DPCL: Constrative Representation Learning with Differential Privacy

Wenjun Li
Guangzhou University
Anli Yan
Hainan University
Di Wu
Guangzhou University
Taoyu Zhu
Guangzhou University
Teng Huang (huangteng1220@buaa.edu.cn)
Guangzhou University
Xuandi Luo
Guangzhou University
Shaowei Wang
Guangzhou University

Research Article

Keywords: Contrastive learning, Differential privacy, Sensitivity analysis, Privacy protection

Posted Date: April 6th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1516950/v1

DPCL: Constrative Representation Learning with Differential Privacy

Wenjun Li[1], Anli Yan[2], Di Wu[1], Taoyu Zhu[1], Teng Huang[1*], Xuandi Luo[1] and Shaowei Wang[1]

[1]Institute of Artificial Intelligence and Blockchain, Guangzhou University, 510006, Guangzhou, China.
[2]School of Cyberspace Security (School of Cryptology), Hainan University, 570228, Haikou, China.

*Corresponding author(s). E-mail(s): huangteng1220@buaa.edu.cn;

Abstract

With the proliferation of unlabeled data, increasing efforts have been devoted to unsupervised learning. As one of the most representative branches of unsupervised learning, contrastive learning has made great progress with its high efficiency. Unfortunately, privacy threats on contrastive learning have become sophisticated, making it imperative to develop effective technologies that are able to deal with such threats. To alleviate the privacy issue in contrastive learning, we propose some novel techniques based on differential privacy, which aim at reducing the high sensitivity of gradient in the private training caused by interactive contrastive learning. Specifically, we add differentially private protection to the connection point related to different per-example gradients, which decreases the sensitivity of the gradients significantly. Our experiments on SimCLR and Barlow Twins demonstrate the superiority of our approach with higher accuracy under the same privacy protection.

Keywords: Contrastive learning, Differential privacy, Sensitivity analysis, Privacy protection.

1 Introduction

Deep learning has ubiquitously been leveraged in a wide range of tasks, ranging from image recognition, natural language processing to object detection. To get a high-performance deep learning model, we need to collect a large amount of data [1, 2]. However, even if we obtain sufficient samples, there's another tricky issue. Due to the lack of sufficient prior knowledge, it is difficult or expensive to manually label. Therefore, unsupervised learning is favored by researchers. As one of the mainstream branches of unsupervised learning, contrastive learning (CL) has made great progress with its high-efficiency [3–6]. However, some studies have exposed potential privacy risks about contrastive learning, which shows that contrastive models are vulnerable to membership inference attacks and attribute inference attacks. For example, an attack model EncoderMI unexpectedly achieves high attack accuracy on the image encoder trained by contrastive learning [7]. Due to the absence of the class label, most contrastive learning models depend on the supervision of examples within the same batch or a memory bank [4]. The calculation of the loss in contrastive models heavily relies on the similarity of each pair of images, which leads to excessive interaction and the potential for privacy issues. To mitigate these privacy issues of contrastive learning, some researchers have made efforts such as Talos [8], which tackles the privacy problem by applying adversarial training on contrastive models. This method has played a certain role to some extent, but it is not a powerful and theoretically rigorous privacy protection method. Therefore, it is significant and urgent to explore new technologies to make the training of contrastive models safer.

To conduct private training on the contrastive model, some challenges need to be overcome: 1) The contrastive model generally has a complex network including encoder, projection head and prediction head, which inevitably suffer from high sensitivity of gradient; 2) Most contrastive models correlate augmented views of different images, i.e., the independence of examples is broken, and the change of a single example in batch processing may affect the gradient of other examples. Because differential privacy is a provable privacy guarantee [9] and has been applied extensively for privacy-preserving machine learning [10–13].

In this paper, we protect the privacy of contrastive model training by making rational use of differential privacy technology. Considering that the most influential factor in the model update is the gradient of the model, adding privacy protection to the gradient may be more efficient and general. However, the correlations among different examples incur a high sensitivity of gradient. To solve this problem, components that undermine the independence between examples should be replaced or given differential privacy protection. Benefiting from that, the sensitivity of the gradient can be decreased to a relatively small value. Furthermore, to give that connection point appropriate protection, we analysed its sensitivity, and theoretically demonstrated an upper bound.

In summary, we make the following contributions:

- We discuss the necessity of differentially private contrastive learning and analyzed the main challenge, i.e., the high sensitivity of gradient.
- We propose a novel contrastive learning method based on differential privacy, which makes the connection point between different per-example gradients differentially private to decrease the sensitivity of the gradient.
- We analyze the sensitivity of the similarity matrix which is the connection point in SimCLR and Barlow Twins respectively and provide the theoretical proof.
- The experiments on SimCLR and Barlow Twins indicate that our methods can effectively conduct differentially private training of the contrastive model.

The rest of the paper is organized as follows. In Section 2, we introduce reports on the work and background in the literature related to this paper. Section 3 shows how to apply differential privacy to contrastive learning and shows the specific practice on SimCLR and Barlow Twins for further illustration. Section 4 describes the analysis of privacy loss in private training in detail. Section 5 reports the experimental results of private training on SimCLR and Barlow Twins. Section 6 gives a brief conclusion of this work.

2 background and related work

In this section, we briefly introduce the background knowledge of contrastive learning, differential privacy, and privacy amplification by subsampling.

2.1 Contrastive Learning

Contrastive learning [14], which is the most representative self-supervised learning paradigm, aims to train a contrastive model which can generate expressive representations to perform downstream supervised machine learning tasks [8, 15, 16]. Specifically, contrastive learning is a framework that learns similar/dissimilar representations from data which are organized into similar/dissimilar pairs [17]. Self-supervised learning mainly uses pretext to mine its own supervised information from large-scale unlabeled data.

InfoNCE [18] is a commonly used contrastive loss function, defined as

$$\mathcal{L}_{q,k^+,\{k^-\}} = -\log \frac{\exp(q \cdot k^+/\tau)}{\exp(q \cdot k^+/\tau) + \sum_{k^-} \exp(q \cdot k^-/\tau)} \tag{1}$$

where q is a query representation, k^+ is a positive example derived from the augmentation version of the same image, k^- is a negative example derived from the augmentation of other examples, τ is a temperature hyper-parameter.

From Eq.(1), we know that InfoNCE maximizes the similarity between positive examples, and minimizes the similarity between positive examples and negative examples. InfoNCE is mainly used in the contrastive model with negative examples such as SimCLR [3], MoCo [4] or one of its variants, e.g.,

JCL [19]. Another example [20] differs from the work described above in that the core difference lies in the use of momentum update mechanisms and the way to form a negative example.

Furthermore, contrastive learning can also work without negative examples. BYOL [5] only uses positive examples' representations to compute mean squared error and gets comparable performance. W-MSE [21] uses a whitening transform to avoid degenerate solutions. Some contrastive models try to construct loss function from the perspective of the feature. Barlow Twins [6] proposes an objective function that tries to make the cross-correlation matrix computed from twin representations as close to the identity matrix as possible.

2.2 Differential Privacy

Privacy has attracted much attention in recent years [22–29], differential privacy (DP) is a rigorous mathematical framework that can guarantee a randomized algorithm behaves similarly on similar input databases.

Definition 1 (Differential Privacy [30]) *A randomized algorithm $\mathcal{M}$ with dimain $\mathbb{N}^{\mathcal{X}}$ is (ε, δ)-differentially private if for all $\mathcal{S} \subseteq Range(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{\mathcal{X}}$ such that $\|x - y\|_1 \leq 1$:*

$$\Pr[\mathcal{M}(x) \in S] \leq \exp(\varepsilon) \Pr[\mathcal{M}(y) \in S] + \delta$$

If $\delta = 0, \mathcal{M}$ is $\varepsilon-$differentially private. ϵ, δ are generally called privacy budget parameters. The smaller they are, the stronger the privacy guarantee is [31]. Other variants of DP use subtle different formulations (e.g., Rényi differential privacy (RDP) [32], Concentrated differential privacy (CDP) [33], Zero-Concentrated differential privacy (zCDP) [34].)

Generally, differential privacy has two main properties which are used widely [30]. First, the composition of several differentially private algorithms is still differentially private (but the privacy budget parameters are bound to decrease). Second, differential privacy is immune to *post-processing*: the composition of a data-independent mapping f with an (ϵ, δ)-differentially private algorithm $\mathcal{M}$ is also (ϵ, δ)-differentially private.

