DP-ImgSyn: Dataset Alignment for Obfuscated, Differentially Private Image Synthesis

Efstathia Soufleri *esoufler@purdue.edu*
Department of Electrical and Computer Engineering
Purdue University

Deepak Ravikumar *dravikum@purdue.edu*
Department of Electrical and Computer Engineering
Purdue University

Kaushik Roy *kaushik@purdue.edu*
Department of Electrical and Computer Engineering
Purdue University

Reviewed on OpenReview: *https://openreview.net/forum?id=KleJZ9ZzYw*

Abstract

The availability of abundant data has catalyzed the expansion of deep learning vision algorithms. However, certain vision datasets cannot be publicly released due to privacy reasons. Releasing synthetic images instead of private images is a common approach to overcome this issue. A popular method to generate synthetic images is using Generative Adversarial Networks (GANs) with Differential Privacy (DP) guarantees. However, GAN-generated synthetic images are visually similar to private images. This is a severe limitation, particularly when the private dataset depicts visually sensitive and disturbing content. To address this, we propose a non-generative framework, Differentially Private Image Synthesis (DP-ImgSyn), to generate and release synthetic images for image classification tasks. These synthetic images: (1) have DP guarantees, (2) retain the utility of the private images, i.e., a model trained using synthetic images results in similar accuracy as a model trained on private images, (3) the synthetic images are visually dissimilar to private images. DP-ImgSyn consists of the following steps: First, a teacher model is trained on the private images using a DP training algorithm. Second, public images are used as initialization for the synthetic images which are optimized to align them with the private images. The optimization uses the teacher network's batch normalization layer statistics (mean, standard deviation) to inject information about the private images into the synthetic images. Third, the synthetic images and their soft labels, obtained from the teacher model, are released and can be deployed for neural network training on image classification tasks. Our experiments on various image classification datasets show that when using similar DP training mechanisms, our framework performs better than generative techniques (up to $\approx$ **20%** in terms of image classification accuracy).[1]

1 Introduction

Deep Learning has benefited from large statistical data available to the broad community. Large datasets in the domain of image classification (Deng et al., 2009), object recognition (Lin et al., 2014), language modeling (Lewis et al., 2004), and recommendation systems (Bennett et al., 2007) have helped these domains make significant advances. However, there are cases in which data cannot be publicly released due to

[1]The PyTorch implementation can be found at `https://github.com/Efstathia-Soufleri/DP-ImgSyn`

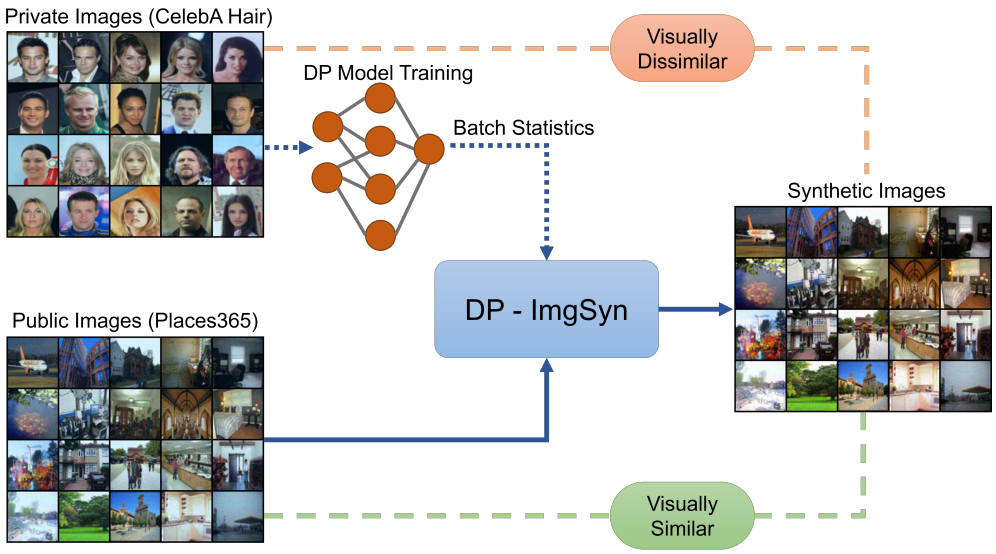


Figure 1: Visualization of the private images (CelebA-Hair), public images (Places365), and the DP-ImgSyn generated images (synthetic images). DP-ImgSyn synthesized images are visually dissimilar to private images. DP-ImgSyn is initialized with Places365 (public images/dataset). The public images and the batch statistics of a DP model (trained on the private images) are the inputs to DP-ImgSyn. DP-ImgSyn outputs synthetic images. A model trained on the synthetic images achieves $\approx 99.5\%$ the performance of a model trained with DP on the CelebA-Hair dataset (private images). Figure 6 in the Appendix contains more visualizations of other datasets.

privacy restrictions. To address this issue, several techniques have been proposed in literature which can be categorized into: model release methods (Abadi et al., 2016), and data release methods (Papernot et al., 2017). Model release methods train a classifier on the private dataset (dataset with privacy restrictions) and release the classifier with privacy guarantees. Data release methods release synthetic data with privacy guarantees instead of the private data.

Data release methods offer more flexibility than model release methods (Triastcyn & Faltings, 2018). When releasing data, downstream users can choose any model architecture (i.e., ResNet, VGG, MobileNet, etc), unlike model release methods where the user is restricted to use the architecture of the released model. Furthermore, users can combine data from different sources to build better models. Also, in the presence of newly available data, they can retrain the model on new and past data avoiding catastrophic forgetting (Kirkpatrick et al., 2017). While data release methods offer improved flexibility, synthetic images generated using existing data release techniques are visually similar to private images. This poses significant challenges when the image content is visually sensitive and/or disturbing, for example with certain medical image datasets, military datasets, content moderation image datasets, etc. This leads us to the question: can we achieve the same learning as with private images using *visually dissimilar synthetic images*? For example, can a neural network learn to classify human faces when it is trained with synthetic images depicting places? In our work, we show that such learning is achievable. Formally, we study the problem of *releasing synthetic data for image classification tasks* that satisfy the following properties:

1. The synthetic image must provide *$(\epsilon, \delta)-$Differential Privacy (DP) guarantees.*
2. Retain the *utility* of the private images, i.e., a model trained using synthetic images should result in similar classification accuracy as the model trained on private images.
3. Be *visually dissimilar* to the private images. This is critical to protect the users from viewing visually disturbing and sensitive content.

The visual dissimilarity between synthetic and private images is important for vision data depicting visually disturbing and sensitive content. This requirement is not satisfied by the existing approaches. In particular,

existing approaches often use Generative Adversarial Networks (GANs Goodfellow et al., 2014) with DP guarantees (DP-GANs Xie et al., 2018; Xu et al., 2019; Zhang et al., 2018; Xie et al., 2018; Cao et al., 2021) to generate synthetic images. While GAN-generated synthetic images have DP guarantees and retain the utility of the private images, they are visually similar to private images (see Appendix Figure 7). Thus, they do not satisfy the third requirement of visual dissimilarity. In addition to this, training a GAN can be challenging due to multiple issues (Arjovsky & Bottou, 2017) such as vanishing gradients (Arjovsky & Bottou, 2017), mode collapse (Srivastava et al., 2017; Brock et al., 2019; Arjovsky et al., 2017; Salimans et al., 2018; Miyato et al., 2018), training instability (Brock et al., 2019), and convergence failure (Salimans et al., 2016).

To address these issues, we propose a non-generative (or discriminative) approach for releasing synthetic images. Our proposed framework, Differentially Private Image Synthesis (DP-ImgSyn), trains a teacher model on the private (sensitive) dataset using a DP training algorithm, such as DP-SGD (Abadi et al., 2016). Next, a public dataset is selected for distillation (Hinton et al., 2015). This dataset is used to distill the sensitive dataset. During distillation, we observe that misalignment between the public-private datasets can result in significant performance degradation. To address dataset misalignment, we propose an alignment technique on the public dataset to improve the performance of our framework. Finally, we obtain soft labels using the teacher model and the aligned public dataset. The soft label is the teacher model output (logits) after applying the softmax function. The aligned public dataset (synthetic images) and the corresponding DP-teacher model generated soft labels, are released for training neural networks on image classification tasks. A detailed explanation of the intuition behind our method is provided in Section 2.

The synthetic images generated by DP-ImgSyn satisfy (1) privacy: the synthetic images have $(\epsilon, \delta)-$DP guarantees because the teacher model is trained with DP guarantees, (2) utility: this is experimentally verified in Section 5. A model trained on the synthetic CelebA-Hair dataset (Liu et al., 2015; Long et al., 2021) achieves $\approx 99.5\%$ the performance of a model trained with DP on the CelebA-Hair dataset, (3) the synthetic images are visually dissimilar to the private images. This is observed in Figure 1, which visualizes the private dataset (CelebA-Hair), the synthetic images generated by DP-ImgSyn (synthetic images), and the public dataset used for initialization (Places365, Zhou et al., 2017). Further, we experimentally evaluate the synthetic images in terms of classification accuracy on MNIST, FashionMNIST, CelebA-Hair, CelebA-Gender, CIFAR10, and ImageNette datasets. We observe that DP-ImgSyn reduces the misalignment between public and private distribution. This alignment improves performance up to $\approx$ **17%** on highly misaligned public-private dataset pairs. Moreover, our method achieves significantly better accuracy (up to $\approx$ **20%**) than state-of-the-art generative methods using a similar DP training algorithm. We also observe similar trends when the teacher and the student models are of different architectures (see section 5.6). Note that the proposed technique is not a new DP mechanism. It is a new approach for releasing visually dissimilar images, leveraging dataset alignment to obfuscate sensitive data in public datasets.

Overall, our contributions are summarized as follows:

- We propose DP-ImgSyn, a non-generative image synthesis framework that generates a DP-guaranteed dataset for public release. The synthetic images (1) are DP-guaranteed, (2) have similar utility to the private images, and (3) are visually dissimilar to private images.
- The DP-ImgSyn framework leverages the teacher model's batch normalization layer statistics to address the distribution misalignment between private and public datasets.
- We show the effectiveness of DP-ImgSyn in image classification tasks on various vision datasets. We also show that DP-ImgSyn performs better than state-of-the-art generative methods when using a similar DP training algorithm.

