Continual Learning with Differential Privacy

Pradnya Desai[1], Phung Lai[1], NhatHai Phan[1(✉)], and My T. Thai[2]

[1] New Jersey Institute of Technology, Newark, USA
{pnd26,tl353,phan}@njit.edu
[2] University of Florida, Gainesville, USA
mythai@cise.ufl.edu

Abstract. In this paper, we focus on preserving differential privacy (DP) in continual learning (CL), in which we train ML models to learn a sequence of new tasks while memorizing previous tasks. We first introduce a notion of continual adjacent databases to bound the sensitivity of any data record participating in the training process of CL. Based upon that, we develop a new DP-preserving algorithm for CL with a data sampling strategy to quantify the privacy risk of training data in the well-known Averaged Gradient Episodic Memory (A-GEM) approach by applying a moments accountant. Our algorithm provides formal guarantees of privacy for data records across tasks in CL. Preliminary theoretical analysis and evaluations show that our mechanism tightens the privacy loss while maintaining a promising model utility.

Keywords: Continual learning · Differential privacy · Deep learning

1 Introduction

The ability to acquire new knowledge over time while retaining previously learned experiences, referred to as continual learning (CL), brings machine learning (ML) closer to human learning [17]. More specifically, given a stream of tasks, CL focuses on training a ML model to quickly learn a new task by leveraging the acquired knowledge after learning previous tasks under a limited amount of computation and memory resources [10]. As a result, the main challenge of existing CL algorithms is that they can be quickly suffered by catastrophic forgetting.

Also, memorizing previous tasks while learning new tasks further exposes CL models to adversarial attacks [7,18]. CL models can disclose private information in the training set, such as healthcare and financial data [9]. Continuously accessing the data from the previously learned tasks, either stored in episodic memories [3] or produced from generative memories [11], incurs additional privacy risk compared to a ML model trained on a single task. However, there is still a lack of scientific study to protect private training data in CL algorithms.

Motivated by this, we propose to preserve differential privacy (DP) [4], offering rigorous privacy protection as probabilistic terms for the training data in CL.

P. Desai and P. Lai—These two authors contributed equally.

T. Mantoro et al. (Eds.): ICONIP 2021, CCIS 1517, pp. 334–343, 2021.
https://doi.org/10.1007/978-3-030-92310-5_39

Merely employing existing DP-preserving mechanisms can either cause a significantly large privacy loss or quickly exhaust the limited computation and memory resources in learning new tasks while memorizing previous tasks through either episodic or generative memories. Thus, effectively and efficiently preserving DP in CL remains a mostly open problem.

Key Contributions. To effectively bound the DP privacy loss in CL, we first define continual adjacent databases (Definition 2) to capture the impact of the current task's data and the episodic memory on the privacy loss and model utility. From that, we incorporate a moments accountant [1] into the A-GEM algorithm [3] in a new **DP-CL** algorithm to preserve DP in CL.

Our idea is to configure the episodic memory $\mathcal{M}$ in A-GEM as independent mini-memory blocks. We store a subset of training data of the current task in a mini-memory block with an associated task index in $\mathcal{M}$ for each task. At each training step, we compute reference gradients on the mini-memory blocks independently. The reference gradients will be used to optimize the process of memorizing previously learned tasks as in A-GEM. Importantly, by keep tracking of the task and mini-memory block index, we can leverage a moments accountant to estimate the privacy cost spent on each mini-memory block. Based upon this, we derive a new strategy (Lemma 2) to bound DP loss in the whole CL process while maintaining the computation efficiency of the A-GEM algorithm.

To our knowledge, our proposed mechanism establishes the first formal connection between DP and CL. Experiments conducted on the permuted MNIST dataset [8] and the Split CIFAR [19] show promising results in preserving DP in CL, compared with baseline approaches.