A standard paradigm to provide privacy-preserving approximations to function $f : \mathcal{D} \to \mathcal{R}^d$ is to add noise proportional to the sensitivity $\mathcal{S}_f$ of function f [35]. The sensitivity of a function gives an upper bound on how much we must perturb its output to preserve privacy [30].

Definition 2 (sensitivity [30]) *The sensitivity of a function $f : \mathbb{N}^{\mathcal{X}''} \to \mathbb{R}^k$, is:*

$$\Delta(f) = \max_{\substack{\|x-y\|_1 \leq 1 \\ x,y \in \mathbb{N}^{\mathcal{X}}}} \|f(x) - f(y)\|$$

where $\|\cdot\|$ denotes L_1 or L_2 norm.

The Laplace Mechanism and Gaussian Mechanism are two commonly used mechanisms. The former scale the noise to the L_1 sensitivity of function f and the latter scale the noise to the L_2 sensitivity of function f. In this paper, we choose the Gaussian mechanism which uses L_2 sensitivity. The Gaussian mechanism $\mathcal{M}(\cdot)$ adds zero-mean Gaussian noise with variance $\Delta^2\sigma^2$ in each coordinate of the output $f(x)$, such as $\mathcal{M}(x) = f(x) + \mathcal{N}(0, \Delta^2\sigma^2 I)$.

As mentioned above, one important property of DP relevant to this paper is that it composes gracefully over multiple access [36]. As a refinement of DP, Rényi differential privacy [32] works more naturally with the composition of multiple differential privacy mechanisms. To better understand how to track the overall privacy loss for the sequence of data accesses, we describe the RDP framework [32] and moments accountant [12] that make algorithm-dependent composition possible.

Definition 3 (Rényi differential privacy [32]) *A randomized mechanism $f : \mathcal{D} \mapsto \mathcal{R}$ is said to have Rényi differential privacy of order α, or (α, ϵ)-RDP for short, if for any adjacent $D, D' \in \mathcal{D}$ it holds that*

$$D_\alpha(f(D)\|f(D')) := \frac{1}{\alpha - 1} \log E_{x \sim f(D')}\left(\frac{f(D)(x)}{f(D')(x)}\right)^\alpha \leq \epsilon \tag{2}$$

The idea of moments accountant in [12] essentially keeps track of the evaluations of the cumulant generating function (CGF) at a list of fixed orders. We are able to get privacy loss of the composition of many mechanisms through Theorem 2.1, and convert the privacy loss to differential privacy parameters with Theorem 2.2.

2.3 Privacy amplification by subsampling

"privacy amplification by subsampling", which ensures that a differentially private mechanism running on a random subsample of a population provides higher privacy guarantees than on the entire population. It promotes most of the recent progress of differentially private deep learning. For example, Balle et al. [37] provided explicit privacy amplification bounds for the most common subsampling methods. Before describing subsampling methods, two neighboring relations that need to be known are the remove/add-one relation and substitute-one relation. The former means that we can obtain neighboring dataset X' by adding or removing one individual from X, the latter means obtaining neighboring dataset X' by replacing one data point from X with another arbitrary data point.

The following definitions introduce these methods.

Definition 4 (Poisson Subsampling [37]) *Given a dataset X, the Poisson subsampling outputs a subset of the data $\{x : x$ is independently sampled from S with probability $\gamma\}$.*

Definition 5 (Sampling Without Replacement [31]) *Given a dataset X of n points, the procedure subsample selects a random sample from the uniform distribution over all subsets of X of size m. The ratio $\gamma := m/n$ is defined as the sampling parameter of the subsampling procedure.*

3 Differentially Private Contrastive Learning

This section shows how to conduct private training on the contrastive model. A popular way to differentially private machine learning is using Stochastic Gradient Descent (SGD) with differentially private releases of gradients evaluated on mini-batches of a dataset. Differentially private contrastive learning can partly imitate this way while some specific problems may be occurred caused by the characteristic of contrastive learning.

We firstly present the basic components of the contrastive model's private training, then concretely show how to differentially private train specific contrastive models such as SimCLR and Barlow Twins.

3.1 Basic components of private training

To protect the privacy of training data, DPSGD needs to perform two operations: (1) limit the sensitivity of gradient by clipping the norm of per-example gradients, and (2) add noise to the gradient of a batch before updating the model's parameters. These two operations can be detailed as: computing per-example gradient, clipping per-example gradient, adding noise to the sum of the per-example gradient for privacy protection, using this noisy gradient to update the model's parameters. At last, analyzing the privacy loss of this private training.

3.1.1 Per-example gradient computation

In general, gradients are computed for a batch of examples. To limit the influence of every example, the gradient with respect to every example $\nabla_\theta \mathcal{L}(\theta, x)$ ($\mathrm{g}(x)$ for short) is needed. The per-example-gradient operator, proposed by Goodfellow, can support batched computation for the loss function $\mathcal{L}$ while under the premise of each x_i is singly connected to $\mathcal{L}$.

However, a common phenomenon in contrastive learning is that model's loss leverages relationships between different examples' representation. Therefore, the gradient of each example cannot be calculated by sequentially inputting one example into the model (also called micro-batch). Fortunately, Goodfellow [38] showed that a fully-connected network's per-example gradients can be efficiently computed using the auto-differentiation library in deep learning frameworks. Jaewoo et al. [39] proposed a fast gradient computation method for convolutional layer and normalization layer. With the assistance of these methods, we can get the true per-example gradients efficiently. More details about fast gradient computation are referred to [39]. Note that the BatchNorm

layer is incompatible with differential privacy, the BatchNorm layer in the contrastive model's network should be replaced by other normalization layers. We use the GroupNorm layer instead of the BatchNorm layer in this paper.

3.1.2 Per-example gradient clipping

Once we have obtained all per-example gradients, the next step is clipping the per-example gradients as:

$$\bar{\mathrm{g}}(x) = \begin{cases} \mathrm{g}(x) & \text{if } \|g(x)\|_2 \leq C \\ \mathrm{g}(x)/(\frac{\|g(x)\|_2}{C}) & \text{if } \|g(x)\|_2 > C \end{cases}$$

The C denotes the clipping threshold. After clipping, all the per-example gradients' norms are at most C. The contrastive model has a relatively complex network compared to the Abadi et al. [12]. It is inevitable to get a high-dimension gradient with a high norm, which makes it challenging to set the clipping threshold appropriately. Abadi et al. [12] mentioned that clipping threshold can be the median of the norms of the unclipped gradients during training. We think it may incur an extra privacy budget. Therefore, the clipping threshold in this paper is set according to the observation of gradients in private training.

3.1.3 Adding noise to the gradient

Adding noise to the gradient needs to analyze its sensitivity. The gradient is calculated as $f(B) = \sum_{x \in B} \bar{\mathrm{g}}(x)$ (B represents a mini-batch of examples). We have mentioned that per-example gradients within the same batch are correlated, which means that the sensitivity of the gradient is a high value. The gradient's sensitivity Δ can be calculated by $\Delta = \|f(B) - f(B')\|_2$. (B and B$'$ are two neighbouring mini-batch).

The relationship between sensitivity and clipping threshold depends on examples' relation. The detailed analysis are presented in the private training of SimCLR and Barlow Twins. Considering that under the same privacy budget, Gaussian mechanism adds less noise than Laplace mechanism, we select the Gaussian mechanism $M(\cdot)$ to make the gradient differentially private, $M(\cdot)$ is as

$$\mathcal{M}(B) = f(B) + \mathcal{N}(0, \Delta^2 \sigma^2 I)$$

Most contrastive models' loss is not a simple average of a batch of individual loss, so the sensitivity analysis of gradients is not similar to supervised learning. The concrete analysis is shown in the latter subsection.

3.1.4 Updating model parameters

Once we obtain the noisy gradient, The next step is to use the noise gradient to update the parameters of the model. As for the choice of the optimizer, original SGD or SGD with momentum is available. Using the first moment of gradients are incur extra privacy loss. With the post-processing property of differential privacy, using the previous noisy gradients to calibrate the current gradient still satisfies differential privacy [40].

3.1.5 Privacy accounting

The last step in private training is analyzing the overall privacy loss. If the accumulated privacy loss exceeds the privacy budget, the training process should be terminated. The composability of differential privacy makes it possible to implement an "accountant" procedure that calculates the privacy loss at every iteration. We use Rényi differential privacy, which is natural for composability to analyze the privacy loss of one training step. Because of the existence of subsampling, we need to consider the "privacy amplification". There are two ways to analyze the overall privacy loss: (1) first, we obtain RDP parameters of the subsampled Gaussian mechanism, then use the adaptive composition theorem of RDP to get the overall RDP parameters. Finally, we use the conversion equation to obtain the parameters of DP; (2) we obtain the RDP parameters of the Gaussian mechanism and use the privacy amplification of RDP to get the RDP parameters of the subsampled mechanism. Then, we compose these mechanisms and convert the parameters of RDP to DP.