2 DP-ImgSyn: Why and How?

Why? We would like to emphasize that DP-ImgSyn shows that it is possible to learn to classify a dataset that looks very different from the training dataset. Note, that upon synthetic image release, it is up to the user to determine how to use them. Below, we provide examples showing why visual dissimilarity is important in certain classes of applications:

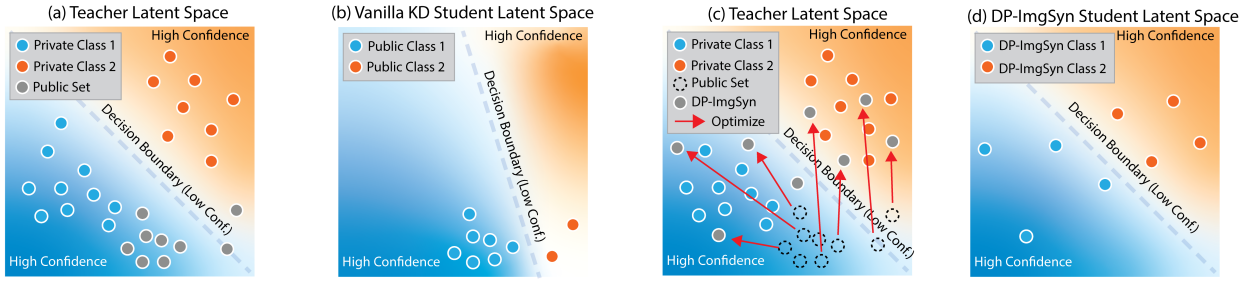


Figure 2: (a) Illustrates a cartoon version of the decision boundary of the teacher model and how public images (gray) sample the latent space. (b) Shows the transfer of the decision boundary of the teacher model in (a) to the student model using public images to sample the latent space. The student in (b) learns a decision boundary that is different from the teacher leading to poor knowledge transfer. (c) Shows the effect of optimization/perturbation of the public images with DP-ImgSyn. With DP-ImgSyn the latent space is sampled more effectively. Finally, (d) illustrates the transfer of the decision boundary of the teacher model to the student model, using synthetic images generated by DP-ImgSyn to sample the latent space.

Defense Operational Security (OPSEC) Developing systems for the detection of classified military assets (e.g., a new stealth aircraft) presents significant OPSEC challenges. Typically, only the team directly involved in the asset's development (e.g., the aircraft engineers) possesses the necessary security clearances. To develop a detection system, traditional data-sharing methods would necessitate granting security clearances to additional engineers or declassifying sensitive imagery, both of which introduce security risks (Dressler et al., 2015). DP-ImgSyn offers a potential mitigation strategy by using synthetic images that are visually obfuscated to protect classified elements while still enabling the development of effective detection models.

Medical datasets are bound by strict privacy regulations (HIPAA (Act, 1996), PIPEDA (Manley & Cavoukian, 2000), and GDPR (Regulation, 2018). While differential privacy (DP) offers robust guarantees, certain datasets may not be suitable for release due to the sensitivity of the body regions depicted (Deng et al., 2006; Abbas et al., 2022). DP-ImgSyn can facilitate the use of such datasets by generating synthetic images that preserve the essential medical information for model training but visually obscuring identifiable details.

Content Moderation DP-ImgSyn potentially addresses privacy and ethical concerns with content moderation by enabling the generation of visually dissimilar synthetic datasets for model training, safeguarding individuals in the original data and mitigating exposure of human moderators to such content (Spence et al., 2023). Other applications include automated parental control and movie age rating systems (Akyon & Temizel, 2022).

How? The proposed DP-ImgSyn method aims to learn/transfer the decision boundaries between the teacher and the student model. This is done by sampling the latent space of the teacher model using the public images and having the student model match the decision boundary of the teacher using soft labels. This implies that latent space sampling needs to be effective. The challenge is that public images may not effectively sample the latent space. Thus, our proposed optimization perturbs the public images such that the latent space can be effectively sampled. This leads to better knowledge transfer between the teacher and the student model. Figure 2 (a) illustrates a cartoon version of the decision boundary of the teacher model and how a public dataset (gray) samples the latent space. Figure 2 (b) shows the transfer of the decision boundary of the teacher model in 2 (a) to the student model using the public images to sample the latent space. We see that the student in 2 (b) learns a decision boundary different from the teacher leading to poor knowledge transfer. Figure 2 (c) shows the effect of optimization of the public images with DP-ImgSyn. With DP-ImgSyn the latent space is sampled more effectively. Finally, Figure 2 (d) illustrates the transfer of the decision boundary of the teacher model to the student model, using synthetic images generated by DP-ImgSyn to sample the latent space. The experimental results in Section 5 are in line with the DP-ImgSyn intuition. For visualizations on deep neural networks refer to Appendix A.3.

‘

3 Background and Related Work

This section briefly introduces the definition of Differential Privacy (DP), followed by an overview of approaches for releasing synthetic images using generative techniques. Differential Privacy (DP) was introduced by Dwork et al. (2006). A randomized algorithm $\mathcal{A}$ is said to be (ϵ, δ) differentially private, if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{A})$ and for all datasets $x, y \in \text{Domain}(\mathcal{A})$ such that $||x - y||_1 \leq 1$:

$$\Pr\left[\mathcal{A}(x) \in \mathcal{S}\right] \leq e^{\epsilon} \cdot \Pr\left[\mathcal{A}(y) \in \mathcal{S}\right] + \delta$$

In the case of deep learning, $\mathcal{A}$ is the training algorithm, and $\mathcal{S}$ is the subset of all possible model parameters that can be output from the training process.

In this paper, we consider the task of generating visually dissimilar synthetic images with DP guarantees. While, semi-private learning (Bie et al., 2022; Ganesh et al., 2023) might seem related to our work, there are fundamental differences. We briefly discuss these differences here. Semi-private learning leverages public data to improve the privacy bounds when learning from private data. Specifically, the private model's parameters are updated with public data to improve privacy guarantees and privacy-utility trade-off. In our work, public data is used for image synthesis and not during the training of the DP neural network. Hence, we do not compare DP-ImgSyn with semi-private learning techniques. Next, we present generative techniques for model and data release.

As far as generative techniques are concerned, they leverage GANs or other generative models by sharing DP-trained models, embeddings, or generating images for releasing datasets while maintaining privacy. Multiple research articles (Xie et al., 2018; Xu et al., 2019; Zhang et al., 2018) have proposed training GANs with DP for image synthesis using DP-SGD (Abadi et al., 2016) under various contexts and domains. The authors of GS-WGAN (Chen et al., 2020) adopt Wasserstein GAN (WGAN, Arjovsky et al., 2017) and propose using Wasserstein-1 loss for training. They show that such an approach can distort gradient information more precisely; thus GS-WGANs generate more informative samples. The authors of DataLens (Wang et al., 2021) leverage GANs to reduce the gradient noise using gradient compression. In a similar direction, to improve information capture from the gradient, the authors of DPGEN[2] (Chen et al., 2022) deploy an energy-guided network. They train on synthetic data to indicate the direction of the actual data distribution via the Langevin Markov chain Monte Carlo sampling method. However, since all of these techniques rely on GANs, they are susceptible to training instability of GANs. To address the issues with GAN-based methods, the authors of DP-MERF (Harder et al., 2021) synthesize images by taking advantage of random feature representations of kernel mean embeddings, while the authors of P3GM (Takagi et al., 2021) use a variant of the private variational autoencoder.

So far, we have discussed generative DP techniques; next, we discuss discriminative DP techniques. Private Aggregation of Teacher Ensembles (PATE, Papernot et al., 2017; 2018) divides the training data into disjoint sets and assigns them to multiple classifiers (teachers). The teachers are queried with either public or GAN-generated images to obtain the corresponding soft labels (Long et al., 2021; Jordon et al., 2019). The student network is trained using public or GAN-generated images and their soft labels using knowledge transfer (Hinton et al., 2015). To maintain the privacy of multiple teacher models, the soft labels of all the teachers are aggregated, and noise is added before they are released for training the student. When PATE employs a GAN for image generation, it encounters the aforementioned challenges related to GANs. When utilizing public images, it struggles to address situations where public and private images are not aligned. Thus, to address the challenges of generative and discriminative data release techniques, we propose a new framework, DP-ImgSyn, for synthetic data release.

4 DP-ImgSyn: Differentially Private Image Synthesis

This section introduces our approach's specifics, as Figure 3 illustrates. Our method consists of three steps: 1) train a teacher model with a DP training algorithm on the private images, 2) perform optimization on the

[2]Please note, DPGEN (Chen et al., 2022) privacy guarantees are compromised due to conceptual errors reported in Dockhorn et al. (2022).

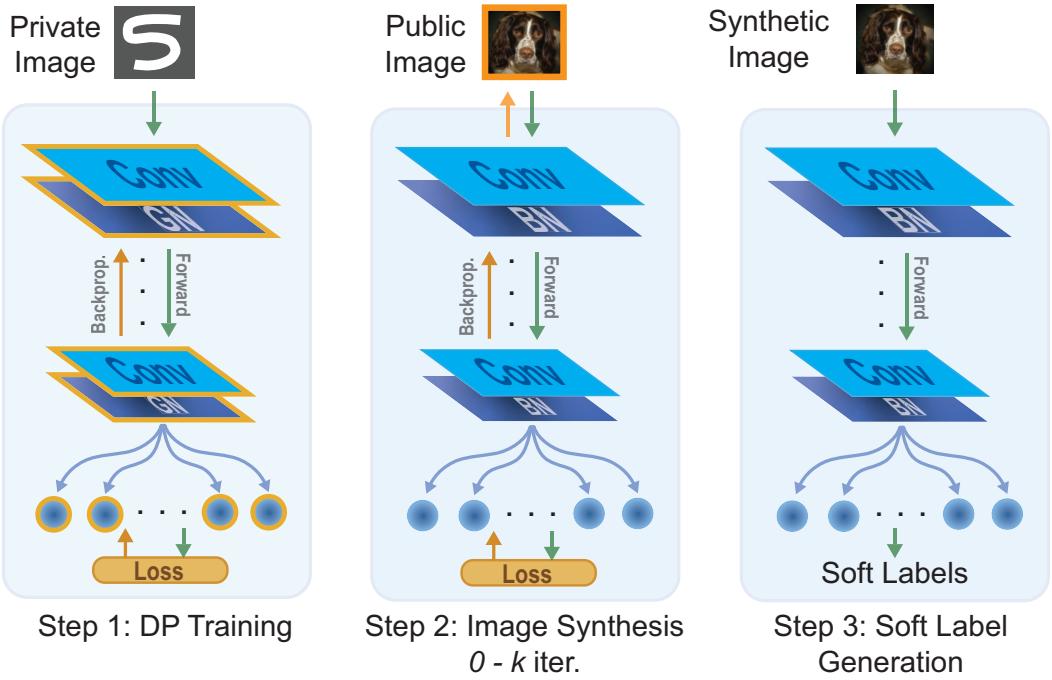


Figure 3: Overview of DP-ImgSyn: (1) Train a teacher model using a DP-training scheme. Capture the batch statistics of the model on the private dataset using the proposed DP guaranteed technique. (2) Perform the proposed public dataset alignment to obtain a better-aligned synthetic public dataset. Note that, the public image is optimized (updated), while the parameters of the model stay constant. (3) Generate soft labels. The synthetic images and their soft labels can be publicly released.

public dataset to align it to the private dataset and 3) generate soft labels for the aligned synthetic public dataset. Finally, the synthetic images and their corresponding soft labels are publicly released to train a student network.

