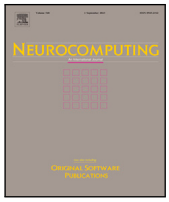
Neurocomputing 608 (2024) 128394

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Dynamic differential privacy-based dataset condensation

Zhaoxuan Wu [a], Xiaojing Gao [a], Yongfeng Qian [a,*], Yixue Hao [b,c], Min Chen [d,e]

[a] School of Computer Science, China University of Geosciences, Wuhan, 430074, China
[b] Guangdong HUST Industrial Technology Research Institute, Dongguan, 523808, China
[c] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China
[d] School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510640, China
[e] Pazhou Laboratory, Guangzhou, 510640, China

ARTICLE INFO

Communicated by L. Oneto

Keywords:
Dataset condensation
Differential privacy

ABSTRACT

With the continuous expansion of data scale, data condensation technology has emerged as a means to reduce costs related to storage, time, and energy consumption. Data condensation can generate a synthesized dataset of reduced size, enabling the training of models that exhibit high performance comparable to the original dataset. Nevertheless, data condensation has also exposed privacy issues. Although many approaches have been proposed to preserve privacy for data condensation, the privacy protection for data condensation has not been well explored. Furthermore, to the best of our knowledge, none of the existing approaches propose dynamic parameters-based differential privacy dataset condensation considering unnecessary noise introduced by the fixed privacy parameter strategy. Most approaches typically inject constant noise with the fixed variance into gradients across all layers using predefined privacy parameters, which can significantly impact model accuracy. In this paper, we investigate alternative approaches for data condensation with differential privacy (DP) that aim to ensure DP while minimizing the noise added to gradients and improving the model accuracy. First, we develop a dynamic threshold method to reduce the noise added to gradients in the later stages of training by using a clipping threshold that decreases with training rounds. Second, noise injection in our method is not arbitrary as in conventional approaches; instead, it is based on the maximum size of the gradient after clipping. This approach ensures that only minimal noise increments are introduced, thereby mitigating accuracy loss and parameter instability that may arise from excessive noise injection. Finally, our privacy analysis confirms that the proposed method provides a rigorous privacy guarantee. Extensive evaluations on different datasets demonstrate that our approach can improve accuracy compared to existing DP data condensation techniques while adhering to the same privacy budget and applying a specified clipping threshold.

1. Introduction

Datasets play a crucial role in any machine learning task. Typically, a machine learning problem starts by taking in a dataset and leveraging it to craft a model that fulfills a predefined objective. However, the increasing use of machine learning has presented two challenges for datasets. One challenge arises from the necessity to train numerous models on extensive datasets in both algorithm design and practical implementation in deep learning, substantially elevating the demands on storage and transmission resources. Another challenge stems from the increasing importance of privacy issues, prompting concerns about potential breaches of privacy resulting from dataset usage.

For the first challenge, the primary difficulty lies in handling massive datasets. One natural idea is to compress the original datasets into smaller ones and store only useful information for target tasks, which alleviates the burden on storage while maintaining model performance. Data distillation (DD), introduced by Wang et al. [1], aims to synthesize a small dataset so that models trained on it achieve high performance on the original large dataset. A dataset distillation algorithm takes a large real dataset (training set) as input and outputs a small synthetic distilled dataset. This synthesized dataset is then evaluated by testing models trained on it against a separate real dataset (validation/test set). Dataset condensation (DC) was proposed by Zhao et al. [2,3], which for the first time employs a gradient matching strategy to distill datasets, resulting in a significant improvement in testing accuracy and generalization ability. Therefore, DC can be a potential technique to address the first challenge. Considering the second challenge, recent work endeavors to demonstrate that training efficiency and privacy can be achieved simultaneously by employing DC [4]. Unfortunately,

* Corresponding author.
E-mail addresses: wuzx22@cug.edu.cn (Z. Wu), gaoxj@cug.edu.cn (X. Gao), yfqian@cug.edu.cn (Y. Qian), yixuehao@hust.edu.cn (Y. Hao), minchen@ieee.org (M. Chen).

https://doi.org/10.1016/j.neucom.2024.128394
Received 1 April 2024; Received in revised form 14 July 2024; Accepted 10 August 2024
Available online 14 August 2024

Table 1
The averaged testing accuracy of ConvNets on different datasets. Comparison under different number of samples per class $spc \in \{1, 10\}$ and clipping norm $C = 0.1$.

$C = 0.1$	Spc = 1		Spc = 10	
	Non-private	PSG	Non-private	PSG
MNIST	88.73%	88.69%	99.66%	94.92%
FashionMNIST	66.14%	65.65%	93.54%	75.61%
SVHN	31.97%	25.73%	**74.63%**	**29.85%**
CIFAR10	30.60%	29.82%	**49.63%**	**25.68%**

Table 2
The averaged testing accuracy of ConvNets on colored datasets. Comparison under different clipping norm $C \in \{0.1, 2, 4, 8, 16\}$ and the number of samples per class $spc = 10$.

		$C = 0.1$	$C = 2$	$C = 4$	$C = 8$	$C = 16$
PSG Spc = 10	SVHN	29.85%	44.57%	43.71%	**48.46%**	44.16%
	CIFAR10	25.68%	32.18%	33.96%	**34.08%**	31.61%

DC fails to improve the privacy of training machine learning models over a naive baseline [5]. The utilization of differential privacy (DP) in machine learning is a leading privacy strategy, attracting attention due to its mathematical guarantee of privacy preservation. To tackle privacy concerns, many recent works combine DD and DC with DP [6–8]. Chen et al. [8] propose private set generation (PSG) to optimize a gradient matching objective for estimating a DP-based distilled dataset. Inspired by DP-SGD [9], PSG incorporates DP constraints by sanitizing the stochastic gradient on real data at each outer iteration, resulting in the generation of high-dimensional private data.

PSG enhances sample utility while maintaining generality and reducing storage and computation consumption. However, when dealing with colored datasets, the excessive noise added by PSG to the gradient can significantly affect both utility and visual quality. Table 1 shows that compared to MNIST and FashionMNIST, PSG has a significantly greater impact on the accuracy of the synthesized datasets generated on SVHN and CIFAR10. This is because Chen et al. [2] use clipping norm $C = 0.1$ as the default hyperparameter for the main experiments, which is suitable for simpler datasets like MNIST. However, for SVHN and CIFAR10, both of which are colored datasets, a small clipping norm has a substantial impact on accuracy. We conduct an experiment to verify the above claim. Table 2 illustrates that the accuracy of the PSG algorithm varies under different sizes of clipping norm C and various datasets. Fig. 1 illustrates that for colored datasets, increasing the clipping norm C to a certain size enhances visual quality. Better visual quality indicates that the generated image is closer to a recognizable image and better reflects the characteristics and distribution of the original dataset. Based on the above findings, we conclude that:

The generated dataset exhibits reduced utility and visual quality due to the significant impact of noise injected using conventional approaches, particularly when dealing with colored datasets. It is imperative to decrease the magnitude of injected noise while still satisfying DP. For colored datasets, a small clipping norm causes the gradient to be excessively clipped, and images cannot be distilled well.