Note that the symbols used in SimCLR and Barlow Twins are a little abused, so the meaning of the symbol subjects to the current definition.

3.2 Private Training of SimCLR

SimCLR [3] can be viewed as a representation of contrastive learning with negative examples. This subsection shows how to differentially private train the SimCLR model. We briefly introduce the framework of SimCLR.

A base encoder $f(\cdot)$ which extracts representation vectors from augmented data examples, no constraint for the network architecture but commonly use ResNet. **A projection head** $g(\cdot)$ that maps representations to the space where contrastive loss is applied. **A contrastive loss function** $\mathcal{L}$ defined for a contrastive prediction task, the loss function $\mathcal{L}$'s formulation is:

$$\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)] \tag{3}$$

and $\ell(i, j)$ is defined as

$$\ell(i, j) = -\log \frac{\exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(sim(z_i, z_k)/\tau)} \tag{4}$$

where $\mathbb{1}_{[k \neq i]} \in \{0, 1\}$ is an indicator function evaluating to 1 if $k \neq i$ and τ denotes a temperature parameter.

It has been mentioned above that relationship among a batch of examples results in a high sensitivity of gradient. But some improvements can be done to weaken this relationship, which can further decrease this sensitivity. In Eq. (4), the similarities between different representations are the connection point that makes per-example gradients correlated. So if this similarity can be made differentially private, the relationship between examples will be significantly weakened. For the private training of the SimCLR model, we adopt the original DPSGD and our improved DPSGD, which leverages both private gradient and private similarity.

There are two notes: one is that sensitivity partly depends on neighboring relations, and this paper adopts substitute-one relation (also called replace-one relation), the other is the SimCLR uses two augmented views of one example as positive examples. Therefore, a batch of examples with size $|B|$ generates a batch of augmented views with size $2|B|$, i.e., there are $2|B|$ per-example gradients in one training step.

3.2.1 Original-DPSGD

The gradient of a batch can be seen as a gradient query function. We know from Eq. (4) that a single example's change can not change other examples' representations, but can change other per-example gradients. Hence two neighboring batches of examples B and B', which differ in one example, may have different similarity values and losses. Then it will generate two batches of totally different per-example gradients g_i and g'_i. So the sensitivity of gradient query function Δ_g can be analyzed as

$$\begin{aligned}
\Delta_g &= \|g_B - g_{B'}\|_2 \\
&= \|\sum_{i=1}^{n} g_i - \sum_{i=1}^{n} g'_i\|_2 \\
&\leq \|\sum_{i=1}^{n} g_i\|_2 + \|\sum_{i=1}^{n} g'_i\|_2 \\
&\leq \sum_{i=1}^{n}(\|g_i\|_2) + \sum_{i=1}^{n}(\|g'_i\|_2) \\
&\leq 2 \cdot n \cdot C = 4|B|C
\end{aligned}$$

where n represents the number of augmented views, so n is $2|B|$, C is the clipping threshold. The first inequality follows from the triangle inequality, the last inequality follows the fact that each per-example gradient's norm is at most C. It can be known from the inequality that this sensitivity is related to the number of a batch of examples, which means a bigger batch size will lead to higher sensitivity of gradient.

We use a Gaussian mechanism $\mathcal{M}_g$ to make the model's gradient differentially private as

$$\mathcal{M}_g(g_B) = g_B + \mathcal{N}(0, \Delta_g^2 \sigma_g^2 I)$$

where σ_g denotes the noise scale of gradient.

$M_g(\cdot)$ adds independently drawn random noise distributed as $\mathcal{N}(0, \Delta_g^2 \sigma_g^2 I)$ to each dimension of g_B. According to the observation of $M_g(\cdot)$, even the noise scale of the gradient σ_g is small, the overall variance of noise $\Delta_g^2 \sigma_g^2$ is big, which results that original DPSGD is hard to conduct efficient private training of the SimCLR model.

3.2.2 DPSGD with DP-similarity

The similarity is the connection point that makes per-example gradients correlated, so the core to decrease the sensitivity of gradient is weakening the relationships between different per-example gradients. To this end, we propose to make the similarity differentially private by adding noise. As we mentioned above, the similarity is calculated by matrix operation, hence the real operation is on similarity matrix S. In general, the SimCLR model gives a L_2 normalization to representations output by the projection head. Let Z represents the representations of a batch of augmented batches (i.e.,$|Z| = 2|B|$), the similarity matrix S can be calculated as

$$S(Z) = Z^T Z$$

Let B and B' represent two neighboring batches of examples which has the same size but differ in one example. Through data augmentation and encoding, we can obtain respective representations Z and Z', cause every example has two augmented views, so Z and Z' differ in two representations. To make the similarity matrix differentially private, we first prove the upper bound of the similarity matrix's sensitivity.

Theorem 1 *Let $Z = \{Z_1, \cdots, Z_N\}$ and $\|Z_i\|_2 = 1$, then S(Z) has L_2 sensitivity Δ_S bounded above by $4\sqrt{(|Z| - 1)}$.*

Proof The sensitivity of the similarity matrix can be calculated as

$$\begin{aligned} \Delta_S &= \|S(Z) - S(Z')\|_2 \\ &= \|Z^T Z - Z'^T Z'\|_2 \end{aligned}$$

while Z and Z' differ in two representations, then $Z^T Z$ and $Z'^T Z'$ differ in two rows and two columns, which are $(4|Z| - 4)$ elements, so there are $(4|Z| - 4)$ different similarities. Following the fact that the dot product between two l_2 normalized representations is at most 1, which also means the similarity is at most 1, then the difference between one similarity from $Z^T Z$ and one similarity from $Z'^T Z'$ is at most

2, so the sensitivity of similarity matrix Δ_S is at most $2\sqrt{(4|Z|-4)} = 4\sqrt{(|Z|-1)}$. □

It can be known from **Theorem 1** that the sensitivity of similarity matrix Δ_S is $4\sqrt{(2|B|-1)}$, σ_S denotes the noise scale of similarity matrix, and the Gaussian mechanism $M_S(Z)$ that make similarity matrix differentially private is as

$$M_S(Z) = S(Z) + \mathcal{N}(0, \Delta_S^2 \sigma_S^2 I)$$

That means M_S adds independently drawn random noise distributed as $\mathcal{N}(0, \Delta_S^2 \sigma_S^2 I)$ to each output of $S(Z)$

Through the noisy similarity matrix, we can significantly weaken the relationships between different per-example gradients, which means the change of single examples in a batch has limited influence on other per-example gradients.

With the assistance of a noisy similarity matrix, it can be thought that two neighboring batches of examples B and B' obtain two similar similarity matrixes and two similar losses. Furthermore, the per-example gradient of all examples excluding the example being changed can be seen unchanged. Hence two neighbouring batches of examples have two batches of per-example gradients which only differ in two per-example gradients, while each example has two augmented views. So the sensitivity of gradient query function Δ_g' can be analyzed as

$$\begin{aligned}
\Delta_g' &= \|g_B' - g_{B'}'\|_2 \\
&= \|\sum_{i=1}^{n} g_i - \sum_{i=1}^{n} g_i'\|_2 \\
&= \|2(g_i - g_i')\|_2 \\
&= 2\|g_i - g_i'\|_2 \\
&\leq 2(\|g_i\|_2 + \|g_i'\|_2) \\
&\leq 4 \cdot C
\end{aligned}$$

Then using a Gaussian mechanism $\mathcal{M}_g'$ to make the model's gradient differentially private as

$$\mathcal{M}'_g(g_B) = g_B' + \mathcal{N}(0, \Delta_g'^2 \sigma_g'^2 I)$$

where σ_g' denotes the noise scale of gradient.

Original DPSGD has a sensitivity of gradient $4|B|C$, using DPSGD with DP-similarity has saved a factor of $2|B|$ in L_2 sensitivity. Although some privacy budgets are allocated for the similarity matrix, means fewer privacy

budgets are allocated for the model's gradient. Although σ'_g is a little bigger than σ_g, $\Delta'^2_g \sigma'^2_g$ is still smaller than $\Delta^2_g \sigma^2_g$.Hence using DPSGD with DP-similarity to conduct efficient private training of the SimCLR model is promising.