4.1 DP Teacher Model Training

The privacy budget ϵ of our method is the combined privacy budget for the teacher model training ϵ_{train} and the statistics capture process $\epsilon_{batch-stats}$. Firstly, our method involves training the teacher model utilizing a DP training algorithm. Note that the teacher model is not trained with standard SGD but uses a DP training algorithm to ensure privacy. We select Differentially Private-Stochastic Gradient Descent (DP-SGD, Abadi et al., 2016) as the DP training algorithm. DP-SGD trains a neural network with (ϵ, δ)-DP guarantee. Similar to standard SGD, the algorithm converges in multiple training steps. At each training step t, DP-SGD computes the gradient $g_t(x)$ of the loss function with respect to the model parameters for a training image x. Then, it clips each gradient vector g to have a maximum l_2 norm of C. That is, the gradient vector g is replaced by $g/max(1, ||g||_2/C)$. The clipping ensures that if $||g||_2 \leq C$ then g is preserved, whereas if $||g||_2 > C$, it gets scaled down to be of norm C. Thus, the contribution of each data point to the batch gradient is bound by a constant C. Noise is added to the gradient $g_t(x) + \mathcal{N}(0, \sigma^2 C^2 I)$ and the descent step $\theta_{t+1} = \theta_t - \eta_t g_t$ is performed, with η_t learning rate. After T iterations, it outputs the $(\epsilon_{train}, \delta)$-DP teacher model.

Next, the synthesis (a.k.a dataset alignment) requires the teacher model's batch statistics. These are obtained from the batch norm layer of the teacher model. However, batch norm layers cannot be used for DP training, because the batch norm computes the mean over multiple training data points. Thus, per sample gradient cannot be obtained during training. This implies that the gradient norm cannot be bound, so we cannot provide a DP guarantee. To address this issue, we propose the following DP-guaranteed approach to obtain batch statistics. First, we use group norm layers instead of batch norm layers. Next, we train the teacher model using the previously described DP-SGD. Once trained, we capture the input to all group norm layers,

Algorithm 1: DP Image Synthesis

Input: DP-trained teacher model $\mathcal{M}$; k number of optimization iterations; synthesis learning rate γ_{syn}; batch statistics μ, σ for the private set; batch of public images x_P

Output: $\hat{x}$, one batch of aligned synthetic images

1 $\hat{x} \leftarrow x_P$
2 $y \leftarrow$ Target labels for batch $\hat{x}$, uniformly distributed over all the classes
3 **for** $i = 1, 2, ..., k$ **do**
4 $\quad \mu(\hat{x}), \sigma(\hat{x}) \leftarrow \mathcal{M}(\hat{x})$
5 $\quad \mathcal{R} \leftarrow \mathcal{R}_{total}(\hat{x}, y, \mu, \sigma, \mu(\hat{x}), \sigma(\hat{x}))$; `// Compute the loss from Equation 5`
6 $\quad \nabla_{\hat{x}}\mathcal{R} \leftarrow$ Backward pass
7 $\quad$ Update $\hat{x} \leftarrow \hat{x} - \gamma_{syn}\nabla_{\hat{x}}\mathcal{R}$
8 **return** $\hat{x}$

i_{gn}. We clip the input i_{gn} to have a maximum l_2 norm of C and add noise, $\hat{i}_{gn} = i_{gn} + \mathcal{N}(0, \sigma^2 C^2 I)$. The noisy input $\hat{i}_{gn}$ is used to calculate the batch statistics of the private set. This process has the same DP guarantee as the Gaussian Mechanism (Dwork et al., 2014) since it uses the Gaussian DP mechanism described in Dwork et al. (2014). Note that obtaining the batch statistics this way consumes some of the privacy budget allocated for training. Thus, when we report our results, the privacy budget ϵ is for the combined training and statistics capture process: $\epsilon = composition(\epsilon_{train}, \epsilon_{batch-stats})$. We use the accountant implementation from Gopi et al. (2021) to calculate the upper bounds on privacy. To ensure correctness, we also verify the upper bounds on privacy using both Opacus (Yousefpour et al., 2021) and Gopi et al. (2021)'s implementation.

4.2 DP Image Synthesis

Image synthesis (a.k.a dataset alignment) optimizes the public dataset to align with the private dataset to ensure that the synthesized (or aligned) images will have good distillation performance. To perform image synthesis, we collect the layerwise batch statistics of the private dataset using the DP-guaranteed technique described in Section 4.1. The batch statistics consists of the mean $\mu = [\mu_1, \cdots, \mu_L]$ and the variance $\sigma = [\sigma_1, \cdots, \sigma_L]$ from the all the L layers of the teacher model $\mathcal{M}$. Let $\hat{x}$ denote a batch of synthetic images and x_P denote a batch of data sampled from a public dataset D_p. Algorithm 1 summarizes the DP image synthesis process. We initialize $\hat{x}$ with x_P, this corresponds to Line 1 in Algorithm 1. For each data point in $\hat{x}$, we assign a target label y (Line 2 in Algorithm 1). The target labels are uniformly distributed over all the classes, and the assignment is such that we have the same number of images for each class. This step of generating the labels is independent of the private dataset to ensure privacy. For the exact implementation refer to Appendix A.1.3.

The next step is to perform k iterations of optimization corresponding to Lines 3 - 7. These k iterations optimize $\hat{x}$ to align with the private set. Each iteration consists of a forward pass of $\hat{x}$ through the DP-trained teacher model $\mathcal{M}$ (Line 4 in Algorithm 1). The forward pass is used to obtain the layer-wise batch statistics for $\hat{x}$, $(\mu(\hat{x}), \sigma(\hat{x}))$. The batch statistics and the generated label y are used to calculate the loss described in Equation 5 (Line 5 in Algorithm 1). The gradient of the loss $\mathcal{R}$ with respect to $\hat{x}$ ($\nabla_{\hat{x}}\mathcal{R}$) is calculated using back-propagation and is used to update the image $\hat{x}$ (Lines 6 - 7 in Algorithm 1). At the end of k update steps, we have synthesized one batch of aligned images $\hat{x}$. Since the teacher model is DP-trained, and the private dataset batch statistics are obtained with a DP guarantee, image synthesis is also DP-guaranteed.

The loss used to guide the optimization is critical. The total loss $\mathcal{R}_{total}$ consists of the following terms: feature loss $\mathcal{R}_{feature}$, classification loss $\mathcal{R}_{classif}$, total variance loss $\mathcal{R}_{tv}$ and l_2 norm loss $\mathcal{R}_{l_2}$. The sum of total variance loss $\mathcal{R}_{tv}$ and the l_2 norm loss $\mathcal{R}_{l_2}$ are referred to as prior loss. Next, we define each of these losses. The feature loss $\mathcal{R}_{feature}$ computes the distance between the batch statistics of the private dataset and the synthetic set $\hat{x}$ and is given by the following equation:

$$\mathcal{R}_{feature}(\mu, \sigma, \mu(\hat{x}), \sigma(\hat{x})) = \sum_{l=1}^{L} ||\mu_l(\hat{x}) - \mu_l||_2^2 + ||\sigma_l(\hat{x}) - \sigma_l)||_2^2 \tag{1}$$

Where $\hat{x}$ is the synthesized-aligned image, $\mu_l(\hat{x})$ and $\sigma_l(\hat{x})$ are the batch-wise mean and variance estimates of feature maps corresponding to the lth layer when $\hat{x}$ is fed to $\mathcal{M}$, and μ_l and σ_l are the lth layer batch statistics obtained from the private dataset described in Section 4.1. The classification loss $\mathcal{R}_{classif}$ is the cross-entropy loss between the teacher output and the target label y and is defined as:

$$\mathcal{R}_{classif}(\hat{x}, y) = \mathcal{L}(p_{\mathcal{M}}(\hat{x}), y) \tag{2}$$

where $\mathcal{L}$ is the cross-entropy loss, $p_{\mathcal{M}}(\hat{x})$ is the output of the teacher model $\mathcal{M}$ when $\hat{x}$ is fed as input, and y is the target label. The total variance loss $\mathcal{R}_{tv}$ ensures no sharp transitions in the synthetic image and restricts the adjacent pixels to have similar values. It is defined as:

$$\mathcal{R}_{tv}(\hat{x}) = \sum_{i,j}((\hat{x}_{i,j+1} - \hat{x}_{i,j})^2 + (\hat{x}_{i+1,j} - \hat{x}_{i,j})^2)^{\frac{1}{2}} \tag{3}$$

The l_2 norm loss is employed to encourage the image range to remain within a target interval rather than diverging. The l_2 norm loss $\mathcal{R}_{l_2}$ for the $\hat{x}$ is defined as:

$$\mathcal{R}_{l_2}(\hat{x}) = ||\hat{x}||_2^2 \tag{4}$$

The total loss is the sum of the aforementioned losses:

$$\mathcal{R}_{total}(\hat{x}, y, \mu, \sigma) = \alpha_f \mathcal{R}_{feature}(\hat{x}, \mu, \sigma) + \alpha_c \mathcal{R}_{classif}(\hat{x}, y) + \alpha_{tv} \mathcal{R}_{tv}(\hat{x}) + \alpha_{l_2} \mathcal{R}_{l_2}(\hat{x}) \tag{5}$$

Each loss term is multiplied by a corresponding scaling factor $\alpha_f, \alpha_c, \alpha_{tv}, \alpha_{l_2}$. The teacher model is not updated during back-propagation, and only $\hat{x}$ is optimized. After k iterations, we obtain the synthetic DP images $\hat{x}$.