Our contributions. In conclusion, the primary objective of this paper is to significantly enhance the method of data condensation with respect to differential privacy. Our proposed solution aims to minimize the noise added to gradients while improving the overall accuracy of the model.

(1) We analyzed the performance of PSG across various datasets and observed a significant decrease in accuracy compared to non-privacy conditions when dealing with colored datasets. However, we found that under high clipping norm conditions, the accuracy decrease caused by DP in PSG when facing colored datasets was alleviated. This realization highlighted the importance of the clipping norm in the method.

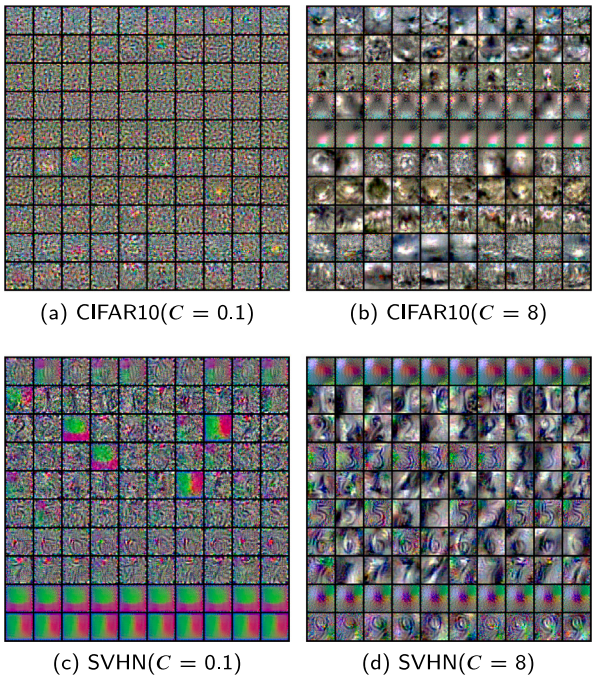


Fig. 1. Synthetic images generated by PSG under $(\epsilon, \delta) = (10, 10^{-5})$ for CIFAR10 and SVHN datasets with clipping norm $C \in \{0.1, 8\}$.

(2) We propose Dyn-PSG, a dynamic differential privacy data condensation method designed to address the limitations associated with conventional noise addition using fixed DP parameters. Dyn-PSG employs adaptive DP parameters to adjust the level of noise added during the condensation process adaptively. Specifically, we utilize a combination of dynamic gradient clipping methods and dynamic sensitivity to apply smaller noise perturbations to the gradient in later rounds.
(3) We conduct extensive experiments to evaluate Dyn-PSG on multiple datasets, including MNIST, FashionMNIST, SVHN, and CIFAR10. Our results demonstrate that compared to existing works, Dyn-PSG effectively improves the utility and visual quality of the images.

Roadmap. In Section 2, we provide an overview of DC and introduce related concepts of DP. Section 3 elaborates on our algorithm and outlines the design of the dynamic privacy parameter strategy. In Section 4, we conduct extensive experiments to demonstrate the superiority of our algorithm and evaluate the impact of key factors on the utility and visual quality of the generated images. We discuss related work in Section 5 and conclude the paper in Section 6.

2. Preliminary

2.1. Dataset condensation with gradient matching

Given a large dataset consisting of $|T|$ pairs of samples and their corresponding labels, denoted as $T = \{(x_i, y_i)\}|_{i=1}^{|T|}$, where $x_i \in \mathbb{R}^d$ represents the feature, $y_i \in \{1, \dots, L\}$ indicates the class label, and L is the total number of label classes. $f_\theta(\cdot)$ refers to the model with parameters θ. Let $S = \{(x_i^S, y_i^S)\}_{i=1}^{|S|}$ denotes the synthetic dataset. Then we can formulate the dataset condensation problem as follows:

$$\arg\min_S \mathbb{E}_{(x,y)\sim T} \ell(f_{\theta(S)}(x), y), \tag{1}$$

$$\text{where } \arg\min_\theta \mathbb{E}_{(x,y)\sim S} \ell(f_\theta(x), y), |S| \ll |T|$$

$\ell(\cdot,\cdot)$ represents a task-specific loss function, such as cross-entropy. $\ell(f_\theta(x), y)$ denotes to the cross-entropy between the model output $f_\theta(x)$ and the label y.

Gradient matching, as proposed in [2], is a method to solve the above optimization problem, which minimizes a matching loss between the model gradients on the original and synthetic data. For gradient matching, let θ^T and θ^S be the parameters trained on T and S, respectively. The objective of DC can be formulated as follows:

$$\mathbb{E}_{(x,y)\sim P_D}[\ell(F(x;\theta^D), y)] \simeq \mathbb{E}_{(x,y)\sim P_D}[\ell(F(x;\theta^S), y)]. \tag{2}$$

The expectation is taken over the real data distribution P_D. Eq. (2) can be naturally achieved once $\theta^S \approx \theta^D$. Specifically, initialized with the same values $\theta_0^S = \theta_0^D$, solving for $\theta_t^S \approx \theta_t^D$ at each training iteration t results in $\theta^S \approx \theta^D$ as desired. This can be achieved by optimizing the synthetic set S to produce a similar gradient as if the network were trained on the real dataset at each iteration i:

$$\min_S \mathcal{L}_{dis}(\nabla_\theta \mathcal{L}(S,\theta_i), \nabla_\theta \mathcal{L}(D,\theta_i)), \tag{3}$$

where $\nabla_\theta \mathcal{L}(S,\theta_i)$ represents the gradient of the classification loss on the synthetic set S, $\nabla_\theta \mathcal{L}(D,\theta_i)$ denotes the stochastic gradient on the real data, and $\mathcal{L}_{dis}$ is a sum of cosine distances between the gradients at each layer.

To replicate the training procedure, both the synthetic set S and the network $F(\cdot,\theta)$ are updated jointly in an iterative manner. In each outer iteration, S is trained to minimize the gradient matching loss $\mathcal{L}_{dis}$, while in each inner iteration, the network parameters θ_i are optimized to minimize the classification loss on the synthetic set S. Moreover, S is optimized over multiple initializations of the network parameters θ_0 to ensure the generalization ability of S across different random initializations when training a downstream model. The objective can be summarized as follows:

$$S = \arg\min_S \ \mathbb{E}_{\theta_0\sim P_{\theta_0}} \sum_{i=0}^{I-1} [\mathcal{L}_{dis}(\nabla_\theta \mathcal{L}(S,\theta_i), \nabla_\theta \mathcal{L}(D,\theta_i))], \tag{4}$$

where P_{θ_0} denotes the distribution over the initialization of network parameters.