3.3 Private Training of Barlow Twins

The above part introduces how to carry out the differentially private training of a contrastive model under the supervision of comparison of positive examples and negative examples. However, the contrastive model can also work without negative examples, such as Barlow Twins[6]. In this subsection, we analyze how to conduct differentially private training of Barlow Twins.

The framework of Barlow Twins is similar to SimCLR. Specifically, it generates two augmented views for all examples of a batch B subsampled from a dataset, then feeds them to a deep network.

Different from other methods for the contrastive model, Barlow Twins uses a innovative loss function $\mathcal{L}_{\mathcal{BT}}$:

$$\mathcal{L}_{\mathcal{BT}} \triangleq \sum_i (1 - \mathcal{C}_{ii})^2 + \lambda \sum_i \sum_{j \neq i} \mathcal{C}_{ij}^2 \tag{5}$$

Where λ is a positive constant that balances the importance of the first and second losses. $\mathcal{C}$ is the cross-correlation matrix computed between the outputs of the two identical networks along the batch dimension:

$$\mathcal{C}_{ij} \triangleq \frac{\sum_b z^A_{b,i} z^B_{b,j}}{\sqrt{\sum_b (z^A_{b,i})^2} \sqrt{\sum_b (z^B_{b,j})^2}} \tag{6}$$

Where b indexes batch samples and i and j index the vector dimension of the network's outputs. Z^A and Z^B have preprocessed features with a mean of zero, which leverage the mean of each feature. $\mathcal{C}$ is square matrix with the dimensionality of the network's output, and with values comprised -1 (i.e., perfect anti-correlation) and 1 (i.e., perfect correlation).

The denominator of Eq (6) can be seen as the variance of ith and jth features over the batch respectively. The preprocessing procedure leverage both the mean and variance of each feature over the batch. Therefore single example's change in Barlow Twins changes other per-example gradients, which means Barlow Twins have a gradient with high sensitivity.

For the private training of Barlow Twins, we adopt the original DPSGD and our improved DPSGD respectively. The former only leverages private gradient while the latter leverages private gradient, private mean and private variance. The sensitivity analysis of gradient in Barlow Twins is similar to SimCLR. We focus on the sensitivity analysis of the two connection points of Barlow Twins.

3.3.1 Original-DPSGD

According to the analysis above, a single example change in a mini-batch result in all per-example gradient changes. Hence, two neighboring batches of examples B and B' have different mean and variance for each feature and loss, which generate two batches of totally different per-example gradients g_i and g'_i. Therefore, the sensitivity of the gradient query function Δ_g in Barlow Twins is the same with SimCLR, i.e., $4|B|C$. We leverage a Gaussian mechanism $\mathcal{M}_g(\cdot)$ to make the model's gradient differentially private such as

$$\mathcal{M}_g(g_B) = g_B + \mathcal{N}(0, \Delta_g^2 \sigma_g^2 I)$$

Due the overall variance of noise $\Delta_g^2 \sigma_g^2$ is big, the original DPSGD is hard to conduct efficient private training of Barlow Twins.

3.3.2 DPSGD with DP-mean and DP-variance

The mean and variance are the connection points that make per-example gradients correlated. To weaken this relationship, we propose to make the mean and variance differentially private. For the sake of argument, we give a L_2 normalization to representations output by the projection head. Denoting the output of projection as Z, which has N representations, function $\mu(\cdot)$ calculates the mean of each feature, function $\sigma(\cdot)$ calculates the variance of each feature, $\mu(\cdot)$ and $\sigma(\cdot)$ are defined as

$$\mu(Z) = \frac{1}{N}(\sum_{i=1}^{N} Z_i) \tag{7}$$

$$\sigma(Z) = \frac{\sum_{i=1}^{N} Z_i^2 - N \cdot (\mu(Z))^2}{N-1} \tag{8}$$

Theorem 2 *Let Z and Z' denotes two neighboring representations, which have the same size N but differ in one representation, and each representation is L_2 normalized. Function $\mu(Z)$ has L_2 sensitivity Δ_μ bounded above by $\frac{2}{N}$.*

Proof The sensitivity of $\mu(Z)$ can be calculated as

$$\begin{aligned}
\Delta_\mu &= \|\mu(Z) - \mu(Z')\|_2 \\
&= \frac{1}{N} \| \sum_{i=1}^{N} (z_i) - \sum_{i=1}^{N} (z'_i) \|_2 \\
&= \frac{1}{N} \|z_i - z'_i\|_2 \\
&\leq \frac{1}{N} (\|z_i\|_2 + \|z'_i\|_2) \\
&\leq \frac{2}{N}
\end{aligned}$$

The first inequality follows the triangle inequality, and the second inequality follows the fact that each representation's L_2 norm is 1. □

According to **Theorem 2**, the L_2 sensitivity of mean representation $\mu(Z)$ is $\frac{2}{N}$, σ_μ is the noise scale of function $\mu(Z)$. We leverage a Gaussian mechanism $M_\mu(Z)$ to make the mean representation $\mu(Z)$ differentially private as

$$M_\mu(Z) = \mu(Z) + \mathcal{N}(0, \Delta_\mu^2 \sigma_\mu^2 I)$$

That means M_μ adds independently drawn random noise distributed as $\mathcal{N}(0, \Delta_\mu^2 \sigma_\mu^2 I)$ to each output of $\mu(Z)$.

The sensitivity analysis of function $\sigma(\cdot)$ can be converted to another function SSE$(\cdot)$ which is referred to as the method in [41] and defined below.

$$SSE(X) = \sum_{i=1}^{N} x_i^2 - N\bar{x}^2$$

Where x_i represents element of X, and $\bar{x}$ represents the mean of X.

The conversion equation between $\sigma(Z)$ and $SSE(Z)$ is as

$$\sigma(Z) = \frac{SSE(Z)}{N-1} \tag{9}$$

That means a private variance can be obtained through a private $SSE(Z)$.

Theorem 3 *Let Z and Z' denotes two neighboring representations, which have the same size N but differ in one representation, and each representation is L_2 normalized. Function $SSE(Z)$ has L_2 sensitivity bounded above by 6.*

Proof Assuming Z and Z' differ in the first representation, the L_2 sensitivity of function $SSE(Z)$ can be calculated as

$$\begin{aligned}
\Delta_{SSE} &= \|SSE(Z) - SSE(Z')\|_2 \\
&= \|\sum_{i=1}^{N} z_i^2 - N\bar{z}^2 - (\sum_{i=1}^{N} z_i'^2 - N\bar{z}'^2)\|_2 \\
&= \|z_1^2 - z_1'^2 - N(\bar{z}^2 - \bar{z}'^2)\|_2 \\
&\leq \|z_1^2\|_2 + \|z_1'^2\|_2 + N\|(\bar{z}^2 - \bar{z}'^2)\|_2 \\
&\leq \|z_1^2\|_2 + \|z_1'^2\|_2 + N \cdot \frac{4}{N} \\
&\leq \|z_1\|_2 + \|z_1'\|_2 + 4 \\
&\leq 6
\end{aligned}$$

The first inequality follows from the triangle inequality, and the third inequality follows the fact that a L_2 normalized representation's square has L_2 norm at most 1, the second inequation follows the result below:

$$
\begin{aligned}
&\|\bar{z}^2 - \bar{z}'^2\|_2 \\
&= \frac{1}{N^2}\|(z_1 + \cdots + z_N)^2 - (z'_1 + \cdots + z_N)^2\|_2 \\
&= \frac{1}{N^2}\|z_1(z_1 + 2\sum_{i=2}^{N}(z_i)) - z'_1(z'_1 + 2\sum_{i=2}^{N}(z_i))\|_2 \\
&\leq \frac{1}{N^2}(\|z_1(z_1 + 2\sum_{i=2}^{N}(z_i))\|_2 + \|z'_1(z'_1 + 2\sum_{i=2}^{N}(z_i))\|_2) \\
&\leq \frac{1}{N^2}(\|2(\sum_{i=1}^{N}(z_1 z_i))\|_2 + \|2(\sum_{i=1}^{N}(z'_1 z_i))\|_2) \\
&\leq \frac{2}{N^2}(\sum_{i=1}^{N}\|z_1 z_i\|_2 + \sum_{i=1}^{N}\|z'_1 z_i\|_2) \\
&\leq \frac{2}{N^2}(N + N) \leq \frac{4}{N}
\end{aligned}
$$

The first and third inequality follows from the triangle inequality, the fourth inequality follows from the fact that the element-wise product of two L_2 normalized representations has L_2 norm at most 1. □

According to **Theorem 3**, the L_2 sensitivity of function $SSE(Z)$ is 6, σ_{SSE} is its noise scale. We leverage a Gaussian mechanism M_σ to make the variance differentially private, the procedure can be formulated as

$$M_\sigma(Z) = \frac{\max(0, SSE(Z) + \mathcal{N}(0, \Delta^2_{SSE}\sigma^2_{SSE}I))}{N-1}$$

Where the variance is non-negative, the $SSE(Z)$ adds with Gaussian noise may be negative, so the minimum result should be truncated to zero.