4.3 DP Image Release

After synthesizing the images, we obtain their corresponding soft labels. The synthetic images are fed to the DP-trained teacher model $\mathcal{M}$, and the corresponding soft labels $\hat{y} = p_{\mathcal{M}}(\hat{x})$ are recorded. Because the teacher model is trained with DP-SGD, querying the teacher to obtain the soft labels does not impose privacy risk. The synthetic images $\hat{x}$, along with their soft labels $\hat{y}$, are publicly released. The student model $\mathcal{S}$ is trained on the synthetic images $\hat{x}$ and their corresponding soft labels $\hat{y}$ using KL-divergence:

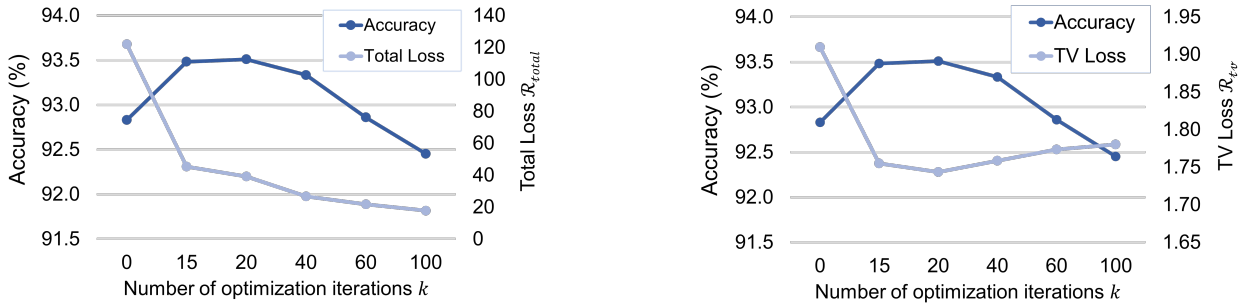
$$\min_{\theta} \sum_{x \in \mathcal{X}^s} KL(\hat{y}, p_{\mathcal{S}}(\hat{x})/T) \tag{6}$$

KL refers to the Kullback-Leibler divergence, $p_{\mathcal{S}}(\hat{x})$ is the output (soft labels) of the student model when the synthetic image $\hat{x}$ is given as input. T is a scaling temperature value. The synthetic images and their corresponding soft labels are publicly released and can be used to train any neural network. Note that, after synthetic image generation, the public images x_P are no longer needed. Thus, the public images will not be needed during the network training, as the network is trained using synthetic images only.

5 Experimental Evaluation

5.1 Experimental Setup

To evaluate our proposal, we use the same vision datasets as previous works; specifically, we use MNIST (LeCun et al., 1998), FashionMNIST (Xiao et al., 2017), CIFAR-10 (Krizhevsky et al., 2009), ImageNette (ima, 2018), CelebA-Hair (Long et al., 2021; Liu et al., 2015), CelebA-Gender (Long et al., 2021; Liu et al., 2015), TinyImageNet (Li et al., 2015), Places365 (Zhou et al., 2017), LSUN (Yu et al., 2015), and Textures (Cimpoi et al., 2014). We use the following network architectures: ResNet18 (He et al., 2016), ResNet34 (He et al., 2016), VGG11 (Simonyan & Zisserman, 2014), MobileNetV2 (Sandler et al., 2018), and ShuffleNetV2 (Ma et al., 2018). All the models were trained till convergence or privacy budget exhaustion. The detailed hyper-parameter settings for image synthesis and model training, computational resources, and dataset statistics are reported in the Appendix. We perform the experiments described in the following sections to evaluate our proposal thoroughly.



(a) Accuracy and $\mathcal{R}_{total}$ versus k.

(b) Accuracy and $\mathcal{R}_{tv}$ versus k.

Figure 4: Plots for student model accuracy, total loss $\mathcal{R}_{total}$ and total variance loss $\mathcal{R}_{tv}$ versus the number of optimization iterations k. The left axis is for accuracy, and the right is for loss $\mathcal{R}_{total}$ and $\mathcal{R}_{tv}$, respectively. Results suggest early stopping is necessary to optimize accuracy. For this plot, DP-ImgSyn is initialized with the TinyImageNet dataset, and the ResNet34 is the teacher model.

5.2 Number of Optimization Iterations k

This section studies the effect of the number of optimization iterations on image synthesis and performance.

Experiment. We train a ResNet34 till convergence on the private dataset CIFAR-10. Next, we perform the alignment optimization detailed in Section 4.2 for various numbers of iterations (i.e., k in DP-ImgSyn Algorithm 1) ranging from 0-100. For the alignment, we use TinyImageNet as the public dataset. The DP-ImgSyn generated dataset is used to train a ResNet18 student model. The student model accuracy and losses versus the number of optimization iterations k are reported. We use privacy budget $\epsilon = \infty$ to isolate all variables. The results are visualized in Figure 4.

Results. Figure 4a illustrates the accuracy of the student model and the total loss $\mathcal{R}_{total}$ versus the number of optimization iterations k. This is visualized with two y-axes: the left axis for accuracy and the right axis for the total loss $\mathcal{R}_{total}$. Note that the student model accuracy peaks around $k = 20$ iterations. Continuing the optimization by increasing k reduces student model performance. The plot in Figure 4b explains the reason for this behavior. Figure 4b plots the accuracy of the student model and the total variance loss $\mathcal{R}_{tv}$ versus the number of optimization iterations k. This is visualized with two y-axes, left for accuracy and the right for total variance loss $\mathcal{R}_{tv}$. The loss $\mathcal{R}_{tv}$ expresses the image prior. The image prior ensures no sharp transitions in the synthetic image. It is used in literature (Dosovitskiy & Brox, 2016; Mahendran & Vedaldi, 2015; Nguyen et al., 2015; Simonyan et al., 2014) as a proxy for how natural synthesized images are. This plot suggests that the images become more artificial or synthesized as we optimize past a certain threshold. It is observed that around 20 iterations, the $\mathcal{R}_{tv}$ loss starts increasing, and the accuracy starts decreasing.

Conclusion. Minimizing $\mathcal{R}_{total}$ does not guarantee optimal image prior ($\sim \mathcal{R}_{tv}$). However, $\mathcal{R}_{tv}$ significantly impacts student model accuracy. Thus, early stopping is necessary to optimize student model performance.

5.3 Privacy and Performance

We evaluate the performance of DP-ImgSyn with 0 iterations and k iterations on various vision datasets as public and private datasets.

Experiment. First, we select a private dataset. We train a teacher ResNet18 model on the private dataset using DP-SGD and capture batch statistics as described in Section 4.1. This results in a ResNet18 teacher model with batch statistics having a privacy budget of ϵ. Next, we select a public dataset and generate the synthetic dataset described in Sections 4.2 and 4.3. A student ResNet18 model is trained on the synthesized dataset with various public datasets as initialization. We report the accuracy results for privacy budgets $\epsilon \in \{1, 10\}$ on MNIST, FashionMNIST, CelebA-Hair, and CelebA-Gender as private datasets. We select public datasets whose size is the same or exceeds the size of the private dataset. This ensures that the synthesized dataset has the same size as the private dataset. When the public dataset is larger than the

Table 1: Accuracy for ResNet18 DP-teacher model with $\epsilon \in \{1, 10\}$ for MNIST, FashionMNIST, CelebA-Hair, and CelebA-Gender.

	MNIST	FashionMNIST	CelebA-Hair	CelebA-Gender
$\epsilon = 1$	86.87%	76.27%	79.95%	91.02%
$\epsilon = 10$	96.30%	81.88%	81.74%	92.35%

private set, we randomly sample from the public set to obtain a subset of the same size as the private set. This public subset is used for the synthesis. The number of classes of the public dataset is not required to be the same as the private set, because we use the soft labels of the teacher model. Note that, there is *no* one-to-one correspondence between public and private set images. Appendix A.1.4 reports an ablation study of the various loss terms.

Results. Table 1 reports the accuracy of the ResNet18 DP-teacher model trained on various datasets. Table 2 summarizes the accuracy results for MNIST, FashionMNIST, CelebA-Hair, and CelebA-Gender as private datasets and TinyImageNet, Places365, FashionMNIST, MNIST, and LSUN as public datasets. Table 2 reports the performance of the student model when performing 0 iterations, reported as DP-ImgSyn(0). The student model is trained on the DP-ImgSyn generated images and the test set is the private test set. We report the performance with early stopping at k_{exp} iterations as DP-ImgSyn(k_{exp}) and the optimal iterations to stop as k_{opt}.

Conclusion. From Table 2, we observe that, on average, when the private datasets are aligned over various public datasets, DP-ImgSyn(k) performs similar to DP-ImgSyn(0). For some datasets, we observe $k_{opt} = 0$, i.e., the proposed alignment process does not improve performance much. However, when the datasets are misaligned, like in the case of FashionMNIST and MNIST, we see DP-ImgSyn(k) performs significantly better (**≈17%** improvement in student model accuracy, for FashionMNIST private dataset with MNIST public dataset initialization for $\epsilon \in \{1, 10\}$). Moreover, in Table 2 we see that for each private dataset, we use multiple public datasets as initialization and they result in similar accuracy. Thus, the impact of the domain gap (accuracy) between public and private images is minimal. For more details about the interference of the private and public images refer to Appendix A.4.