2.2. Differential privacy

Definition 1 (*Differential Privacy(DP) [10]*)**.** For two adjacent datasets D and D', where D and D' differ from each other with only one training example, and for every possible output set O, if a randomized mechanism M satisfies

$$Pr[M(D) \in O] \le e^\epsilon \cdot Pr[M(D') \in O] + \delta, \tag{5}$$

then M obeys (ϵ,δ)-DP.

Definition 2 (*Sensitivity [11]*)**.** Let D denote the domain of possible input data and R denote the domain of all possible output. The sensitivity of a function $f: D \to R$ is the maximum amount by which the function value changes when a single entry of the input is perturbed.

$$S' = max_{A,A'\subseteq D, \|A-A'\|_0=1} \|f(A) - f(A')\|_p. \tag{6}$$

Definition 2 implies that to create a randomized differential privacy algorithm $M(f)$ by adding noise that follows some randomization distribution while preserving the utility of f, we need to normalize the noise by the maximum change, defined as the sensitivity of function f with neighboring inputs.

Definition 3 (*Gaussian Mechanism [10]*)**.** Let $f: X \to R_d$ be an arbitrary d-dimensional function with sensitivity defined as

$$\Delta_2 f = max D, D \|f(D) - f(D')\|_2 \tag{7}$$

over all adjacent datasets D and D'. The Gaussian mechanism M_σ, parameterized by σ, adds noise into the output, *i.e.*,

$$M_\sigma(x) = f(x) + N(0, \sigma^2 I) \tag{8}$$

M_σ satisfies (ϵ,δ)-DP for $\sigma \ge \sqrt{2ln(1.25/\delta)}\Delta_2 f/\epsilon$.

Any privacy cost is normalized upon releasing the private set of generated data due to the closure of DP under post-processing.

Lemma 1. *Let the noise variance ς^2 in the Gaussian mechanism be $\sigma^2 S'^2$, where σ is the noise scale and S' is the l_2 sensitivity. We have the noise scale σ satisfying $\sigma^2 > \frac{2lo f(1.25/\delta)}{\epsilon^2}$.*

According to Lemma 1, the noise scale σ and privacy loss ϵ have an inverse correlation given a fixed δ. That is, a large noise scale indicates a small ϵ, while a small noise scale implies the expenditure of a large privacy budget ϵ.

Theorem 1 (*Post-processing [10]*)**.** *If M satisfies (ϵ,δ)-DP, then $F \circ X$ will also satisfy (ϵ,δ)-DP for any data-independent function F, where $\circ$ denotes the composition operator.*

2.3. Dataset condensation with differential privacy

Some works, such as PSG [8], modify the gradient matching framework by clipping and adding white noise to the gradients obtained from the original dataset during the optimization process. These methods integrate DP constraints by sanitizing the stochastic gradient on real data at each outer iteration, while maintaining the inner iterations unchanged as their privacy is guaranteed by the post-processing, as demonstrated in Theorem 1. The final objective can be formulated as follows:

$$S = \arg\min_S \ \mathbb{E}_{\theta_0\sim P_{\theta_0}} \sum_{i=0}^{I-1} [\mathcal{L}_{dis}(g_{\theta_t}^S, \widetilde{g_{\theta_t}^T})]$$

where $\widetilde{g_{\theta_t}^T}$ denotes the parameter gradient on D that is sanitized via Gaussian mechanism (Definition 3), and $g_{\theta_t}^S$ denotes the parameter gradient on T. Such a routine has been demonstrated to offer improved sample utility, while also satisfying strict differential privacy guarantees.

3. Our algorithm: Dyn-PSG

In this section, we introduce our method, Dyn-PSG, which utilizes dynamic gradient clipping and dynamic sensitivity to minimize the noise added to the gradient while maintaining accuracy performance.

3.1. Outline of Dyn-PSG

We demonstrate the process of dataset condensation with differential privacy in Fig. 2(a) and outline Dyn-PSG in Algorithm 1. Dyn-PSG is devised based on matching gradients of original and synthetic data. Given a dataset $D = \{(x_i, y_i)_{i=1}^N\}$ where $x_i \in \mathbb{R}^d$ represents the feature, $y_i \in \{1,\dots,L\}$ denotes the class label, N is the number of samples, and L is the number of label classes. Our objective is to synthesize a set of samples $S = \{(x_i^S, y_i^S)_{i=1}^N\}$ such that samples in S have the same form as data in D.

For each uniformly sampled random batch $\{(x_i, y_i)_{i=1}^B\}$, let $g_{\theta_t}^D(x_i)$ denote the per-example gradient on real data. With the clipping norm $C_t = DECAY(C,t)$, the per-example gradient on real data $g_{\theta_t}^D(x_i)$ is preserved if its l_2 norm satisfies $\|g_{\theta_t}^D(x_i)\|_2 \le C_t$. Otherwise, if $\|g_{\theta_t}^D(x_i)\|_2 > C_t$, the gradient vector $g_{\theta_t}^D(x_i)$ needs to be brought down so that its l_2 norm is capped by C_t. This is achieved by multiplying every coordinate of the gradients with a scaling factor: $C_t/\|g_{\theta_t}^D(x_i)\|_2$. Such per-example gradient clipping is applied to each example (x_i, y_i) in the batch of the current iteration. Then, we calculate $S_{dyn} = max_i \|g_{\theta_t}^D(x_i)\|_2$ and inject the (ϵ,δ)-differential privacy controlled Gaussian noise to the average gradient: $\widetilde{g_{\theta_t}^D} = \frac{1}{B}\sum_{i=1}^B (\widetilde{g_{\theta_t}^D}(x_i) + \mathcal{N}(0, \sigma^2 {S_{dyn}}^2 I))$. The entire process of gradient clipping and noise addition is shown in Fig. 2(b). We compute parameter gradients on synthetic data $g_{\theta_t}^S$ and update S by