With the private mean and private variance, the relationships between different per-example gradients are significantly weakened. The Barlow Twins with DP-mean and DP-variance has the same gradient sensitivity as SimCLR with DP-similarity, i.e., $4 \cdot C$. Then, we use a Gaussian mechanism $\mathcal{M}'_g$ to make the model's gradient differentially private as

$$\mathcal{M}'_g(g_B) = g'_B + \mathcal{N}(0, \Delta'^2_g\sigma'^2_g I)$$

Where σ'_g denotes the noise scale of the gradient.

Noted that noisy variance can't be obtained by using noisy mean, although differential privacy is immune to post-processing. Variance uses both noisy mean and the original data to get the difference between data and its means. Therefore, variance still needs extra differential privacy protection.

4 Privacy Analysis

A fundamental component in a differentially private stochastic gradient descent algorithm is iteratively sampling a mini-batch of examples in the training dataset as the model's input. In privacy analysis of differentially private training algorithms, different subsampling methods have different privacy analysis frameworks. Subsampling methods include Poisson subsampling, sampling without replacement, and sampling with replacement, etc. [37]

Roughly speaking, there exist two routes to analyze the privacy of subsampled a Gaussian mechanism which is the mechanism we use in this paper, one is directly analyzing Rényi differential privacy of subsampled Gaussian mechanism [42], the other is analyzing the subsampled Rényi differential privacy [31].

4.1 Framework of privacy analysis

In this paper, we fix the batch size of a mini-batch, which is equivalent to a fixed size of a sample. For the sake of rigor, we adopt the analysis strategy of "sampling without replacement", proposed by Wang et al. [31]. The complete analysis procedure can be done according to the order: obtaining the RDP of the Gaussian mechanism, amplifying RDP by subsampled, composing the RDP of multiple mechanisms, and converting the RDP to DP. The details are shown below.

4.1.1 RDP of Gaussian mechanism

The RDP guarantees of the Gaussian mechanism are given by corollary 3 in [32].

Corollary 1 ([32],Corollary 3) *If function f has sensitivity 1,then the Gaussian mechanism $\mathbf{G}_\sigma f$ satisfies $(\alpha, \alpha/(2\sigma^2))$-RDP, where σ^2 represents the variance of the Gaussian mechanism.*

In general, RDP is represented as a function of α, i.e., $\epsilon(\alpha) = \alpha/(2\sigma^2)$. This paper aims to explore different differences in private training performance. The noise-multiplier is set before training, i.e., the ratio of the standard deviation of additive Gaussian noise to the L_2 sensitivity of additive Gaussian noise. The RDP of the Gaussian mechanism can be calculated by corollary1.

4.1.2 Privacy amplification for RDP

The widespread use of mini-batch SGD urges us to consider subsampling. This paper adopts a fixed size of mini-batch, which corresponds to the Theorem 9 in [31].

Theorem 4 ([31],Theorem 9) *Given a dataset of n points drawn from a domain* $\mathcal{X}$ *and a mechanism* $\mathcal{M}$ *that takes an input from* $\mathcal{X}^m$ *for* $m \leq n$. *Let the randomized algorithm* $\mathcal{M} \circ$ *subsample be defined as: (1) subsample without replacement m datapoints of the dataset (sampling parameter* $\gamma = m/n$*),and (2) apply* $\mathcal{M}$ *to the subsampled dataset. For all integers* $\alpha \geq 2$, *if* $\mathcal{M}$ *obeys* $(\alpha, \epsilon(\alpha)) - RDP$, *then the subsampled mechanism* $\mathcal{M} \circ$ *subsample obeys* $(\alpha, \epsilon'(\alpha)) - RDP$ *where,*

$$\begin{aligned}\epsilon'(\alpha) \leq \frac{1}{\alpha-1}\log\Bigg(1+\gamma^2\binom{\alpha}{2}\min\Big\{4(e^{\epsilon(2)}-1),\\ e^{\epsilon(2)}\min\{2,(e^{\epsilon(\infty)}-1)^2\}\Big\}\\ +\sum_{j=3}^{\alpha}\gamma^j\binom{\alpha}{j}e^{(j-1)\epsilon(j)}\min\{2,(e^{\epsilon(\infty)}-1)^j\}\Bigg)\end{aligned} \tag{10}$$

As mentioned previously, this paper adopts the Gaussian mechanism, it can be seen from Corollary(1) that the Gaussian mechanism does not have a bounded $\epsilon(\infty)$ term. Eq(10) can be further simplified as

$$\begin{aligned}\epsilon'(\alpha) \leq \frac{1}{\alpha-1}\log\Bigg(1+\gamma^2\binom{\alpha}{2}\min\Big\{4(e^{\epsilon(2)}-1),2e^{\epsilon(2)}\Big\}\\ +\sum_{j=3}^{\alpha}\gamma^j\binom{\alpha}{j}2e^{(j-1)\epsilon(j)}\Bigg)\end{aligned} \tag{11}$$

The RDP of a subsampled Gaussian mechanism without composition is already obtained. The next step is considering the composition of multiple Gaussian mechanisms.

4.1.3 Composition of RDP

This step is to consider the composition of Gaussian mechanisms, such as the composition of the same Gaussian mechanism in different steps or different Gaussian mechanisms in one step. The original DPSGD only needs to consider the cumulative privacy loss of gradient, but DPSGD with extra differential privacy protection needs to consider the cumulative privacy loss of gradient and connection point. In the framework of RDP, these two kinds of composition leverage the same composition theorem in [32].

Lemma 1 ([32],Proposition 1) *Let* $f : \mathcal{D} \mapsto \mathcal{R}_1$ *be* (α, ϵ_1)*-RDP and* $g : \mathcal{R}_1 \cdot \mathcal{D} \mapsto \mathcal{R}_2$ *be* (α, ϵ_2)*-RDP, then the mechanism defined as (X,Y), where* $X \leftarrow f(D)$ *and* $Y \leftarrow g(X,D)$, *satisfies* $(\alpha, \epsilon_1 + \epsilon_2)$*-RDP.*

Through this lemma, we can know that the overall RDP of differentially private training is simply the sum of the RDP of each Gaussian mechanism, which includes the same differential privacy mechanism and different differential privacy mechanisms.

4.1.4 Conversion from RDP to DP

The overall RDP of differentially private training is already obtained, the last step is converting the RDP to DP. In [32], Mironov gives a conversion equation like below.

Lemma 2 ([32],Proposition 3) *If f is an (α, ϵ)-RDP mechanism, it also satisfies $(\epsilon + \frac{\log 1/\delta}{\alpha-1}, \delta)$-differential privacy for any $0 < \delta < 1$.*

Using privacy regions, Balle et al. [43] refined Mironov's conversion law from RDP to DP in a simple way, proposed the following better conversion law:

Theorem 5 ([43],Theorem 21) *If a mechanism $\mathcal{M}$ is (α, ϵ)-RDP, then it is $(\epsilon + \log(\frac{(\alpha-1)}{\alpha}) - \frac{(\log \delta + \log \alpha)}{(\alpha-1)}, \delta)$-differential privacy for any $0 < \delta < 1$.*

Accomplishing the conversion from RDP to DP means the end of the privacy analysis. The next part shows how to conduct privacy analysis in a specific contrastive model such as SimCLR and Barlow Twins.

4.2 Privacy analysis of SimCLR

The privacy analysis of the SimCLR's private training can be forwarded as the steps we mentioned above. Due to that original DPSGD and DPSGD with DP-similarity have a difference in privacy loss, we analyzed the privacy loss respectively.

4.2.1 Original DPSGD

We only add noise to the sum of per-example gradients. Once the noise scale of gradient σ, sampling probability γ and the number of steps T are set. First, the RDP $\epsilon(\alpha, \sigma)$ of the Gaussian mechanism with noise scale σ for each order α is obtained through the corollary 3 in [32]. Second, the RDP $\epsilon'(\alpha)$ of the subsampled Gaussian mechanism with sampling probability γ for each α is calculated by Theorem 9 in [31]. Third, the RDP of T subsampled Gaussian mechanisms for each α is $T \cdot \epsilon'(\alpha)$. Lastly, through Theorem 21 in [43] calculating a series of DP i.e. $\{\epsilon_{(\alpha,\delta)}\}$ for each α and the same given δ and picking the smallest ϵ as the final privacy loss of this differentially private training.