2 Background

In this section, we revisit continual learning, differential privacy, and introduce our problem statement. The goal of CL is to learn a model through a sequence of tasks $T = [t_i]_{i \in [1,N]}$ such that the learning of each new task will not cause forgetting of the previously learned tasks. Let $\mathcal{D}_\mathcal{T}$ be the dataset at task $\mathcal{T}$ consisting of $S_\mathcal{T}$ samples, each of which is a sample $x \in \mathbb{R}^d$ associated with a label y. Each y is a one-hot vector of C categories: $y = [y_c]_{c \in [1,C]}$. A classifier outputs class scores $f : \mathbb{R}^d \rightarrow \mathbb{R}^C$ mapping an input x to a vector of scores $f(x) = [f_c(x)]_{c \in [1,C]}$ s.t. $\forall c \in [1, C] : f_c(x) \in [0, 1]$ and $\sum_{c=1}^{C} f_c(x) = 1$. The class with the highest score is selected as the predicted label for the sample. The classifier f is trained by minimizing a loss function $\mathcal{L}(f(x), y)$ that penalizes mismatching between the prediction $f(x)$ and the original value y.

Averaged Gradient Episodic Memory (A-GEM) [3]**.** There is a sequence of tasks $[t_i]_{i \in [1,\mathcal{T}-1]}$ that have been learnt, where $\mathcal{T} < N$. The goal is to train the model at the current task $\mathcal{T}$ so that it minimizes the loss on the task $\mathcal{T}$ and does not forget previous learned tasks $i < \mathcal{T}$. The key feature of A-GEM is to store a subset of data from task i, denoted as $\mathcal{M}_i$, in an episodic memory $\mathcal{M}$. Then the algorithm ensures that the loss on an average episodic memory across all the previously learned tasks, i.e., $\mathcal{M} = \cup_{i<\mathcal{T}} \mathcal{M}_i$, does not increase at every step. In A-GEM, the objective function of learning the current task $\mathcal{T}$ is:

$$\theta^{\mathcal{T}} = \min_{\theta} \mathcal{L}\big(f(\theta^{\mathcal{T}-1}, \mathcal{D}_{\mathcal{T}})\big) \text{ s.t. } \mathcal{L}\big(f(\theta^{\mathcal{T}}, \mathcal{M})\big) \leq \mathcal{L}\big(f(\theta^{\mathcal{T}-1}, \mathcal{M})\big) \tag{1}$$

where $\theta^{\mathcal{T}-1}$ is the values of model parameters θ learned after training the task $\mathcal{T} - 1$, and $\mathcal{L}\big(f(\theta^{\mathcal{T}-1}, \mathcal{D}_{\mathcal{T}})\big) = \frac{1}{|\mathcal{D}_{\mathcal{T}}|} \sum_{x \in \mathcal{D}_{\mathcal{T}}} \mathcal{L}\big(f(\theta^{\mathcal{T}-1}, x)\big)$.

The constrained optimization problem of Eq. 1 can be approximated quickly and the updated gradient $\tilde{g}$ is as follows:

$$\tilde{g} = g - \frac{g^T g_{ref}}{g_{ref}^T g_{ref}} g_{ref} \tag{2}$$

where g is the proposed gradient update on $\mathcal{T}$ and g_{ref} is the reference gradient computed from the episodic memory $\mathcal{M}$ from previous tasks.

Differential Privacy [4,5]**.** To avoid the training data leakage, DP guarantees to restrict what the adversaries can learn from the training data given the model parameters by ensuring similar model outcomes with and without any single data sample in the dataset. The definition of DP is as follows:

Definition 1 (ϵ, δ)*-DP [4]. A randomized algorithm A fulfills (ϵ, δ)-DP, if for any two adjacent databases D and D' differ at most one sample, and for all outcomes $\mathcal{O} \subseteq Range(A)$, we have: $Pr[A(D) = \mathcal{O}] \leq e^{\epsilon} Pr[A(D') = \mathcal{O}] + \delta$, where ϵ is the privacy budget and δ is the broken probability.*

DP in Continual Learning. There are several works of DP in CL [6,13]. In [6], the authors train a DP-GAN to approximate the distribution of the past datasets. They leverage a small portion of public data (i.e., the data that does not need to keep private) to initialize and train the GAN in the first few iterations of each task, then continue training the GAN model under DP constraint. The trained generator produces adversarial examples imitating real examples of past tasks. Then, the adversarial examples are employed to supplement the actual data of the current training task. DPL2M [13] perturbs the objective functions using a DPAL mechanism [12,14] and applies A-GEM to optimize the perturbed objective function. However, there is a lack of a concrete definition of adjacent databases with unclear or not well-justified DP protection in [6,13]. Different from existing works, we provide a formal DP protection for CL models.