5.4 Comparison with other Techniques

Experiment. This section compares our proposal with previous approaches: DP-GAN (Xie et al., 2018), DP-MERF (Harder et al., 2021), P3GM (Takagi et al., 2021), DataLens (Wang et al., 2021) and G-PATE (Jordon et al., 2019). For a fair comparison, we compare with techniques that use similar DP-training schemes (i.e., variants of DP-SGD). Note, that we exclude comparison with DPGEN (Chen et al., 2022) because privacy guarantees are compromised due to errors as reported in Dockhorn et al. (2022). The results for other techniques are the best accuracy results reported in prior publications. For our results, we report our best performance results from Table 2. Specifically, for $\epsilon = 1$, we use as public dataset initialization FashionMNIST, MNIST, Places, and Places for the private datasets MNIST, FashionMNIST, CelebA-Hair, and CelebA-Gender, respectively. For $\epsilon = 10$, we use as public dataset initialization TinyImageNet, TinyImageNet, LSUN, and Places for the private datasets MNIST, FashionMNIST, CelebA-Hair, and CelebA-Gender, respectively. Regarding DP-ImgSyn initialization, we could consider the use of GAN-generated images to initialize the DP-ImgSyn. However, doing this would violate the property of visual dissimilarity between the synthetic and the private images, since GAN-generated images are visually similar to the private images. Therefore, in this paper, we only consider the use of public images and noise (see Appendix A.2 for a more thorough discussion on noise) for the DP-ImgSyn initialization.

Results. Table 3 compares the performance of the proposed technique with state-of-the-art techniques.

Conclusion. We observe that our proposed method significantly outperforms generative techniques that use similar DP training schemes (up to ≈ **20%**, for both CelebA-Hair and CelebA-Gender for $\epsilon = 1$). We also provide results when using stronger privacy $\epsilon = 0.2$ in Appendix A.5 for interested readers, where we see similar results.

Table 2: Comparative Table with $\epsilon \in \{1, 10\}$ for MNIST, FashionMNIST, CelebA-Hair, and CelebA-Gender as private datasets using TinyImageNet, Places365, FashionMNIST, MNIST, and LSUN as public datasets. Results are mean ± std over three different seeds. The models are trained on the synthetic images generated by DP-ImgSyn (training set) and evaluated on the test set of the private dataset (testing set).

Private Dataset	ϵ	Public Dataset	DP-ImgSyn(0)	DP-ImgSyn(k_{exp})	k_{exp}	k_{opt}
MNIST	$\epsilon = 1$	TinyImageNet	85.83 ± 0.13	**85.98 ± 0.06**	10	10
		Places365	85.00 ± 0.30	**86.01 ± 0.22**	10	10
		FashionMNIST	85.56 ± 0.32	**86.24 ± 0.03**	10	10
	$\epsilon = 10$	TinyImageNet	92.97 ± 0.65	**94.03 ± 0.64**	10	10
		Places365	92.63 ± 0.23	**93.74 ± 0.18**	10	10
		FashionMNIST	93.61 ± 0.37	**93.90 ± 0.30**	10	10
FashionMNIST	$\epsilon = 1$	TinyImageNet	74.99 ± 0.21	74.93 ± 0.02	1	0
		Places365	75.08 ± 0.15	74.94 ± 0.20	1	0
		MNIST	51.58 ± 2.28	**68.38 ± 0.34**	10	10
	$\epsilon = 10$	TinyImageNet	79.04 ± 0.04	78.71 ± 0.20	1	0
		Places365	78.73 ± 0.14	**78.80 ± 0.05**	1	1
		MNIST	54.78 ± 1.15	**71.51 ± 1.49**	10	10
CelebA-Hair	$\epsilon = 1$	LSUN	79.89 ± 0.08	79.42 ± 0.14	1	0
		Places365	79.91 ± 0.08	79.50 ± 0.15	1	0
	$\epsilon = 10$	LSUN	81.31 ± 0.04	79.28 ± 0.20	1	0
		Places365	81.33 ± 0.12	78.73 ± 0.44	1	0
CelebA-Gender	$\epsilon = 1$	LSUN	89.91 ± 0.15	89.17 ± 0.26	1	0
		Places365	90.06 ± 0.05	89.03 ± 0.19	1	0
	$\epsilon = 10$	LSUN	90.99 ± 0.16	89.90 ± 0.26	1	0
		Places365	91.22 ± 0.04	89.27 ± 0.99	1	0

Table 3: Comparison Table with state-of-the-art techniques for $\epsilon \in \{1, 10\}$ for MNIST, FashionMNIST, CelebA-Hair, and CelebA-Gender. Results for DP-ImgSyn are mean over three different seeds. The best-performing framework is highlighted in bold, and the second-best is underlined. DP-GAN refers to Xie et al. (2018), DP-MERF refers to Harder et al. (2021), P3GM refers to Takagi et al. (2021), DataLens refers to Wang et al. (2021) and G-PATE refers to Long et al. (2021).

Dataset	ϵ	DP-GAN	DP-MERF	P3GM	DataLens	G-PATE	DP-ImgSyn (ours)
MNIST	1	40.36%	63.67%	<u>73.69%</u>	71.23%	58.80%	**86.24%**
	10	80.11%	67.38%	79.81%	80.88%	<u>80.92%</u>	**94.03%**
FashionMNIST	1	10.53%	58.62%	<u>72.23%</u>	64.78%	58.12%	**75.08%**
	10	60.98%	61.62%	<u>74.80%</u>	70.61%	69.34%	**79.04%**
CelebA-Hair	1	34.47%	44.13%	45.32%	<u>60.61%</u>	49.85%	**79.91%**
	10	39.20%	52.25%	44.89%	<u>62.24%</u>	62.17%	**81.33%**
CelebA-Gender	1	53.30%	59.36%	56.73%	<u>69.96%</u>	67.02%	**90.06%**
	10	52.11%	60.82%	58.84%	<u>72.87%</u>	68.97%	**91.22%**

5.5 Beyond Generative Methods

This section presents results on higher resolution (224 x 224) and more varied datasets with which generative techniques often have difficulty.

Table 4: Comparative Table with $\epsilon \in \{1, 10\}$ for MNIST, and CelebA-Gender as private datasets using TinyImageNet, Places365, FashionMNIST, and LSUN and Places365 as public datasets respectively. Results are mean ± std over three different seeds. The models are trained on the synthetic images generated by DP-ImgSyn (training set) using a ResNet18 as teacher model. The student model architecture is VGG11, MobileNetV2, ShuffleNetV2, and ResNet18. The student models are evaluated on the test set of the private dataset (testing set).

			Student Model Architecture			
Private Dataset	ϵ	Public Dataset	VGG11	MobileNetV2	ShuffleNetV2	ResNet18
MNIST	$\epsilon = 1$	TinyImageNet	85.56 ± 0.28	84.34 ± 0.70	85.15 ± 0.16	85.98 ± 0.06
		Places365	85.65 ± 0.48	84.02 ± 0.33	85.28 ± 0.22	86.01 ± 0.22
		FashionMNIST	86.37 ± 0.42	84.95 ± 0.28	85.91 ± 0.06	86.24 ± 0.03
	$\epsilon = 10$	TinyImageNet	94.57 ± 0.25	92.08 ± 0.62	94.04 ± 0.09	94.03 ± 0.64
		Places365	93.91 ± 0.26	91.75 ± 0.74	93.22 ± 0.24	93.74 ± 0.18
		FashionMNIST	94.70 ± 0.18	92.66 ± 0.95	92.91 ± 0.91	93.90 ± 0.30
CelebA-Gender	$\epsilon = 1$	LSUN	88.32 ± 0.32	88.31 ± 0.30	89.44 ± 0.17	89.17 ± 0.26
		Places365	88.61 ± 0.38	88.49 ± 0.14	89.16 ± 0.44	89.03 ± 0.19
	$\epsilon = 10$	LSUN	89.36 ± 0.55	88.85 ± 0.25	89.54 ± 0.28	89.90 ± 0.26
		Places365	89.40 ± 0.50	87.85 ± 0.26	89.02 ± 0.02	89.27 ± 0.99

Experiment. We consider ImageNette and CIFAR-10 as private datasets and we use the proposed DP-ImgSyn technique. For ImageNette, we use a resolution of 224 × 224 with $\epsilon = 105$, initialized with Textures as public dataset; for CIFAR-10, we select $\epsilon = 10$, initialized with TinyImageNet as public dataset. Note that generative methods do not report results on these datasets.

Results. A ResNet18 student model trained on the DP-ImgSyn generated dataset for ImageNette (224 × 224) achieved **39.38%** accuracy on the test set, while the teacher model achieves 43.26% accuracy. Similarly, for CIFAR-10, a ResNet18 student trained on DP-ImgSyn generated dataset achieved **45.66%** accuracy on the test set, while the teacher model achieves 42.95% accuracy.

Conclusion. The ImageNette results suggest that DP-ImgSyn is not limited by the image resolution, unlike existing generative techniques. ImageNette and CIFAR-10 results indicate that the DP-ImgSyn technique is better suited to more complex and higher-resolution datasets when compared to generative techniques.

5.6 Generalization to Other Network Architectures

This section presents results with various network architectures for the student network to evaluate whether our released dataset can be used to train any network architecture.

Experiment. We use DP-ImgSyn to generate the synthetic images using a ResNet18 as the teacher model. The student model architecture is VGG11 (Simonyan & Zisserman, 2014), MobileNetV2 (Sandler et al., 2018), ShuffleNetV2 (Ma et al., 2018), and ResNet18 (He et al., 2016). For $\epsilon = 1, 10$ we use MNIST and CelebA Gender as the private dataset initialized with TinyImageNet, Places365, FashionMNIST, and LSUN, Places365 respectively.

Results. Table 4 summarizes the results across the various student architectures. We observe that the accuracy is similar across the architectures (VGG11, MobileNetV2, ShuffleNetV2), and similar to ResNet18 which has the same architecture as the teacher model.

Conclusion. The outcome of this experiment is that the synthetic images can be used to train any student network architecture. Thus, DP-ImgSyn is not restricted by the teacher model architecture.