4.2.2 DPSGD with DP-similarity

We add noise not only to the sum of per-example gradients but also to the similarity matrix. There are two-part to privacy loss. The noise scale of gradients and similarity matrix is set to be σ_1 and σ_2 respectively. First, the RDP of Gaussian mechanism applying to gradient $\epsilon_1(\alpha, \sigma_1)$ and similarity

matrix $\epsilon_2(\alpha, \sigma_2)$ are obtained according to the corollary 3 in [32]. Second, the subsampled version $\epsilon_1'(\alpha)$, $\epsilon_2'(\alpha)$ are calculated by Theorem 9 in [31]. Third, the overall RDP of T subsampled Gaussian mechanisms for each α is $(T \cdot (\epsilon_1'(\alpha) + \epsilon_2'(\alpha)))$.Lastly, the DP of differentially private SimCLR training with DP-similarity is picked among a series of DP $\{\epsilon_{(\alpha,\delta)}\}$, which are calculated through Theorem 21 in [43].

4.3 privacy analysis of Barlow Twins

To conduct a privacy analysis of Barlow Twins's private training, the original DPSGD and DPSGD with DP-mean and DP-variance should be discussed respectively.

4.3.1 Original DPSGD

In consideration of Barlow Twins with DPSGD is similar to SimCLR with DPSGD, so we don't state here again.

4.3.2 DPSGD with DP-mean and DP-variance

In this case, we add Gaussian noise to the mean, the variance and the gradient. The noise scale of mean, variance and gradient are set to be σ_1, σ_2, σ_3 respectively. According to the framework of privacy analysis, the RDP of corresponding Gaussian mechanisms is $\epsilon_1(\alpha, \sigma_1)$, $\epsilon_2(\alpha, \sigma_2)$, $\epsilon_3(\alpha, \sigma_3)$. The RDP of corresponding subsampled Gaussian mechanisms is $\epsilon_1'(\alpha)$, $\epsilon_2'(\alpha)$, $\epsilon_3'(\alpha)$. The overall RDP of Gaussian mechanisms' composition is $(T \cdot (\epsilon_1'(\alpha) + \epsilon_2'(\alpha) + \epsilon_3'(\alpha)))$. The DP is obtained by picking the smallest ϵ through the conversion equation.

5 implementation

According to the analysis above, we can theoretically analyze the benefits and drawbacks of different differentially private training in contrastive learning. With the same privacy budget, The one who only adds noise to the gradient can have all the privacy budget, which means that the noise scale of the gradient may be smaller. However, the existence of high-sensitivity leads to high-variance noise of gradients, which significantly reduces the performance of the model and even makes the model unavailable; Another method weakens the relationship between different per-example gradients, which can greatly reduce the sensitivity of the gradient and then reduce the variance of the additive noise of the gradient. A part of the privacy budget is allocated to the connection point, the privacy budget of the gradient is smaller and the noise scale of the gradient is be larger.

This section conducts experiments on different differentially private training methods within the same privacy budget. The contrastive model we use is still SimCLR. and Barlow Twins.

5.1 Common Experimental Setup

There are some common settings for the differentially private training of SimCLR and Barlow Twins.

- **Dataset.** This paper aims to investigate different contrastive models' performances in different differentially private training. From a computational efficiency perspective, we conduct our experiments on the CIFAR10 dataset [44]. The CIFAR10 dataset contains 60,000 colour images from 10 labels and is normally partitioned into 50,000 training images and 10,000 test images, the size of each image is 32×32.
- **Data Augmentation.** Unless otherwise specified, for the experiments of SimCLR and Barlow Twins on CIFAR10, we use the same augmentations as [3] but override the input size and blur. We use random crop, resize(with random flip), and color distortions, but without gaussian blur due to the low resolution of CIFAR-10 images.
- **Encoder.** A common component of the contrastive model is the base encoder. The base encoder we used for SimCLR and Barlow Twins in this paper is ResNet-18. Although the base encoder of SimCLR in [3] and Barlow Twins in [6] is ResNet-50. As the complex network calculates the per-example gradient with high computational complexity, we select a relatively simple network like ResNet-18 as the base encoder.
- **Evaluation Protocol.** The most common evaluation protocol for unsupervised feature learning is based on freezing the network encoder after unsupervised pre-training and then obtaining the accuracy of a k-nearest neighbors classifier (KNN,$k = 5$) [21].
- **Privacy Parameters.** The privacy parameters of private training for SimCLR and Barlow Twins (ϵ, δ) is $(1000, 10^{-5})$, while ϵ is privacy budget and δ is breaking probability. Once the privacy loss exceeds the privacy budget, the training process will be terminated.
- **Others.** The batch size $|B|$ is 128, the optimizer is Stochastic Gradient Descent with Momentum, the momentum is 0.9. The sampling method is sampling without replacement $|B|$ datapoints of the dataset, and every sampling is independent, which can't be implemented via random reshuffling.

As for the setting of the privacy budget, we would like to give an explanation, which is the actual privacy in practical scenarios is greater than what can be shown theoretically [45]. So to achieve greater model utility, we set a relative big privacy budget.

We use the lightly framework to construct SimCLR and Barlow Twins, the framework is proposed by Susmelj et al. [46], and the fast per-example gradient computation we used is proposed by Jaewoo et al. [39]. The code is available at https://github.com/ppmlguy/fastgradclip.

5.2 SimCLR

Architecture Setting. A ResNet-18 as a base encoder to encode the image into a 512-dimensional representation, and a 2-layer MLP projection head to project the representation to a 128-dimensional latent space, the former has 512 hidden units and the latter has 128 hidden units. The batch normalization layer in ResNet-18 is replaced with a group normalization layer and weight standardization is used in the projection head. Moreover, we find that weight standardization added to the projection head can improve the model's performance.

Implementation details. The learning rate is 0.1, weight decay = 1e-6, clipping threshold $C = 0.02$. While the CIFAR10 dataset has 50,000 training images, the sampling ratio of each mini-batch $q = 0.00256$, and the number of epochs E is 200 (so the number of steps is $T = E/q$).

As the privacy loss is independent of the algorithm, once the sampling rate, privacy budget, and running epochs are deterministic, the noise scale for gradient and similarity matrix can be obtained.

5.2.1 Original DPSGD

According to the analysis in Section 3, the sensitivity of gradients is $(4 \cdot |B| \cdot C)$, then the minimum variance of noise to satisfy the privacy budget is $\sigma = 0.3812$.

5.2.2 DPSGD with DP-similarity

With the assistance of the noisy similarity matrix, the gradient's sensitivity has been significantly decreased to $(4 \cdot C)$, which saves a factor of $|B|$ compared to the original DPSGD. We allocated the privacy budget equally to the gradient and similarity matrix. So the noise scale of similarity matrix and gradient is set to be the same value 0.4021, which is calculated by privacy analysis. Moreover, we choose the true loss in the visualization of training results, while the noisy loss can not reflect the true convergence situation.

Since DPSGD with DP-similarity is too lengthy, it is abbreviated as "improved DPSGD" below, which is the same in the experiment figure in Barlow Twins.

The experiment results of these two private training methods are presented in Fig.1 and Fig.2. For original DPSGD, the variation curve of training loss looks a straight line, we argued this is because the gradient with momentum changes a little, which can be confirmed by Fig.3. The test accuracy changes randomly, we argued this is because the encoder can learn something but it's hard to get further improvement, hence it's reasonable to argue that original DPSGD can't effectively conduct differentially private training of SimCLR. For DPSGD with DP-similarity, the training loss steadily decreases and the test accuracy gradually increases. The curve of test accuracy started from a very low base and the loss curve started from a very high base are probably because of the interference of the noisy similarity matrix. The comparison of

the two methods shows that our method can effectively conduct differentially private training, and performs better than the original DPSGD.