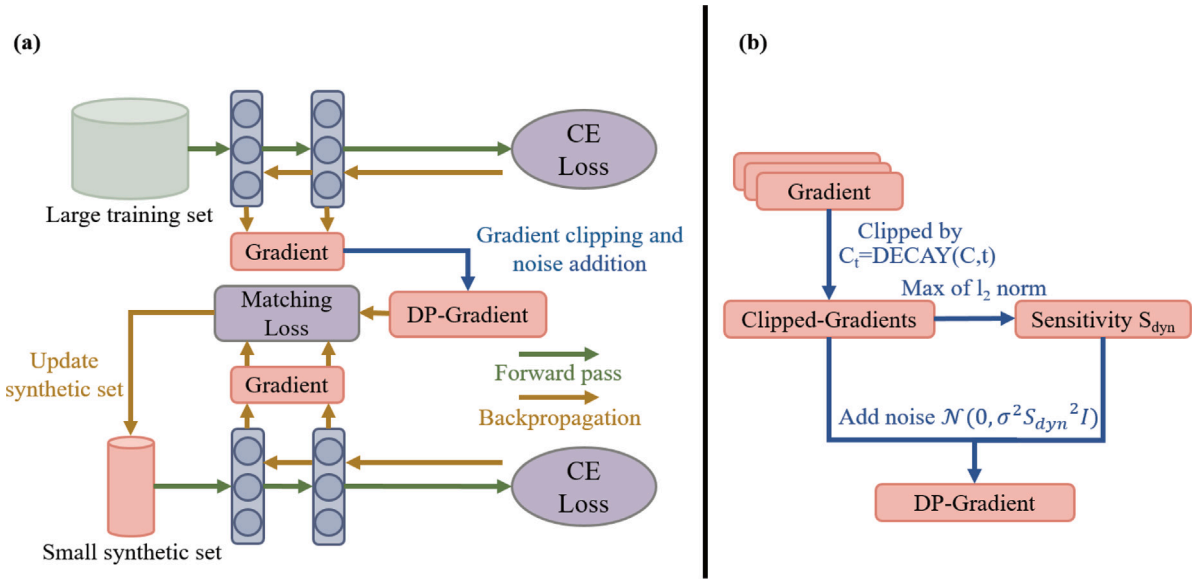


Fig. 2. Dataset condensation with differential privacy (left) is achieved by clipping gradients and adding noise to stochastic gradient on large training set. Our algorithm uses dynamic parameters during gradient clipping and noise addition (right), thereby reducing the noise added during training. CE denotes Cross-Entropy.

$\nabla_S \mathcal{L}_{dis}(g_{\theta_t}^S, \widetilde{g_{\theta_t}^D})$, where $\mathcal{L}_{dis}$ is the sum of cosine distances between the gradients at each layer [2,3]. Subsequently, the network parameters θ_t are optimized to minimize the classification loss on the synthetic set S in each inner iterations.

Proposition 1. *The Dyn-PSD algorithm produces a (ϵ, δ)-DP distilled dataset.*

Proof. Our privacy computation is based on the notion of Rényi-DP, which we recall as follows.

Definition 4 (*Rényi Differential Privacy (RDP) [12]*)**.** A randomized mechanism M is (α, ϵ)-RDP with order α, if

$$D_\alpha(M(D) \parallel M(D')) = \frac{1}{\alpha - 1} log\mathbb{E}_{x\sim M(D)}\left[\left(\frac{Pr[M(D) = X]}{Pr[M(D') = x]}\right)^{\alpha-1}\right] \leq \epsilon \tag{9}$$

holds for any adjacent datasets D and D', where

$$D_\alpha(P \parallel Q) = \frac{1}{\alpha - 1} log\mathbb{E}_{x\sim Q}[(P(x)/Q(x))^\alpha]$$

is the Rényi divergence of order $\alpha > 1$ between the distributions P and Q.

To compute the privacy cost of our approach, we numerically compute $D_\alpha(M(D) \parallel D(D'))$ in Definition 4 for a range of orders α [13,14] in each training step, which involves accessing the real gradient g_θ^D. To obtain the overall accumulated privacy cost over multiple training iterations, we utilize the composition properties of RDP, as summarized by the following theorem.

Theorem 2 (*Adaptive Composition of RDP [13]*)**.** *Let $f : D \to R_1$ be $(\alpha, \epsilon_1) - RDP$ and $g : R_1 \times D \to R_2$ be $(\alpha, \epsilon_2) - RDP$, then the mechanism defined as (X, Y), where $X \sim f(D)$ and $Y \sim g(X, D)$, satisfies $(\alpha, \epsilon_1 + \epsilon_2)$-RDP*

In total, our Dyn-PSG approach, as depicted in Algorithm 1, can be regarded as a composition over RTK (*i.e.*, the number of iterations where the real gradient is used) homogeneous subsampled Gaussian mechanisms, with the subsampling ratio set to B/N, in terms of the privacy cost [15].

Theorem 3 (*From RDP to (ϵ, δ)-DP [12]*)**.** *if M is a (α, ϵ)-RDP mechanism, then M is also $(\epsilon + \frac{log1/\delta}{\alpha-1}, \delta)$-DP for any $0 < \delta < 1$.*

Lastly, we leverage Theorem 3 to convert (α, ϵ)-RDP to (ϵ, δ)-DP, and prove that the Dyn-PSD algorithm produces a (ϵ, δ)-DP distilled dataset.

3.2. Dynamic privacy parameters in Dyn-PSG

In order to mitigate the excessive noise added to gradients, we adopt a dynamic privacy parameter strategy as follow.

***C*_decay.** Given that gradients tend to leak more information during early training iterations than in later stages [16], a natural approach is to design a differential privacy algorithm that injects larger noise during the early stages of training and gradually reduces the noise as training progresses. In the subsequent experiments, we used a linear decay method to make clipping norm C_t decay with t: Defined as $C_t = C(1 - \gamma t)$ where $\gamma > 0$ is the smooth controlling term for clipping at iteration t. Given that clipping controls the largest gradient change during training, the maximum gradient changes decrease with training and progressively approach zero as convergence is reached. Therefore, a dynamic decaying clipping method can predict the maximum changes in gradients and adjust the dynamic sensitivity correspondingly.

l_2-max sensitivity. The sensitivity is calibrated to the l_2-norm of the gradients to prevent the addition of excessive noise. A smaller noise variance is then employed to inject noise during the later stages of training, enhancing the convergence speed while maintaining high accuracy performance.

Given that the constant noise generated by the numerical value of C is excessive for the later stages of model training, the method for dynamically adjusting the noise variance calibrated is to use the max l_2 norm measured on per-example gradients in a batch B as the sensitivity of the training function f_t for iteration t. Consider the following scenario: if the l_2 norm of all per-example gradients in a batch is smaller than the pre-defined clipping norm C, then C becomes an overestimation of the true sensitivity of the function f_t at iteration t.

To address this, we redefine the sensitivity of f_t as the maximum l_2 norm among these per-example gradients in the batch. In other words, the l_2-max sensitivity will be the smaller value between the maximum l_2 norm and the clipping norm C. Since the l_2 norm of the gradients closely follows the trend of gradient changes throughout the training process, our dynamic l_2 sensitivity approach adjusts the sensitivity S' accordingly. This results in the injection of smaller differential privacy noise during the later stages of training.

We integrate both the l_2-max sensitivity and the clipping decay sensitivity C_decay into PSG to leverage the benefits of both approaches.