6 Conclusion

Deep neural networks are state-of-the-art solutions for various tasks in multiple domains, but they require a significant amount of data for training. However, certain data cannot be publicly released due to privacy restrictions. This paper presents a discriminative approach (DP-ImgSyn) for releasing synthetic images that have DP guarantees, maintain the utility of the private images, and are visually dissimilar to the private images. The proposed framework leverages dataset alignment to obfuscate private sensitive images in public images. This alignment/synthesis process improves the performance of even highly misaligned public-private dataset pairs. We observe $\approx$ **17%** improvement in the performance of highly misaligned datasets. Also, we show that the non-generative DP-ImgSyn approach significantly outperforms (up to $\approx$ **20%** improvement in classification accuracy) generative techniques using similar DP-training schemes. Further, we present results on higher resolution (224 x 224) and more varied datasets with which generative techniques often have difficulty. Our findings suggest discriminative (a.k.a non-generative) approaches might better suit synthetic dataset release. However, further research is needed to identify the limits of discriminative and generative techniques.

7 Acknowledgments

This work was supported in part by, the Center for the Co-Design of Cognitive Systems (COCOSYS), a DARPA-sponsored JUMP2.0 center, the Semiconductor Research Corporation (SRC), the National Science Foundation, and DARPA ShELL.

References

Imagenette - a subset of 10 easily classified classes from the imagenet dataset. `https://github.com/fastai/imagenette`, 2018.

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Tariq O Abbas, Mohamed AbdelMoniem, and Muhammad EH Chowdhury. Automated quantification of penile curvature using artificial intelligence. *Frontiers in Artificial Intelligence*, 5:954497, 2022.

Accountability Act. Health insurance portability and accountability act of 1996. *Public law*, 104:191, 1996.

Fatih Cagatay Akyon and Alptekin Temizel. Deep architectures for content moderation and movie content rating. *arXiv preprint arXiv:2212.04533*, 2022.

Martin Arjovsky and Leon Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=Hk4_qw5xe`.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

James Bennett, Stan Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, pp. 35. New York, 2007.

Alex Bie, Gautam Kamath, and Vikrant Singhal. Private estimation with public data. *Advances in Neural Information Processing Systems*, 35:18653–18666, 2022.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=B1xsqj09Fm`.

Tianshi Cao, Alex Bie, Arash Vahdat, Sanja Fidler, and Karsten Kreis. Don't generate me: Training differentially private generative models with sinkhorn divergence. *Advances in Neural Information Processing Systems*, 34:12480–12492, 2021.

Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. Gs-wgan: A gradient-sanitized approach for learning differentially private generators. *Advances in Neural Information Processing Systems*, 33:12673–12684, 2020.

Jia-Wei Chen, Chia-Mu Yu, Ching-Chia Kao, Tzai-Wei Pang, and Chun-Shien Lu. Dpgen: Differentially private generative energy-guided network for natural image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8387–8396, 2022.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Jing Deng, MARGARET A HALL-CRAGGS, D Pellerin, Alfred D Linney, William R Lees, Charles H Rodeck, and ANDREW TODD-POKROPEK. Real-time three-dimensional ultrasound visualization of erection and artificial coitus. *international journal of andrology*, 29(2):374–379, 2006.

Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. Differentially private diffusion models. *arXiv preprint arXiv:2210.09929*, 2022.

Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4829–4837, 2016.

Judson C Dressler, Christopher Bronk, and Daniel S Wallach. Exploiting military opsec through open-source vulnerabilities. In *MILCOM 2015-2015 IEEE Military Communications Conference*, pp. 450–458. IEEE, 2015.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pp. 265–284. Springer, 2006.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

L eon Bottou. Online learning and stochastic approximations. *On-linelearning in neural networks*, 17(9): 142, 1998.

Arun Ganesh, Mahdi Haghifam, Milad Nasr, Sewoong Oh, Thomas Steinke, Om Thakkar, Abhradeep Guha Thakurta, and Lun Wang. Why is public pretraining necessary for private model training? In *International Conference on Machine Learning*, pp. 10611–10627. PMLR, 2023.

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

Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. *Advances in Neural Information Processing Systems*, 34:11631–11642, 2021.

Richard A Haddad, Ali N Akansu, et al. A class of fast gaussian binomial filters for speech and image processing. *IEEE Transactions on Signal Processing*, 39(3):723–727, 1991.

Frederik Harder, Kamil Adamczewski, and Mijung Park. 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. PMLR, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

Fei-Fei Li, Andrej Karpathy, and Justin Johnson. Tiny imagenet visual recognition challenge. `http://cs231n.stanford.edu/tiny-imagenet-200.zip`, 2015.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Yunhui Long, Boxin Wang, Zhuolin Yang, Bhavya Kailkhura, Aston Zhang, Carl Gunter, and Bo Li. G-pate: scalable differentially private data generator via private aggregation of teacher discriminators. *Advances in Neural Information Processing Systems*, 34:2965–2977, 2021.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5188–5196, 2015.

J Manley and A Cavoukian. The personal information protection and electronic documents act (pipeda). *Priv. gc. ca*, 2000.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=B1QRgziT-`.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, 2015.

Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HkwoSDPgg.

Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Ulfar Erlingsson. Scalable private learning with PATE. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkZB1XbRZ.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

General Data Protection Regulation. General data protection regulation (gdpr). *Intersoft Consulting, Accessed in October*, 24(1), 2018.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Tim Salimans, Han Zhang, Alec Radford, and Dimitris Metaxas. Improving GANs using optimal transport. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkQkBnJAb.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ICLR, 2014.

Ruth Spence, Antonia Bifulco, Paula Bradbury, Elena Martellozzo, and Jeffrey DeMarco. The psychological impacts of content moderation on content moderators: A qualitative study. *Cyberpsychology: Journal of Psychosocial Research on Cyberspace*, 17(4), 2023.

Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. *Advances in neural information processing systems*, 30, 2017.

Shun Takagi, Tsubasa Takahashi, Yang Cao, and Masatoshi Yoshikawa. P3gm: Private high-dimensional data release via privacy preserving phased generative model. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 169–180. IEEE, 2021.

Aleksei Triastcyn and Boi Faltings. Generating artificial data for private deep learning. *arXiv preprint arXiv:1803.03148*, 2018.

Boxin Wang, Fan Wu, Yunhui Long, Luka Rimanic, Ce Zhang, and Bo Li. Datalens: Scalable privacy preserving training via gradient compression and aggregation. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2146–2168, 2021.

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

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

Chugui Xu, Ju Ren, Deyu Zhang, Yaoxue Zhang, Zhan Qin, and Kui Ren. Ganobfuscator: Mitigating information leakage under gan via differential privacy. *IEEE Transactions on Information Forensics and Security*, 14(9):2358–2371, 2019.

Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. Opacus: User-friendly differential privacy library in pytorch. In *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021. URL `https://openreview.net/forum?id=EopKEYBoI-`.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Xinyang Zhang, Shouling Ji, and Ting Wang. Differentially private releasing via deep generative model (technical report). *arXiv preprint arXiv:1801.01594*, 2018.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

A Appendix

A.1 Training Details

DP-ImgSyn implementation uses the Pytorch (Paszke et al., 2019) framework, and the experiments were conducted on NVIDIA GeForce GTX 1080 Ti with 11 GB of memory with the Ubuntu operating system.

Table 5: Hyperparameters used for DP training and batch statistics capture experiments on the vision datasets: MNIST, FashionMNIST, CelebA-Hair, CelebA-Gender, CIFAR-10, and ImageNette. The ϵ denotes the privacy budget, η_{tr} is the number of training epochs, Ω_{tr} is the batch size used for DP-SGD training, γ_{tr} is learning rate used for training, C denotes the maximum norm limit for the gradient vector g $(g/max(1, ||g||_2/C))$, σ controls the amount of noise added to g, η_{bn} is the number of epochs used for capturing batch statistics, and Ω_{bn} is the batch size used for capturing batch statistics

Dataset	ϵ	η_{tr}	Ω_{tr}	γ_{tr}	C	σ	η_{bn}	Ω_{bn}
MNIST	1	4	128	0.01	1.0	0.8	2	64
	10	14	128	0.01	1.0	0.5	2	64
FashionMNIST	1	30	50	0.01	1.2	1	2	64
	10	20	128	0.01	1.2	0.5	2	64
CelebA Hair	1	18	128	0.001	1.0	0.8	5	128
	10	22	128	0.001	1.0	0.45	3	128
CelebA Gender	1	18	128	0.001	1.0	0.8	4	128
	10	22	128	0.001	1.0	0.5	4	128
CIFAR-10	10	12	128	0.001	1.0	0.5	5	128
ImageNette	105	57	8	0.001	1.0	0.3	3	8

A.1.1 DP Teacher Model Hyperparameters

For the DP statistics capture for the teacher model (Section 4.1 from the main paper), we used the hyperparameters reported in Table 5. The ϵ denotes the privacy budget, η_{tr} is the number of training epochs, Ω_{tr} is the batch size used for DP-SGD training, γ_{tr} is learning rate used for training, C denotes the maximum norm limit for the gradient vector g $(g/max(1, ||g||_2/C))$, σ controls the amount of noise added to g $(g + \mathcal{N}(0, \sigma^2 C^2 I))$, η_{bn} is the number of epochs used for capturing batch statistics, and Ω_{bn} is the batch size used for capturing batch statistics in Table 5. The accuracy results of the corresponding teacher models are reported in Table 1 in the main paper.

A.1.2 DP Image Synthesis Hyperparameters

For the DP Image Synthesis (described in Section 4.2, and Algorithm 1 in the main paper), we use the Adam optimizer (Kingma & Ba, 2014) with synthesis learning rate $\gamma_{syn} = 0.1$, betas $\beta_1 = 0.5$, $\beta_2 = 0.99$. For MNIST and FashionMNIST, we use a batch size of 80; for CelebA-Hair and CelebA-Gender, we use a batch size of 60. The main paper in Table 2 reports the number of optimization iterations for each dataset and privacy budget.

The total loss optimized during image synthesis is the summation of the following losses: $\mathcal{R}_{feature}$, $\mathcal{R}_{classif}$, $\mathcal{R}_{tv}$, and $\mathcal{R}_{l_2}$ (Equation 5 in the main paper). The scaling coefficients corresponding to each of these losses are denoted as α_f, α_c, α_{tv}, and α_{l_2}, respectively. Table 7 reports the values of the scaling coefficients used in our simulations. Furthermore, we perform an ablation study on the scaling coefficients that control the total loss on MNIST as the private dataset for epsilon 10, and TinyImageNet as the public dataset initialization. Specifically, we search the values for each loss scaling factor while keeping the remaining

Table 6: Ablation study on the scaling coefficients α_f, α_c, α_{tv}, and α_{l_2} for MNIST as the private dataset, $\epsilon = 10$, and TinyImageNet and Gaussian Noise as the public dataset initialization.