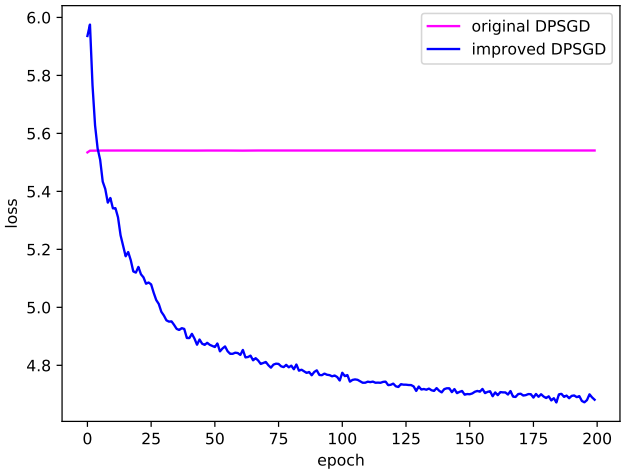

Fig. 1: The training loss of SimCLR

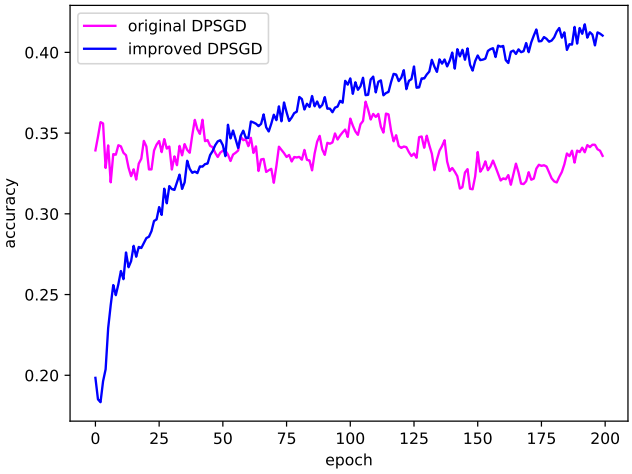

Fig. 2: The test accuracy of SimCLR

As shown in Fig.3, the mean gradient norm in improved DPSGD does not change so much, which manifests a fixed gradient clipping threshold that may be enough to handle the training. Therefore, it's not necessary to adaptively adjust the clipping threshold. However, it doesn't mean that a gradient clipping threshold can apply to all training situations. Different scales of noise

should equip with different clipping thresholds. The mean gradient norm in the original DPSGD changes slightly. It is considered that the noise scale of the gradient is too large and the model does not know how to optimize, so a gradient with small change is obtained.

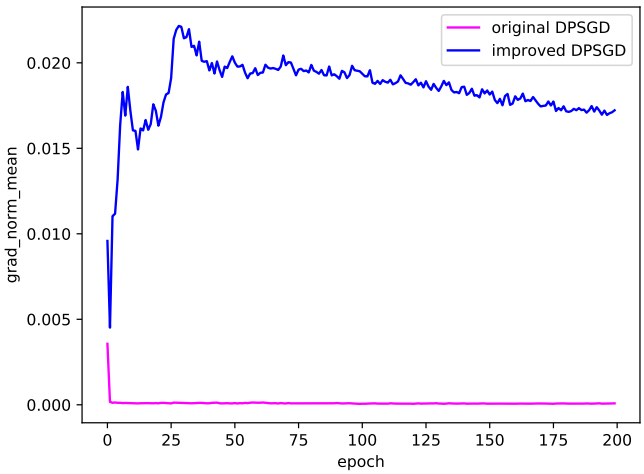

Fig. 3: The mean gradient norm of SimCLR

5.3 Barlow Twins

Architecture Setting. We use a ResNet-18 as a base encoder to encode the image into a 512-dimensional representation. Followed by a projector network, it has three linear layers, each with 2048 output units. In [6], the first two layers of the projector are followed by a batch normalization layer and rectified linear units. We use weight standardization in [47] to replace the batch normalization, while the differential privacy model is not incompatible with the batch normalization layer.

Implementation details. The learning rate is 0.1, the weight decay = 5e-4, the sampling rate $q = 0.00256$, and the number of epochs E is 100.

5.3.1 Original DPSGD

According to the analysis in section 3, in this situation, the sensitivity of gradients is $(4 \cdot |B| \cdot C)$, then the minimum variance of noise to satisfy the privacy budget is $\sigma = 0.3633$, clipping threshold $C = 50$.

5.3.2 DPSGD with DP-mean and DP-variance

With the noisy mean and noisy variance, the gradient's sensitivity has been significantly decreased to $(4 \cdot C)$. The privacy budget is equally allocated to the

gradient, the mean, and the variance. But each batch of augmented representation has a respective mean and variance, so the privacy budget for the mean and the variance need to be further divided equally into two parts respectively. In this setting, the noise scale of the gradient is 0.3931, and the noise scale of all the mean and the variance is set to be the same value 0.416. Another note is that the existence of noisy mean and variance make the gradient's norm smaller, so we set the clipping threshold $C = 1$. Moreover, we still use true loss, which is calculated by true mean and variance to reflect true model performance in the training process.

The experiment results of the above methods are presented in Fig.4 and Fig.5. The performance is a little different from SimCLR. For original DPSGD, the training loss and the test accuracy have similar performance, and they rarely change with the training process. Hence it's hard to effectively conduct differentially private training of Barlow Twins. For DPSGD with DP-mean and DP-variance, the training loss decreases along with training, but not steadily. We believe that in different private training, it sometimes produces very noisy mean and variance, and then the model may update in the wrong direction. Hence, It generates a big loss and a bad parameters solution, i.e.,, it needs more time for private training to find the optimal parameters solution.

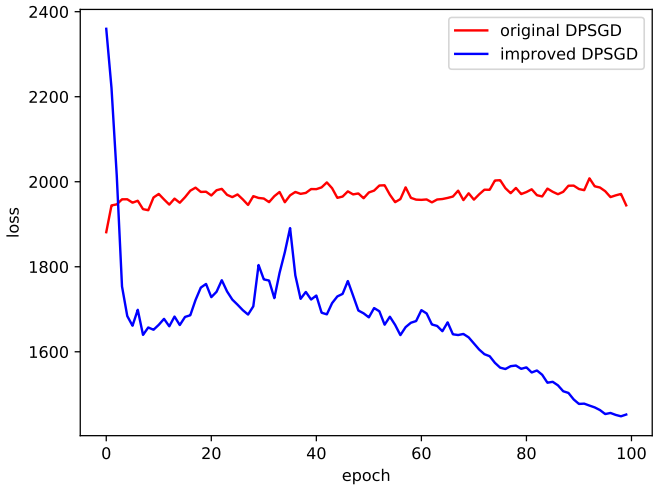

Fig. 4: The training loss of Barlow Twins

6 conclusion

In this paper, we analyze the privacy risk of contrastive learning. Compared to the supervised model (i.e., it has independent per-example gradients), the contrastive model normally has correlated per-example gradients, which brings the challenge of the high sensitivity of the gradient. To solve this issue, we

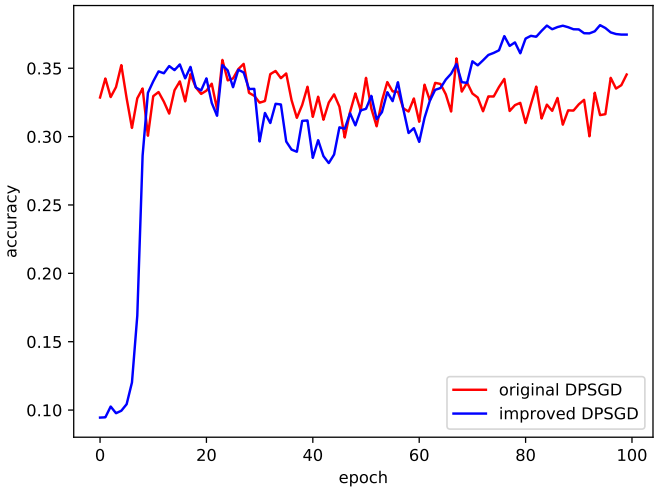

Fig. 5: The test accuracy of Barlow Twins

propose to add differentially private protection to the connection point related to different per-example gradients, which is able to decrease the sensitivity of the gradients significantly.

Specifically, we show our method on SimCLR and analyze the sensitivity of their connection points. More private points mean more privacy budgets, i.e., the noise scale for the gradient is bigger. To demonstrate the superiority of our method, we conduct a comprehensive and detailed evaluation on SimCLR. The results have verified that our improved private training methods can complete effective training while the original DPSGD can't. Although the training loss in our improved methods converged a little slowly, which is a common phenomenon in private training. Therefore, to achieve the same performance as no privacy protection, the version with privacy protection costs more time.

Moreover, this paper also completely analyzes the privacy loss of private training. We clearly show how to leverage the privacy amplification theorem, composition theorem, and conversion theorem to calculate the overall privacy loss of the training process. Through a series of steps, we can obtain the differential privacy of any private training.

This paper can be seen as a preliminary attempt in contrastive learning with differential privacy, our method is innovative while the performance of the contrastive model is not satisfactory enough. So a tremendous effort needs to be done for effectively training the contrastive model with differential privacy. Some avenues for further work are attractive, such as the allocation strategy of privacy budget or finding the compatibility difference between different contrastive models' private training.