Algorithm 1: Dyn-PSG

Input: Dataset $D = \{(x_i, y_i)_{i=1}^{N}\}$, learning rate η_θ/η_S, clipping norm C, batch size B, number of experiment E, outer iterations T, inner iterations J, batches K, desired privacy cost ϵ given a pre-defined δ

Output: Synthetic set S

Compute the required DP noise scale σ numerically [11,13] so that the privacy cost equals ϵ after the training; Initialize synthetic set S using Kaiming initialization and the synthetic samples using standard Gaussian ;

for *exp* ***in*** *{1,..., E}* **do**
  Initialize model parameter $\theta_0 \sim P_{\theta_0}$;
  for *outer_iter* ***in*** *{1,...,T}* **do**
    $\theta_{t+1}=\theta_t$;
    $C_t = C(1-\gamma t)$;
    for *batch_index* ***in*** *{1,...,K}* **do**
      Uniformly sample random batch $\{(x_i, y_i)\}_{i=1}^{B}$ from D;
      for *each* (x_i, y_i) **do**
        // Compute per-example gradients on real data
        $g_{\theta_t}^{D}(x_i) = \ell(F(x_i;\theta_t), y_i)$;
        // Clip gradients
        $\widetilde{g_{\theta_t}^{D}}(x_i) = g_{\theta_t}^{D}(x_i) \cdot min(1, C_t/||g_{\theta_t}^{D}(x_i)||)$;
      end
      // Compute the max of l_2 norm over M layers on the batch gradient for iteration t, assign the sensitivity S_{dyn}.
      $S_{dyn} = max_i||\widetilde{g_{\theta_t}^{D}}(x_i)||_2$;
      // Add noise to average gradient with Gaussian mechanism
      $\widetilde{g_{\theta_t}^{D}} = \frac{1}{B}\sum_{i=1}^{B}(\widetilde{g_{\theta_t}^{D}}(x_i) + \mathcal{N}(0, \sigma^2 {S_{dyn}}^2 I))$;
      // Compute parameter gradients on synthetic data and update S
      $g_{\theta_t}^{S} = \nabla_\theta \mathcal{L}(S, \theta_t) = \frac{1}{M}\sum_{i=1}^{M}\ell(F(x_i^S;\theta_t), y_i^S)$;
      $S = S - \eta_S \cdot \nabla_S \mathcal{L}_{dis}(g_{\theta_t}^{S}, \widetilde{g_{\theta_t}^{D}})$;
    end
    for *inner_iter* ***in*** $1, \ldots, J$ **do**
      // Update network parameter using S
      $\theta_t = \theta_t - \eta_\theta \cdot \nabla_\theta \mathcal{L}(S, \theta_t)$;
    end
  end
end
return Synthetic set S

This integration involves using the decaying clipping sensitivity C instead of the initially large clipping norm. According to Lemma 1, neither the decay clipping-based dynamic sensitivity nor l_2-max sensitivity directly affect the differential privacy guarantee. However, when aiming for a specific accuracy target, the model optimized with combined sensitivity approaches may achieve the target accuracy and terminate training earlier, resulting in reduced overall privacy cost.

4. Experiments

In this section, we conduct experiments to verify the effectiveness of Dyn-PSG. Specifically, we compare Dyn-PSG to baseline methods on datasets with varying resolutions. We assess its cross-architecture performance and the generalization capabilities of the synthetic datasets. Additionally, we analyze the impact of various hyperparameters.

Table 3
The accuracy of ConvNets on DP Sinkhorn, DP-MERF, PSG and Dyn-PSG. We set $\epsilon = 1$ and show the averaged accuracy over three independent runs.

Method	MNIST	FMNIST	SVHN	CIFAR10
DP Sinkhorn	86.95%	43.22%	19.92%	12.65%
DP-MERF	84.88%	64.65%	22.31%	17.26%
PSG(Spc = 1)	86.34%	65.53%	25.46%	25.13%
Dyn-PSG(Spc = 1)	87.11%	66.70%	25.73%	29.10%
PSG(Spc = 10)	90.01%	70.49%	47.30%	31.23%
Dyn-PSG(Spc = 10)	92.55%	71.83%	49.09%	33.85%

Table 4
The averaged testing accuracy of ConvNets on SVHN and CIFAR10. Comparison under different clipping norm $C \in \{2, 4, 8, 16\}$ and the number of samples per class $spc = 10$.

Spc = 10	Method	$C = 2$	$C = 4$	$C = 8$	$C = 16$
SVHN	Non-private		76.1%		
	PSG	44.57%	43.71%	**48.46%**	44.16%
	Ours	44.04%	49.01%	**52.12%**	51.96%
CIFAR10	Non-private		44.9%		
	PSG	32.18%	33.96%	**34.08%**	31.61%
	Ours	31.59%	35.06%	**35.84%**	35.45%

4.1. Experimental setup

4.1.1. Datasets

We utilize two widely used benchmark datasets in this paper.

- **MNIST** [17] consists of 60,000 samples of 28 × 28 grayscale images depicting handwritten digits sorted into 10 classes.
- **FashionMNIST** [18] consists of 60,000 samples of 28 × 28 grayscale images depicting items of clothing sorted into 10 classes.
- **CIFAR10** [19] consists of 60,000, 32 × 32 colored images in 10 classes, with 6000 images per class. There are 50,000 training images and 10,000 test images. The classes are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.
- **SVHN** [20] is a digit classification benchmark dataset containing images of printed digits (from 0 to 9) clipped from pictures of house number plates. Each sample is sized at 32 × 32. The dataset comprises 73,257 digits for training and 26,032 digits for testing.

4.1.2. Baselines

We compare our method to PSG, DP Sinkhorn [21] and DP-MERF [22]. For PSG, we use Chen et al. [8]'s code and set $\epsilon = 1$ or $\epsilon = 10$. For DP Sinkhorn, we use Cao et al. [21]'s code to run the experiments. We set m to 1 and $\epsilon = 1$. For DP-MERF, we utilize Harder et al. [22]'s code and set $\epsilon = 1$.

4.1.3. Model and hyperparameters

We utilize a ConvNet with 3 blocks as the default architecture. Each block consists of one Convolutional layer with 128 filters, followed by Instance Normalization [23], ReLU activation, and Average Pooling modules. The final output layer is a fully connected (FC) layer. We initialize the network parameters using Kaiming initialization [24] and the synthetic samples are initialized using a standard Gaussian distribution. The default hyperparameters used for the main experiments are as follows: $R = 1000$, $K = 10$, and $T = 10$. For MNIST and FashionMINST, $J = 1$, while for SVHN and CIFAR10, $J = 10$.

4.2. Main result

We conduct comparative experiments using Dyn-PSG, PSG, DP-MERF and DP Sinkhorn on MNIST, FashionMNIST, SVHN, and CIFAR10, and analyze the experimental results. The main experimental

Table 5
Comparison of generalization ability across different network architecture with $(\epsilon, \delta) = (10, 10^{-5})$ and clipping norm $C = 8$. Generated set is optimized with ConvNet, while the downstream classifiers are of different architecture.