3 Continual Learning with DP

This section establishes a connection between differential privacy and continual learning. We first propose a definition of continual adjacent databases in CL, as follows: Two databases D and D' are continual adjacent if they differ in a single sample of the training data and differ in a single sample of the episodic memory across all the tasks. The definition is presented as follows:

Definition 2 *Continual Adjacent Databases. Two databases $D = (\mathcal{D}, \mathcal{M})$ and $D' = (\mathcal{D}', \mathcal{M}')$, where $\mathcal{D} = \cup_{i=1}^{N} \mathcal{D}_i$, $\mathcal{D}' = \cup_{i=1}^{N} \mathcal{D}'_i$, $\mathcal{M} = \cup_{i=1}^{N} \mathcal{M}_i$, and $\mathcal{M}' = \cup_{i=1}^{N} \mathcal{M}'_i$, are called continual adjacent if: $\|\mathcal{D} - \mathcal{D}'\|_1 \leq 1$ and $\|\mathcal{M} - \mathcal{M}'\|_1 \leq 1$.*

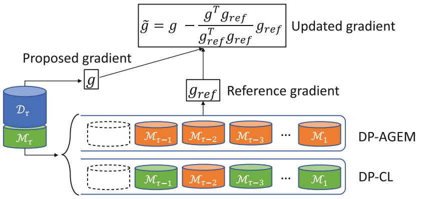


Fig. 1. DP in CL protects privacy for a stream of different tasks. Here, blue box indicates training data of task $\mathcal{T}$, orange and green boxes indicate mini-memory blocks in $\mathcal{M}$, and the orange ones are for computing g_{ref}. (Color figure online)

A Naive Algorithm. Based upon Definition 2, a straightforward approach, called DP-AGEM, is to simply apply a moments accountant [1] into A-GEM [3], to preserve DP in CL. At each task $\mathcal{T}$, we divide the dataset $D_{\mathcal{T}}$ into $D_{\mathcal{T}}^{train}$ and $D_{\mathcal{T}}^{ref}$ such that $D_{\mathcal{T}}^{train}$ and $D_{\mathcal{T}}^{ref}$ are disjoint: $D_{\mathcal{T}}^{train} \cap D_{\mathcal{T}}^{ref} = \emptyset$. By using the training data $\mathcal{D}_{\mathcal{T}}^{train}$ with a sampling rate p, DP-AGEM computes a proposed gradient g, which is bounded by a predefined l_2-norm clipping bound β. It is beneficial in real-world to keep track of the privacy budget spent on each task independently, and the total privacy budget used in the entire training process. To achieve this, in computing the reference gradients g_{ref}, the algorithm first randomly samples data from all the data samples in the episodic memory $\mathcal{M}$ with a sampling probability q. Given a particular D_i^{ref} ($i \in [1, \mathcal{T} - 1]$) in the episodic memory, the sampled data is used to compute a reference gradient g_{ref}^i, which is clipped with the l_2-norm bound β. Then Gaussian mechanism is employed to inject random Gaussian noise $\mathcal{N}(0, \sigma^2 \beta^2 I)$ with a predefined hyper-parameter σ into both g and g_{ref}^i. The reference gradient g_{ref} is the average of all the reference gradients computed on each D_i^{ref}, as follows: $g_{ref} = \frac{1}{\mathcal{T}-1} \sum_{i \in [1, \mathcal{T}-1]} g_{ref}^i$. Finally, the updated gradient $\tilde{g}$ computed using Eq. 2 with g_{ref} and g can be used to update the model parameters. After training the task $\mathcal{T}$, $D_{\mathcal{T}}^{ref}$ is added into the episodic memory $\mathcal{M}$. The training process will continue until the model is trained on all the tasks.

Since the l_2-norms of g and g_{ref}^i are bounded, we can leverage a moments accountant to bound the privacy loss for a single task $\mathcal{T}$ and for accumulation across all tasks. Let $\epsilon_{\mathcal{T}}$ be the privacy budget used to compute g on $\mathcal{D}_{\mathcal{T}}^{train}$, and ϵ'_i is the privacy budget spent on computing the reference gradient g_{ref}^i at each training task. The privacy budget used for a specific task $i \in [1, \mathcal{T})$, denoted as $\epsilon_i(\mathcal{T})$ and the total privacy budgets $\epsilon_{\mathcal{T}}^{all}$ of DP-AGEM accumulated until the task $\mathcal{T}$ can be computed in the following lemma.