Declarations

Ethical Approval and Consent to participate

Not applicable

Human and Animal Ethics

Not applicable

Consent for publication

Not applicable

Availability of supporting data

The CIFAR10 data that support the findings of this study are available in https://www.cs.toronto.edu/ kriz/cifar.html .

Competing interests

The authors declare no competing financial interests.

Funding

Not applicable

Authors' contributions

Conceptualization: Wenjun Li; Investigation: Wenjun Li, Anli Yan; Formal analysis: Di Wu; Methodology and Validation: Taoyu Zhu; Writing – original draft: Wenjun Li; Writing – review editing: Anli Yan, Xuandi Luo; Resources and Funding acquisition: Teng Huang; Supervision: Shaowei Wang. All authors reviewed the manuscript.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No.62102107, 62072132, 62002074, 62072127, 62002076), the Joint Funds of the National Natural Science Foundation of China (No. U20A20176) and National Natural Science Foundation of China for Joint Fund Project (No. U1936218).

Authors' information

Affiliations
Institute of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou, China
Wenjun Li, Di Wu, Taoyu Zhu, Teng Huang, Xuandi Luo, Shaowei Wang

School of Cyberspace Security (School of Cryptology), Hainan University, Haikou, China
Anli Yan
Corresponding authors
Correspondence to Teng Huang.

References

[1] Li, W., Ke, L., Meng, W., Han, J.: An empirical study of supervised email classification in internet of things: Practical performance and key influencing factors. International Journal of Intelligent Systems **37**(1), 287–304 (2022)

[2] Cai, J., Fu, H., Liu, Y.: Deep reinforcement learning-based multitask hybrid computing offloading for multiaccess edge computing. International Journal of Intelligent Systems (2022)

[3] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607 (2020). PMLR

[4] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738 (2020)

[5] Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., et al.: Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:2006.07733 (2020)

[6] Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. arXiv preprint arXiv:2103.03230 (2021)

[7] Liu, H., Jia, J., Qu, W., Gong, N.Z.: Encodermi: Membership inference against pre-trained encoders in contrastive learning. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, pp. 2081–2095 (2021)

[8] He, X., Zhang, Y.: Quantifying and mitigating privacy risks of contrastive learning. arXiv preprint arXiv:2102.04140 (2021)

[9] Dwork, C.: Differential privacy. In: International Colloquium on Automata, Languages, and Programming, pp. 1–12 (2006). Springer

[10] Bassily, R., Smith, A., Thakurta, A.: Private empirical risk minimization:

Efficient algorithms and tight error bounds. In: 2014 IEEE 55th Annual Symposium on Foundations of Computer Science, pp. 464–473 (2014). IEEE

[11] Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1310–1321 (2015)

[12] Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 308–318 (2016)

[13] Yu, L., Liu, L., Pu, C., Gursoy, M.E., Truex, S.: Differentially private model publishing for deep learning. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 332–349 (2019). IEEE

[14] Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2, pp. 1735–1742 (2006). IEEE

[15] Gao, G., Shao, H., Wu, F., Yang, M., Yu, Y.: Leaning compact and representative features for cross-modality person re-identification. World Wide Web, 1–18 (2022)

[16] Wang, Y., Dai, Z., Cao, J., Wu, J., Tao, H., Zhu, G.: Intra-and inter-association attention network-enhanced policy learning for social group recommendation. World Wide Web, 1–24 (2022)

[17] Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)

[18] Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)

[19] Cai, Q., Wang, Y., Pan, Y., Yao, T., Mei, T.: Joint contrastive learning with infinite possibilities. arXiv preprint arXiv:2009.14776 (2020)

[20] Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. arXiv preprint arXiv:2104.14548 (2021)

[21] Ermolov, A., Siarohin, A., Sangineto, E., Sebe, N.: Whitening for self-supervised representation learning. In: International Conference on Machine Learning, pp. 3015–3024 (2021). PMLR

[22] Yan, H., Hu, L., Xiang, X., Liu, Z., Yuan, X.: Ppcl: Privacy-preserving collaborative learning for mitigating indirect information leakage. Information Sciences **548**, 423–437 (2021)

[23] Yan, H., Chen, M., Hu, L., Jia, C.: Secure video retrieval using image query on an untrusted cloud. Applied Soft Computing **97**, 106782 (2020)

[24] Mo, K., Huang, T., Xiang, X.: Querying little is enough: Model inversion attack via latent information. In: International Conference on Machine Learning for Cyber Security, pp. 583–591 (2020). Springer

[25] Wang, X., Li, J., Kuang, X., Tan, Y.-a., Li, J.: The security of machine learning in an adversarial setting: A survey. Journal of Parallel and Distributed Computing **130**, 12–23 (2019)

[26] Kong, L., Wang, L., Gong, W., Yan, C., Duan, Y., Qi, L.: Lsh-aware multitype health data prediction with privacy preservation in edge environment. World Wide Web, 1–16 (2021)

[27] Ni, H., Wang, S., Cheng, P.: A hybrid approach for stock trend prediction based on tweets embedding and historical prices. World Wide Web **24**(3), 849–868 (2021)

[28] Wang, H., Xu, Z., Jia, S., Xia, Y., Zhang, X.: Why current differential privacy schemes are inapplicable for correlated data publishing? World Wide Web **24**(1), 1–23 (2021)

[29] Wu, Z., Li, G., Shen, S., Lian, X., Chen, E., Xu, G.: Constructing dummy query sequences to protect location privacy and query privacy in location-based services. World Wide Web **24**(1), 25–49 (2021)

[30] Dwork, C., Roth, A., *et al.*: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3-4), 211–407 (2014)

[31] Wang, Y.-X., Balle, B., Kasiviswanathan, S.P.: Subsampled rényi differential privacy and analytical moments accountant. In: The 22nd International Conference on Artificial Intelligence and Statistics, pp. 1226–1235 (2019). PMLR

[32] Mironov, I.: Rényi differential privacy. In: 2017 IEEE 30th Computer Security Foundations Symposium (CSF), pp. 263–275 (2017). IEEE

[33] Dwork, C., Rothblum, G.N.: Concentrated differential privacy. arXiv preprint arXiv:1603.01887 (2016)

[34] Bun, M., Steinke, T.: Concentrated differential privacy: Simplifications, extensions, and lower bounds. In: Theory of Cryptography Conference,

pp. 635–658 (2016). Springer

[35] Pichapati, V., Suresh, A.T., Yu, F.X., Reddi, S.J., Kumar, S.: Adaclip: Adaptive clipping for private sgd. arXiv preprint arXiv:1908.07643 (2019)

[36] Zhu, Y., Wang, Y.-X.: Poission subsampled rényi differential privacy. In: International Conference on Machine Learning, pp. 7634–7642 (2019). PMLR

[37] Balle, B., Barthe, G., Gaboardi, M.: Privacy amplification by subsampling: Tight analyses via couplings and divergences. arXiv preprint arXiv:1807.01647 (2018)

[38] Goodfellow, I.: Efficient per-example gradient computations. arXiv preprint arXiv:1510.01799 (2015)

[39] Lee, J., Kifer, D.: Scaling up differentially private deep learning with fast per-example gradient clipping. Proc. Priv. Enhancing Technol. **2021**(1), 128–144 (2021)

[40] Bu, Z., Dong, J., Long, Q., Su, W.J.: Deep learning with gaussian differential privacy. Harvard data science review **2020**(23) (2020)

[41] Campbell, Z., Bray, A., Ritz, A., Groce, A.: Differentially private anova testing. In: 2018 1st International Conference on Data Intelligence and Security (ICDIS), pp. 281–285 (2018). IEEE

[42] Mironov, I., Talwar, K., Zhang, L.: Rényi differential privacy of the sampled gaussian mechanism. arXiv preprint arXiv:1908.10530 (2019)

[43] Balle, B., Barthe, G., Gaboardi, M., Hsu, J., Sato, T.: Hypothesis testing interpretations and renyi differential privacy. In: International Conference on Artificial Intelligence and Statistics, pp. 2496–2506 (2020). PMLR

[44] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)

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

[46] Susmelj, I., Heller, M., Wirth, P., Prescott, J., Ebner, M.e.a.: Lightly. arXiv preprint arXiv:2104.14548 (2020)

[47] Qiao, S., Wang, H., Liu, C., Shen, W., Yuille, A.: Micro-batch training with batch-channel normalization and weight standardization. arXiv preprint arXiv:1903.10520 (2019)