	SVHN						CIFAR10					
	ConvNet	MLP	LeNet	AlexNet	VGG11	ResNet18	ConvNet	MLP	LeNet	AlexNet	VGG11	ResNet18
non-private	74.63%	28.86%	27.61%	28.67%	52.28%	20.01%	49.63%	30.87%	27.47%	15.11%	39.08%	22.25%
PSG	48.46%	**21.04%**	17.47%	13.87%	29.64%	15.34%	34.08%	27.91%	27.67%	**26.95%**	31.71%	17.08%
Ours	**52.12%**	17.92%	**22.01%**	**19.74%**	**30.17%**	**15.41%**	**35.84%**	**28.60%**	**29.00%**	26.35%	**32.99%**	**21.58%**

results are shown in Table 3. PSG and our Dyn-PSG algorithm achieve comparable performance to DP-MERF and DP-Sinkhorn with 1 sample per class. However, when using 10 samples per class, Dyn-PSG outperforms PSG, DP-Sinkhorn, and DP-MERF, particularly on colored datasets.

In order to provide a more comprehensive comparison with PSG, we conduct extensive experiments comparing Dyn-PSG with PSG. Table 4 reports the results of experiments comparing Dyn-PSG with PSG on colored datasets with different clipping norms. Our experiments on colored image classification datasets demonstrate that compared with PSG, Dyn-PSG improves accuracy by 3.66% on SVHN and 1.76% on CIFAR10 under the same privacy budget and a certain clipping threshold. Table 4 also indicates that PSG's performance is sensitive to the settings of clipping bound for colored datasets, leading to lower accuracy when the clipping bound is set too large (*e.g.* C = 16). In contrast, Dyn-PSG uses dynamic l_2-max sensitivity, which adapts from iteration to iteration, closely aligning with the decreasing trend of gradients throughout the training. From Fig. 3, we observe that when $(\epsilon, \delta) = (1, 10^{-5})$ and the clipping norm $C = 8$, the accuracy of Dyn-PSG increases more rapidly than that of the baseline and ultimately achieves better results. This demonstrates that our method reaches the target accuracy earlier and surpasses the final performance of the baseline.

As shown in Table 5, we further train various model architectures on the synthetic data generated by PSG and Dyn-PSG and present the testing accuracy.

We visualize synthetic images generated by PSG and Dyn-PSG in Fig. 4. Compared to PSG-generated images, we find that the images generated by Dyn-PSG, particularly when optimized with ConvNet, exhibit superior visual quality, even when utilized with downstream classifiers of different architectures.

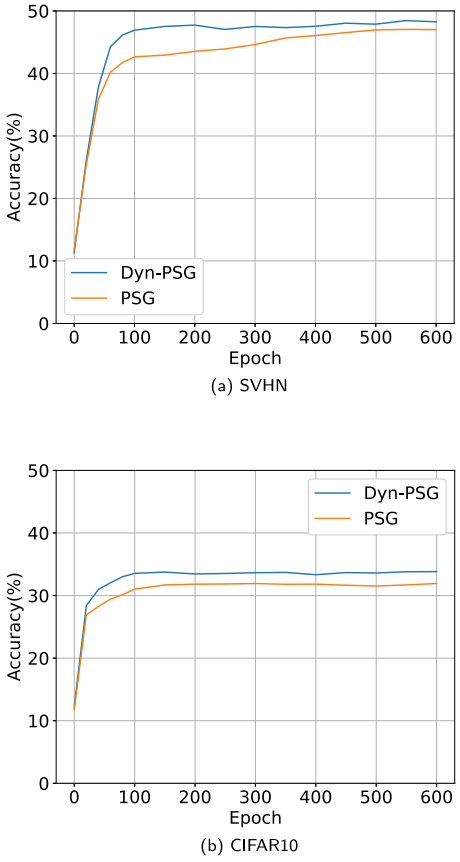


Fig. 3. Accuracy on SVHN and CIFAR10. We set $(\epsilon, \delta) = (1, 10^{-5})$ and clipping norm $C = 8$.

5. Related work

5.1. Synthetic data generation

Previous research has explored a variety of generative methods for synthesizing data [25–30]. Due to the growing concerns of personal privacy and the escalating frequency of data breaches, research in the field of data privacy has been rapidly expanding in recent years. This surge has witnessed a substantial increase in research efforts aimed at developing differential private generative methods, particularly within the domain of medical image research [31,32]. Existing work primarily leverages DP-SGD to implement privacy within the Generative Adversarial Networks (GANs) framework by privatizing the generator. In GANs, the generator never has direct access to training data and thus requires no privatization, as long as the discriminator is differential private. Examples of this approach include DP-CGAN [33], DP-GAN [34], G-PATE [35], DataLens [36], and GS-WGAN [37]. Recently, some new works outside the GANs framework have been proposed, such as DP-MERF [22] and DPHP [38], which adopt the approximate versions of the maximum mean discrepancy (MMD) as their loss function; and DP-Sinkhorn [21] that proposes using the Sinkhorn divergence in privacy settings.

5.2. Dynamic privacy parameters for deep learning

Pichapati et al. [39] propose Adaclip, a differential private SGD algorithm that provably adds less noise compared DP-SGD, by using coordinate-wise adaptive clipping of the gradient. Andrew et al. [40] propose Quantileclip wherein instead of a fixed clipping norm, one clips to a value at a specified quantile of the update norm distribution. Both of the above methods reduce excessive noise by changing the gradient clipping method. Wei et al. [41] propose DP-dyn[S, σ], which have the sensitivity calibrated to the l_2-norm of the gradients, and have a smoothly decaying noise scale such that the noise variance follows the trend of gradient updates across the T training iterations. Zhu et al. [42] propose a fine-grained differentially private mechanism to mitigate the issue of privacy leakage in federated learning systems, specifically focusing on the potential leakage from gradients. The fine-grained approach to differential privacy in federated learning is innovative and addresses a critical privacy concern. Liu et al. [43] propose PADL, a privacy-aware and asynchronous deep learning framework designed for Internet of Things (IoT) applications, which is specifically designed for IoT applications, addressing the unique challenges of these environments, such as heterogeneity and resource constraints.

