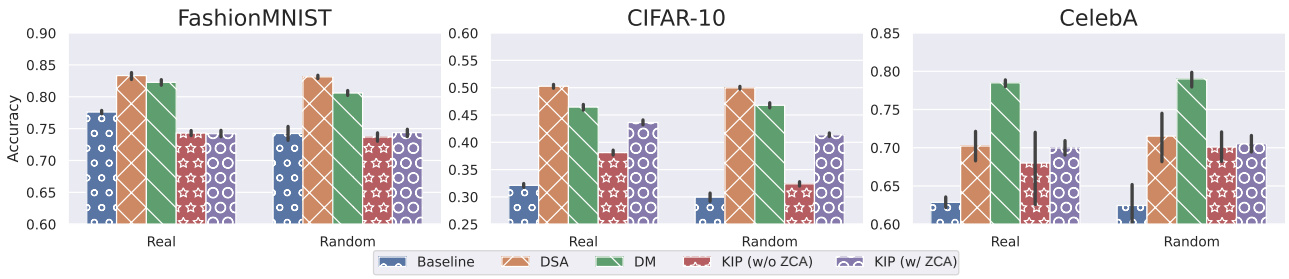

Figure 10. Accuracy of models trained on data synthesized by different DC methods and on data generated by baselines for $r_{ipc} = 0.002$.

generators $\mathcal{G}$, so the MIA is based on the (calibrated) reconstructed loss L_{cal}:

$$M(\mathbf{x}) = \mathbb{1}(L_{cal}(\mathbf{x}, \mathcal{G}(\mathbf{z})) \leq \tau). \tag{50}$$

The adversary optimizes L_{cal} by varying $\mathbf{z}$ to estimate whether $\mathbf{x}$ belongs to the training dataset. According to the adversary' knowledge, the attack can be divided into black-box attack, partial black-box attack and white-box attack. We conducted the white-box attack for scenarios where the adversary has access to the generators. The results on CelebA are in Table 4, indicating that vanilla GAN can be used to infer the membership of training data. Chen et al. also validated that partial black-box attack can achieve similar attack performance as white-box, because the adversary has access to $\mathbf{z}$ and can leverage non-differentiable optimization, e.g., the Powell's Conjugate Direction Method (Powell, 1964)), to approximately minimize L_{cal}.

Table 4. Results of GAN-leak attack against cGANs averaged over 10 shadow models.

Dataset	ROC AUC	Advantage (%)
CelebA	56.06 ± 2.03	22.98 ± 4.27

E.5. Comparison of accuracy for models trained on synthetic dataset for $r_{ipc} = 0.002$

Figure 10 presents the accuracy comparison results of models trained on data synthesized by DC and baseline methods for $r_{ipc} = 0.002$. We can see that KIP significantly outperforms baselines and achieves similar performance with DSA and DM on CIFAR-10. Moreover, we can observe that the ZCA preprocessing is effective for improving the utility of KIP-synthesized dataset.