Loss	Dataset	Hyperparameter Value and Accuracy		
		$\alpha_f = 0.1$	$\alpha_f = 1.0$	$\alpha_f = 10.0$
$R_{feature}$	TinyImageNet	93.99 ± 0.30	94.56 ± 0.15	94.03 ± 0.64
	Gaussian Noise	89.55 ± 0.92	90.30 ± 1.43	88.53 ± 0.33
		$\alpha_c = 0.01$	$\alpha_c = 1.0$	$\alpha_c = 10.0$
$R_{classif}$	TinyImageNet	94.42 ± 0.31	94.03 ± 0.64	94.49 ± 0.04
	Gaussian Noise	90.99 ± 0.86	88.53 ± 0.33	90.81 ± 0.83
		α_{tv} = 2e-5	$\alpha_{tv} = 0.01$	$\alpha_{tv} = 10.0$
R_{tv}	TinyImageNet	94.03 ± 0.64	94.27 ± 0.42	94.32 ± 0.12
	Gaussian Noise	88.53 ± 0.33	89.63 ± 1.90	89.81 ± 2.16
		α_{l2} = 3e-8	$\alpha_{l2} = 0.0001$	$\alpha_{l2} = 0.01$
R_{l2}	TinyImageNet	94.03 ± 0.64	94.45 ± 0.10	94.53 ± 0.16
	Gaussian Noise	88.53 ± 0.33	90.93 ± 1.33	89.87 ± 1.01

Table 7: Values of scaling coefficients for the loss terms $\mathcal{R}_{feature}$, $\mathcal{R}_{classif}$, $\mathcal{R}_{tv}$, and $\mathcal{R}_{l_2}$ in Equation 5 in the main paper: α_f, α_c, α_{tv}, and α_{l_2}.

α_f	α_c	α_{tv}	α_{l_2}
10	1	2.5e-5	3e-8

scaling factors constant. Table 6 summarizes our results. We observe that our method is robust to the scaling factor hyperparameter selections. Then, we repeated the experiment using the same setup (MNIST as the private dataset with epsilon set to 10), but with Gaussian noise as initialization. The findings in Table 6 consistently affirm our earlier observation, highlighting the robustness of our method to different scaling factor hyperparameter selections.

A.1.3 DP-ImgSyn Label Generation Implementation

Here we provide the PyTorch code that we use for generating the targets for DP-ImgSyn, given the number of classes and batch size. Since the label generation algorithm is independent of the data, there is no privacy leakage.

```
def generate_labels(num_classes, batch_size):
  x = torch.arange(num_classes)
  targets = torch.squeeze(x.repeat(1, int(batch_size/num_classes)))
  return targets
```

A.1.4 Student Model Training on DP-ImgSyn Synthetic Images Hyperparameters

We use Stochastic Gradient Descent (SGD) optimizer (eon Bottou, 1998) with a learning rate $\eta = 0.1$, momentum 0.9, and weight decay 1e-4 for training a student model on the synthetic images. We use the multistep learning rate scheduler with $\gamma = 0.1$ and milestones at 120, 150, and 180 epochs. We train the models for 200 epochs with 256 as batch size.

Table 8: Ablation study on the loss terms for MNIST as the private dataset, $\epsilon = 10$, and TinyImageNet, Places365, FashionMNIST, and Gaussian Noise as initialization.

$\mathcal{R}_{feature}$	$\mathcal{R}_{classif}$	$\mathcal{R}_{tv}$	$\mathcal{R}_{l_2}$	TinyImageNet	Places365	FashionMNIST	Gaussian Noise
✓	✗	✗	✗	94.30 ± 0.25	93.74 ± 0.13	94.02 ± 0.09	90.07 ± 1.59
✓	✓	✗	✗	94.31 ± 0.09	93.73 ± 0.24	94.19 ± 0.25	89.42 ± 2.36
✓	✓	✓	✗	94.34 ± 0.31	93.74 ± 0.17	93.94 ± 0.28	89.82 ± 1.00
✓	✓	✓	✓	94.03 ± 0.64	93.74 ± 0.18	93.90 ± 0.30	88.53 ± 0.33
✗	✓	✓	✓	93.03 ± 0.63	93.32 ± 0.30	91.70 ± 2.24	63.48 ± 13.62

The optimization loss function is the KL-divergence between the soft labels of the teacher model and the soft labels of the student model, defined as:

$$\min_{\theta} \sum_{x \in \mathcal{X}^s} KL(p_{\mathcal{M}}(\hat{x}), p_{\mathcal{S}}(\hat{x})/T) \tag{7}$$

Where KL refers to the Kullback-Leibler divergence, $p_{\mathcal{M}}(\hat{x})$ and $p_{\mathcal{S}}(\hat{x})$ are the output distributions (soft labels) of the teacher and the student model respectively when the synthetic image $\hat{x}$ is given as input. The temperature value used in our simulations is $T = 100$ for MNIST and FashionMNIST and $T = 10$ for CelebA-Hair and CelebA-Gender.

A.2 Loss Term Ablation Study

In this section, we evaluate the effect of each loss term on the accuracy of the model: $\mathcal{R}_{feature}$, $\mathcal{R}_{classif}$, $\mathcal{R}_{tv}$, and $\mathcal{R}_{l_2}$. The loss term ablation study is summarized in Table 8. We present the results for MNIST as private set with epsilon=10 when using TinyImageNet, Places365, and FashionMNIST as public datasets using three different seeds. To get further insight, we also present results when using Gaussian Noise with mean 0 and standard deviation 1 as initialization for the synthetic images. We make the following observations: 1) from the Gaussian Noise results we see that the feature loss that uses the batch normalization statistics significantly affects the accuracy ($\approx 26\%$ accuracy drop when feature loss was excluded in the synthesis loss). 2) when all the loss terms were used this led to a lower standard deviation between the seeds and thus more stability between the runs. The significant effect of the feature loss can be attributed to the value of the scaling coefficient associated with it.

A.3 Visualizations of ImgSyn Using a Real Example

In this section, we present a real example of Figure 2. In Figure 5 we show how ImgSyn can better sample the latent space and improve the decision boundary of the student model. For this experiment, we generated 2D data by sampling from a Uniform distribution in the range $[0, 1]$. The labels were set to 0 or 1 based on the cubic $y = -1.4x^3 + 0.9x1^2 + 0.34$. That is the cubic formed the decision boundary, if $y > 0$ label $= 1$ and vice versa. The cubic was chosen to have a non-linear decision boundary. Since the data is 2D we can visualize the decision boundary in the input space. The top left in Figure 5 shows the training data, the two classes shown in green (class 1) and orange (class 2). The teacher model was trained on this classification problem. Please note no DP was used during training since we want to demonstrate ImgSyn. The network confidence is visualized as a heatmap. Blue is high confidence that the sample is in class 1 and yellow is high confidence that the sample is in class 2. The top right shows the same visualization for the student model when applying vanilla knowledge distillation (KD) between the teacher and the student model. It shows the decision boundary at the end of training and the data (illustrated in red) that was used for performing knowledge distillation. For this example, we used two Gaussian clusters centered at $(0.25, 0.25), (0.75, 0.75)$ with a deviation of 0.25 as the dataset to perform knowledge distillation from the teacher to the student. Vanilla KD is not able to transfer the decision boundary well from the teacher to the student. The bottom

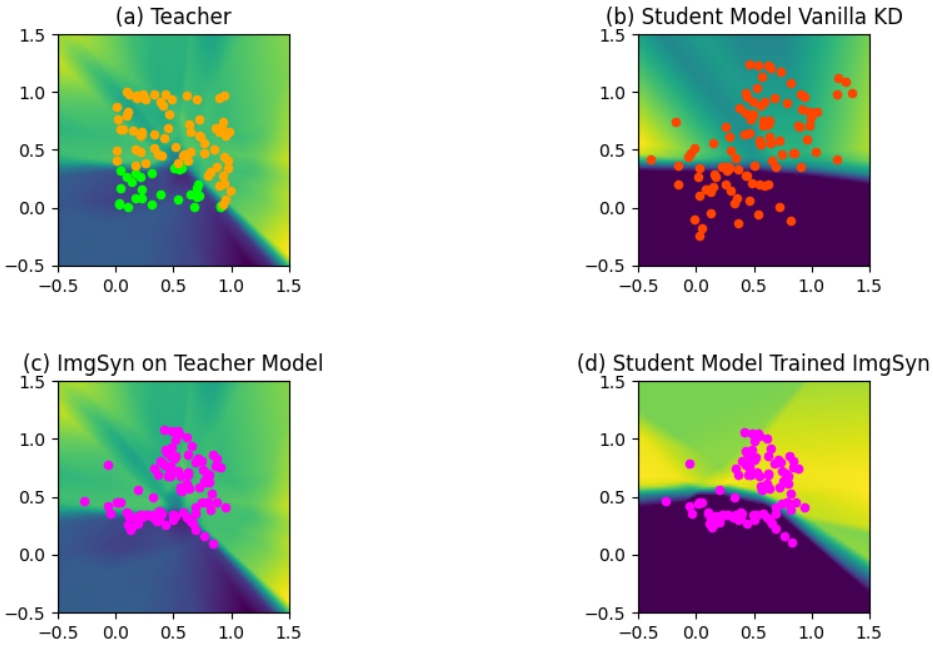


Figure 5: Example visualizing deep neural networks decision boundary when using ImgSyn and not using ImgSyn, a data-driven version of Figure 2: (a) visualization of the teacher decision boundary, and the training data belonging to two classes. The two classes are shown in green (class 1) and orange (class 2). The teacher network confidence is visualized as a heatmap. Blue illustrates high confidence that the sample is in class 1 and yellow is high confidence that the sample is in class 2, (b) visualization of the decision boundary for the student model when applying vanilla KD between the teacher and the student model, and the Gaussian data used for the vanilla KD (visualized in red) this is analogous to public data, (c) visualization of the teacher decision boundary and the ImgSyn generated data (magenta). The ImgSyn data are generated after applying our ImgSyn on the Gaussian data (illustrated in red color in image (b)) to align it with the source distribution, (d) visualization of the decision boundary of the student model trained on ImgSyn data, and the ImgSyn data (magenta) used for training the student model. The decision boundary of the student model trained on ImgSyn data has a better match to the teacher model than the student trained with on Gaussian data.

left shows the data after applying our ImgSyn on the same Gaussian dataset to align it with the source distribution, and the teacher decision boundary when passing the ImgSyn generated data. We see how the points move closer into the range of $[0, 1]$ to match the source distribution and on the bottom right we show the result of training a student model on the ImgSyn data. The decision boundary of the student model trained on ImgSyn data has a better match to the teacher model than the student trained with vanilla KD on Gaussian data.