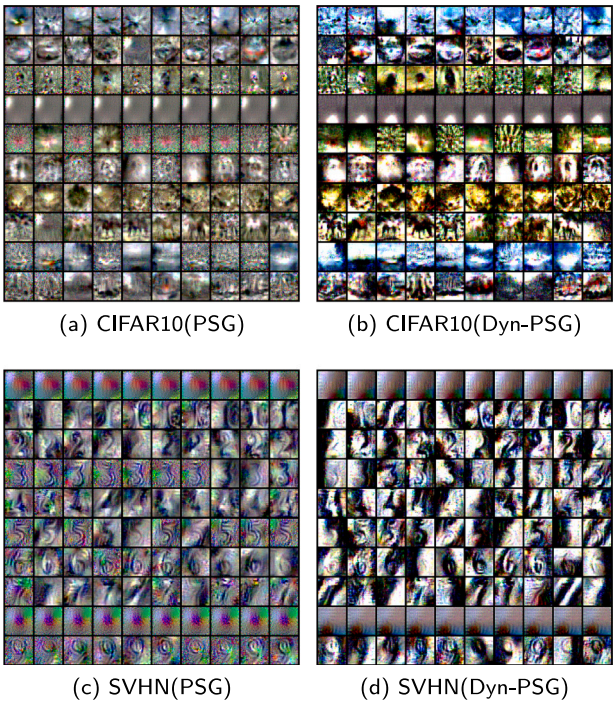


Fig. 4. Synthetic images generated by PSG and Dyn-PSG under $(\epsilon, \delta) = (10, 10^{-5})$ for CIFAR10 and SVHN datasets with clipping norm $C = 8$.

6. Conclusion

In this paper, we revisit the challenge of privacy-preserving dataset condensation and identify issues associated with fixed clipping norms, particularly when dealing with colored data. Our experiments on PSG with colored datasets demonstrate that a too-small clipping norm restricts the condensation on such datasets, while a larger pruning threshold helps mitigate the impact of gradient clipping. To address these challenges and enhance the utility of the generated dataset, we propose dynamic privacy parameters dataset condensation algorithm called Dyn-PSG. We show that datasets optimized with Dyn-PSG offer improved utility and visual quality. Furthermore, we observe our generated sets optimized with ConvNet provides better utility even when downstream classifiers have different architectures. We hope our work can inspire further research in this promising direction to alleviate the cost burden and privacy concern in deep learning.

CRediT authorship contribution statement

Zhaoxuan Wu: Writing – review & editing, Writing – original draft, Visualization, Validation, Investigation, Formal analysis. **Xiaojing Gao:** Writing – review & editing, Software, Resources. **Yongfeng Qian:** Writing – review & editing, Supervision, Project administration, Methodology . **Yixue Hao:** Writing – review & editing, Formal analysis, Conceptualization. **Min Chen:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

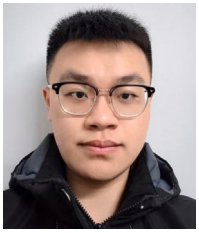
Acknowledgments

This work was supported by Guangdong Provincial Key Laboratory of Manufacturing Equipment Digitization (2023B1212060012), and Guangdong Basic and Applied Basic Research Foundation 2024A1515030017. This work was also supported by the National Natural Science Foundation of China (NSFC) under Grant No.62276109 and No.62301515; the Natural Science Foundation of Hubei Province (Grant number 2023AFB045).

References

[1] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, Alexei A. Efros, Dataset distillation, CoRR (2018) arXiv:1811.10959.
[2] Bo Zhao, Konda Reddy Mopuri, Hakan Bilen, Dataset condensation with gradient matching, 2020, arXiv preprint arXiv:2006.05929.
[3] Bo Zhao, Hakan Bilen, Dataset condensation with differentiable siamese augmentation, 2021, arXiv:2102.08259.
[4] Tian Dong, Bo Zhao, Lingjuan Lyu, Privacy for free: How does dataset condensation help privacy? in: International Conference on Machine Learning, PMLR, 2022, pp. 5378–5396.
[5] Nicholas Carlini, Vitaly Feldman, Milad Nasr, No free lunch in "privacy for free: How does dataset condensation help privacy", 2022, arXiv e-prints.
[6] Tianhang Zheng, Baochun Li, Differentially private dataset condensation, 2023.
[7] Margarita Vinaroz, Mi Jung Park, Differentially private kernel inducing points (DP-KIP) for privacy-preserving data distillation, 2023.
[8] Dingfan Chen, Raouf Kerkouche, Mario Fritz, Private set generation with discriminative information, 2022.
[9] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, Li Zhang, Deep learning with differential privacy, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 308–318.
[10] Cynthia, Dwork, Aaron, Roth, The algorithmic foundations of differential privacy, Found. Trends Theor. Comput. Sci. (2013).
[11] Cynthia Dwork, Frank McSherry, Kobbi Nissim, Adam Smith, Calibrating noise to sensitivity in private data analysis, in: S. Halevi, T. Rabin (Eds.), THEORY of CRYPTOGRAPHY, PROCEEDINGS, in: Lecture Notes in Computer Science, Vol. 3876, Int Assoc Cryptol Res; Columbia Univ, Comp Sci Dept, 2006, pp. 265–284, 3rd Theory of Cryptography Conference, Columbia Univ, New York, NY, MAR 04-07, 2006.
[12] Ilya Mironov, Rényi differential privacy, in: 2017 IEEE 30th Computer Security Foundations Symposium, CSF, IEEE, 2017, pp. 263–275.
[13] Ilya Mironov, Kunal Talwar, Li Zhang, Rényi differential privacy of the sampled Gaussian mechanism, 2019.
[14] Yu Xiang Wang, Borja Balle, Shiva Prasad Kasiviswanathan, Subsampled rényi differential privacy and analytical moments accountant, J. Priv. Confid. (2020).
[15] Yu-Xiang Wang, Borja Balle, Shiva Prasad Kasiviswanathan, Subsampled rényi differential privacy and analytical moments accountant, CoRR (2018) arXiv:1808.00087.
[16] Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, Yanzhao Wu, A framework for evaluating client privacy leakages in federated learning, in: Liqun Chen, Ninghui Li, Kaitai Liang, Steve Schneider (Eds.), Computer Security – ESORICS 2020, Springer International Publishing, Cham, 2020, pp. 545–566.
[17] Yann LeCun, Corinna Cortes, MNIST handwritten digit database, 2010.
[18] Han Xiao, Kashif Rasul, Roland Vollgraf, Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017.
[19] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Handb. Syst. Autoimmun. Dis. 1 (4) (2009).
[20] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y. Ng, et al., Reading digits in natural images with unsupervised feature learning, in: NIPS Workshop on Deep Learning and Unsupervised Feature Learning, Vol. 2011, (5) Granada, Spain, 2011, p. 7.
[21] Tianshi Cao, Alex Bie, Arash Vahdat, Sanja Fidler, Karsten Kreis, Don't generate me: Training differentially private generative models with sinkhorn divergence, in: Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, Jennifer Wortman Vaughan (Eds.), Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, Virtual, 2021, pp. 12480–12492, URL https://proceedings.neurips.cc/paper/2021/hash/67ed94744426295f96268f4ac1881b46-Abstract.html.
[22] Frederik Harder, Kamil Adamczewski, Mijung Park, DP-MERF: differentially private mean embeddings with RandomFeatures for practical privacy-preserving data generation, in: Arindam Banerjee, Kenji Fukumizu (Eds.), The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event, in: Proceedings of Machine Learning Research, Vol. 130, PMLR, 2021, pp. 1819–1827, URL http://proceedings.mlr.press/v130/harder21a.html.

[23] Dmitry Ulyanov, Andrea Vedaldi, Victor S. Lempitsky, Instance normalization: The missing ingredient for fast stylization, CoRR (2016) arXiv:1607.08022.
[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, in: 2015 IEEE International Conference on Computer Vision, ICCV, 2015, pp. 1026–1034.
[25] Diederik P. Kingma, Max Welling, Auto-encoding variational Bayes, 2022.
[26] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, Yoshua Bengio, Generative adversarial nets, in: Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, Kilian Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, 2014, pp. 2672–2680, URL https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html.
[27] Mehdi Mirza, Simon Osindero, Conditional generative adversarial nets, 2014.
[28] Irina Higgins, Loic Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, Alexander Lerchner, Beta-vae: Learning basic visual concepts with a constrained variational framework., ICLR (Poster) 3 (2017).
[29] Martín Arjovsky, Soumith Chintala, Léon Bottou, Wasserstein generative adversarial networks, in: Doina Precup, Yee Whye Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, in: Proceedings of Machine Learning Research, Vol. 70, PMLR, 2017, pp. 214–223, URL http://proceedings.mlr.press/v70/arjovsky17a.html.
[30] Andrew Brock, Jeff Donahue, Karen Simonyan, Large scale GAN training for high fidelity natural image synthesis, CoRR (2018) arXiv:1809.11096.
[31] Guang Li, Ren Togo, Takahiro Ogawa, Miki Haseyama, Soft-label anonymous gastric x-ray image distillation, in: 2020 IEEE International Conference on Image Processing, ICIP, IEEE, 2020, pp. 305–309.
[32] Guang Li, Ren Togo, Takahiro Ogawa, Miki Haseyama, Compressed gastric image generation based on soft-label dataset distillation for medical data sharing, Comput. Methods Programs Biomed. 227 (2022) 107189.
[33] Reihaneh Torkzadehmahani, Peter Kairouz, Benedict Paten, Dp-cgan: Differentially private synthetic data and label generation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019.
[34] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, Jiayu Zhou, Differentially private generative adversarial network, CoRR (2018) arXiv:1802.06739.
[35] Yunhui Long, Boxin Wang, Zhuolin Yang, Bhavya Kailkhura, Aston Zhang, Carl A. Gunter, Bo Li, G-PATE: scalable differentially private data generator via private aggregation of teacher discriminators, in: Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, Jennifer Wortman Vaughan (Eds.), Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, Virtual, 2021, pp. 2965–2977, URL https://proceedings.neurips.cc/paper/2021/hash/171ae1bbb81475eb96287dd78565b38b-Abstract.html.
[36] Boxin Wang, Fan Wu, Yunhui Long, Luka Rimanic, Ce Zhang, Bo Li, DataLens: Scalable privacy preserving training via gradient compression and aggregation, in: Yongdae Kim, Jong Kim, Giovanni Vigna, Elaine Shi (Eds.), CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021, ACM, 2021, pp. 2146–2168.
[37] Dingfan Chen, Tribhuvanesh Orekondy, Mario Fritz, Gs-wgan: A gradient-sanitized approach for learning differentially private generators, Adv. Neural Inf. Process. Syst. 33 (2020) 12673–12684.
[38] Margarita Vinaroz, Mohammad-Amin Charusaie, Frederik Harder, Kamil Adamczewski, Mijung Park, Hermite polynomial features for private data generation, in: Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, Sivan Sabato (Eds.), International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA, in: Proceedings of Machine Learning Research, Vol. 162, PMLR, 2022, pp. 22300–22324, URL https://proceedings.mlr.press/v162/vinaroz22a.html.
[39] Venkatadheeraj Pichapati, Ananda Theertha Suresh, Felix X. Yu, Sashank J. Reddi, Sanjiv Kumar, AdaCliP: Adaptive clipping for private SGD, 2019.
[40] Galen Andrew, Om Thakkar, H. Brendan McMahan, Swaroop Ramaswamy, Differentially private learning with adaptive clipping, 2022.
[41] Wenqi Wei, Ling Liu, Gradient leakage attack resilient deep learning, 2021.
[42] Linghui Zhu, Xinyi Liu, Yiming Li, Xue Yang, Shu-Tao Xia, Rongxing Lu, A fine-grained differentially private federated learning against leakage from gradients, IEEE Internet Things J. 9 (13) (2021) 11500–11512.
[43] Xiaoyuan Liu, Hongwei Li, Guowen Xu, Sen Liu, Zhe Liu, Rongxing Lu, PADL: Privacy-aware and asynchronous deep learning for IoT applications, IEEE Internet Things J. 7 (8) (2020) 6955–6969.



Zhaoxuan Wu received the B.Sc. degree from the School of Computer Science and Technology, China University of Geosciences, Wuhan, China, in 2022. He is currently pursuing a master's degree in Computer Science and Technology at China University of Geosciences, Wuhan, China. His research interests include security and privacy, dataset Condensation.

Xiaojing Gao received the Ph.D. degrees in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2018, where she is currently working as a Lecturer. Her research interests include secure communications, chaotic encryption, chaos synchronization, and nonlinear time series analysis.

Yongfeng Qian received the Ph.D. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology (HUST), in 2018. She is currently an Associate Professor with the China University of Geosciences, Wuhan, China. Her research interests include vehicular networks, security and privacy, mobile crowdsening, cloud computing, and the Internet of Things.

Yixue Hao is an Associate Professor in the School of Computer Science and Technology at Huazhong University of Science and Technology (HUST). He received the Ph.D degree in computer science from HUST, Wuhan, China, in 2017. He was named in Clarivate Analytics Highly Cited Researchers List in 2020. His current research interests include cognitive computing, edge computing, and multi-agent reinforcement learning.

Min Chen (Fellow, IEEE) has been a Full Professor with School of Computer Science and Engineering, South China University of Technology. He is also the director of Embedded and Pervasive Computing (EPIC) Lab at Huazhong University of Science and Technology. He is the founding Chair of IEEE Computer Society Special Technical Communities on Big Data. He was an assistant professor in School of Computer Science and Engineering at Seoul National University before he joined HUST. He is the Chair of IEEE Globecom 2022 eHealth Symposium. His Google Scholar Citations reached 40,000+ with an h-index of 96. His top paper was cited 4,304+ times. He was selected as Highly Cited Researcher from 2018 to 2022. He got IEEE Communications Society Fred W. Ellersick Prize in 2017, the IEEE Jack Neubauer Memorial Award in 2019, and IEEE ComSoc APB Oustanding Paper Award in 2022. He is a Fellow of IEEE and IET.