A.4 Interference of the features between Public and Private Images

In this section, we quantify the interference of the public image features on the student model test accuracy. To evaluate this, we perform the following experiment: we initialize the synthetic images with Gaussian noise and then perform DP-ImgSyn, using MNIST as the private dataset and epsilon equal to 10. This will set the lower bound on the interference that the public dataset has on the optimization since the starting images are random. We evaluate the accuracy of the student model trained on the synthetic images initialized with Gaussian noise (Table 9). We observe that the accuracy drops about 3.4% for the DP-ImgSyn with Gaussian

Table 9: The interference of the features of the public images on the private images. The private dataset is MNIST, $\epsilon = 10$. The results for the datasets marked with * are from the main paper (Table 2), and are included in this table as a reference for comparison with the Gaussian Noise.

Private Dataset	ϵ	Initialization	DP-ImgSyn(0)	DP-ImgSyn(k)
MNIST	$\epsilon = 10$	Gaussian Noise	30.12 ± 5.08	**90.55 ± 1.67**
		TinyImageNet*	92.97 ± 0.65	**94.03 ± 0.64**
		Places365*	92.63 ± 0.23	**93.74 ± 0.18**
		FashionMNIST*	93.61 ± 0.37	**93.90 ± 0.30**

Table 10: Comparison Table with state-of-the-art techniques for $\epsilon = 0.2$ for MNIST and FashionMNIST and TinyImageNet as the public dataset initialization. Results for DP-ImgSyn are mean over three different seeds. DP-GAN refers to Xie et al. (2018), DP-MERF refers to Harder et al. (2021), P3GM refers to Takagi et al. (2021), DataLens refers to Wang et al. (2021) and G-PATE refers to Long et al. (2021).

	DP-GAN	DP-MERF	P3GM	DataLens	G-PATE	DP-ImgSyn (Our)
MNIST	11.04%	62.61%	8.20%	23.44%	22.30%	77.37%
FashionMNIST	10.21%	52.61%	12.80%	22.26%	18.74%	70.63%

noise initialization compared to best-performing public images. Even though there is interference from the public set, this effect is minimal compared to the effect of DP-ImgSyn. Without DP-ImgSyn, the student model trained using Gaussian noise achieves 30.12% accuracy on MNIST while with DP-ImgSyn it achieves 90.55% (Table 9).

A.5 Comparison With SOTA under Strong Privacy Guarantees

In this section, we present experiments with MNIST and FashionMNIST as private datasets with an epsilon value of 0.2 and TinyImageNet as the public dataset initialization. Table 10 summarizes the comparison results. Our method demonstrates better accuracy than the best-performing SOTA (DP-MERF) by 14.76% on MNIST and 18.02% on FashionMNIST.

A.6 Gaussian Noise Initialization with Low-pass Filtering

In this section, we present the results when Gaussian noise is used as initialization with MNIST as the private dataset and $\epsilon = 10$, and then we apply low pass filtering implemented as a Gaussian blur Haddad et al. (1991). The results for various kernel sizes are summarized in Table 11. Notably, low pass filtering improves the accuracy of the Gaussian noise initialization by up to 1.07% (results for kernel size 3). However, it is still lower ($\approx$ 2.4%) than the best-performing public dataset initialization. Therefore, random noise can as initialization when public images are unavailable, yielding satisfactory performance albeit with some accuracy degradation.

A.7 More Visualizations

Figure 6 provides illustrations for MNIST, FashionMNIST, and CelebA Gender as private images, the DP-ImgSyn generated images and the corresponding public images and Gaussian Noise that were used as initialization for DP-ImgSyn.

A.8 Dataset Statistics

Table 12 summarizes the statistics for the vision datasets used for the experimental evaluation in Section 5 in the main paper: MNIST, FashionMNIST, CIFAR-10, Imagenette, CelebA-Hair, CelebA-Gender, Tiny-

Table 11: Gaussian Noise with low-pass filtering as initialization with MNIST as the private dataset and $\epsilon = 10$. The low pass filtering is implemented as a Gaussian blur Haddad et al. (1991) for various kernel sizes.

Without Low Pass Filtering	Low-pass Filtering			
	Kernel Size = 3	Kernel Size = 7	Kernel Size = 11	Kernel Size = 15
90.55 ± 1.67	91.62± 1.35	90.82 ± 0.96	91.10 ± 0.41	91.18 ± 1.23

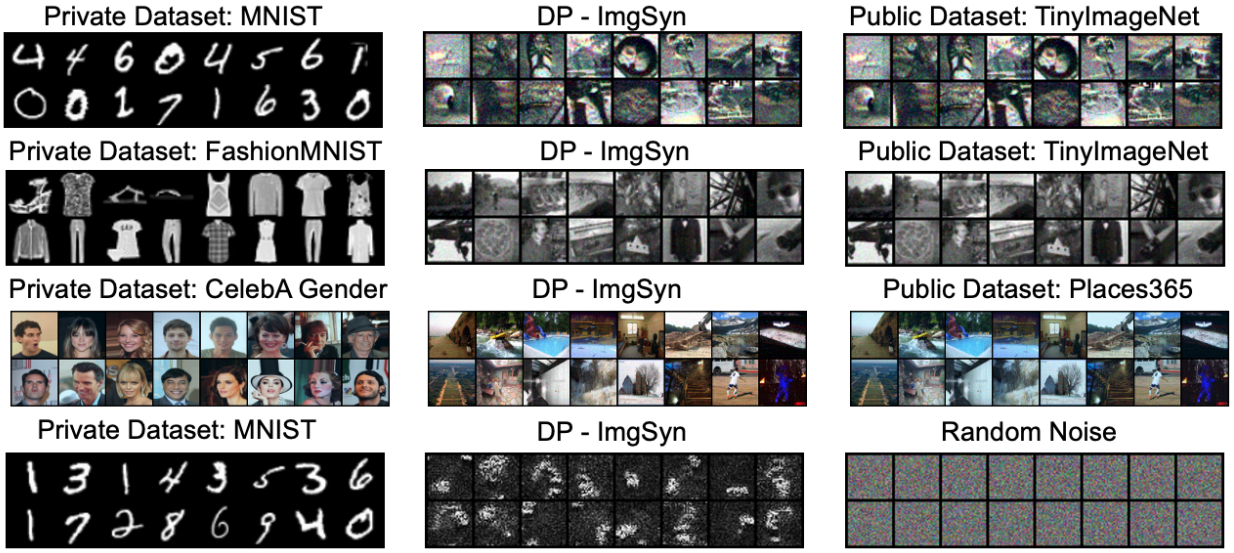


Figure 6: We visualize the private, synthetic, and public images (from left to right) for: (first row) MNIST as private dataset with TinyImageNet as public dataset (on TinyImageNet we apply grayscale image transformation because MNIST images are grayscale), (second row) FashionMNIST as private dataset with TinyImageNet as public dataset (on TinyImageNet we apply grayscale image transformation because FashionMNIST images are grayscale), (third row) CelebA Gender as private dataset with Places365 as public dataset, (fourth row) MNIST as private dataset with Gaussian Noise with zero mean and one standard deviation as initialization.

ImageNet, Places365, LSUN, and Textures. We report the train and test size, the resolution of the images, and the number of classes in each dataset.

A.9 Image Quality

In this section, we use Frechet Inception Distance (FID) (Heusel et al., 2017) to measure image quality which is common practice in literature (Wang et al., 2021). FID score is calculated based on the feature representations extracted from an ImageNet pretrained deep neural network (Inception-v3 model). However, the FID score is not recommended as a metric for grayscale images (MNIST, FashionMNIST) because it involves a network pretrained on RGB images. This affects the evaluation and the resulting evaluation is not meaningful, as explained in Wang et al. (2021). Thus we report the results only on the CelebA dataset. Moreover, we compare private dataset images, DP-ImgSyn generated images, and DP-GAN (Xie et al., 2018) generated images in Figure 7.

Experiment. We evaluate DP-ImgSyn generated images using the FID score and compare with prior works.

Table 12: Dataset statistics, training, and test set sizes for the datasets used.

Dataset	Train Set Size	Test Set Size	Resolution	Number of Classes
MNIST	60,000	10,000	28x28	10
FashionMNIST	60,000	10,000	28x28	10
CIFAR-10	50,000	10,000	32x32	10
Imagenette	10,000	5,000	224x224	10
CelebA-Hair	162,770	19,962	64x64	3
CelebA-Gender	162,770	19,962	64x64	2
TinyImageNet	100,000	10,000	32x32	200
Places365	1,803,460	10,000	32x32	365
LSUN	9,895,373	303,304	64x64	10
Textures	5,640	1,880	224x224	47

Figure 7: Comparing Private Dataset (left), DP-ImgSyn generated images for $\epsilon = 10$, $\delta = 10^{-5}$ with Places365 as the public dataset initialization (center) and DP-GAN (Xie et al., 2018) generated images, $\epsilon = 100$, $\delta = 10^{-5}$ (right).

Results. Table 13 compares the FID score of the generated images from various prior works when using CelebA as the private dataset. When using DP-ImgSyn we initialize it with Places365 (reported as A) and LSUN (reported as B) as the public dataset.

Conclusion. Our proposal achieves a low FID score (lower is better) and outperforms state-of-the-art works.

Table 13: Comparison Table with state-of-the-art techniques using FID score (lower is better) for CelebA dataset with Places365 (A) and LSUN (B) as public datasets for DP-ImgSyn.

Methods	DP-GAN	DP-MERF	P3GM	DataLens	G-PATE	DP-ImgSyn (ours)
ϵ	10^4	10^4	10^4	10	10	10
FID ↓	403.94	327.24	435.60	320.84	305.92	**188.62 (A), 194.88 (B)**