Lemma 1. *Until the task $\mathcal{T}$, 1) the privacy budget used for a specific and previously learned task $i \in [1, \mathcal{T}]$ is: $\epsilon_i(\mathcal{T}) = \epsilon_i + (\mathcal{T} - i)\epsilon'_i$, and 2) the total privacy budget $\epsilon_{\mathcal{T}}^{all}$ of DP-AGEM is: $\epsilon_{\mathcal{T}}^{all} = \sum_{i=1}^{\mathcal{T}} \epsilon_i(\mathcal{T})$.*

Proof. We use induction to prove Lemma 1. When $\mathcal{T} = 1$, $\mathcal{M}$ is empty; therefore, $\epsilon_1^{all} = \epsilon_1 = \epsilon_1(1)$. Hence, Lemma 1 is true for $\mathcal{T} = 1$. Assuming that it is true for $\mathcal{T} = k$, so $\epsilon_i(k) = \epsilon_i + (k-i)\epsilon'_i$ and $\epsilon_k^{all} = \sum_{i=1}^{k} \epsilon_i(k)$. We need to show that Lemma 1 is true for $\mathcal{T} = k+1$. We have: $\epsilon_i(k+1) = \epsilon_i(k) + \epsilon'_i = \epsilon_i + (k+1-i)\epsilon'_i$, and $\epsilon_{k+1}^{all} = \sum_{i=1}^{k} \epsilon_i(k) + \epsilon_{k+1} + \sum_{i=1}^{k} \epsilon'_i = \sum_{i=1}^{k+1} \epsilon_i(k+1)$. Thus, Lemma 1 holds.

Two Levels of DP Protection. In Lemma 1, based on our definition of continual adjacent databases (Definition 2), it is essential that there are two levels of DP protection provided to an arbitrary data sample, as follows. Until the task $\mathcal{T} \in [1, N]$: **(1)** Given the DP budget $\epsilon_i(\mathcal{T})$ for a specific task $i \in [1, \mathcal{T}]$, the participation information of an arbitrary data sample in the task i is protected under a $(\epsilon_i(\mathcal{T}), \delta)$-DP given the released parameters θ. This can be presented as: $Pr[\text{DP-AGEM}(\mathcal{D}_i) = \theta] \leq e^{\epsilon_i(\mathcal{T})} Pr[\text{DP-AGEM}(\mathcal{D}'_i) = \theta] + \delta$, for any adjacent databases $\mathcal{D}_i$ and $\mathcal{D}'_i$; and **(2)** The participation information of an arbitrary data sample in the whole training data ($\mathcal{D} = \cup_{i=1}^{\mathcal{T}} \mathcal{D}_i^{train}, \mathcal{M} = \cup_{i=1}^{\mathcal{T}} \mathcal{D}_i^{ref}$) is protected under a $(\epsilon_{\mathcal{T}}^{all}, \delta)$-DP given the released parameters θ. This can be presented as: $Pr[\text{DP-AGEM}(\mathcal{D}, \mathcal{M}) = \theta] \leq e^{\epsilon_{\mathcal{T}}^{all}} Pr[\text{DP-AGEM}(\mathcal{D}', \mathcal{M}') = \theta] + \delta$, for any continual adjacent databases $(\mathcal{D}, \mathcal{M})$ and $(\mathcal{D}', \mathcal{M}')$. This is fundamentally different from existing works [6,13], which do not provide any formal DP in CL.

Although DP-AGEM can preserve DP in CL, it suffers from a large privacy budget accumulation across tasks with an $O(\mathcal{T}^2)$ for $\epsilon_{\mathcal{T}}^{all}$. To address this impractical issue, we present an algorithm to tighten the DP loss.

DP-CL Algorithm. Our DP-CL (Algorithm 1 and Fig. 1) takes a sequence of tasks $T = [t_i]_{i \in [1,N]}$ and dataset $\mathcal{D} = \cup_{i=1}^{N} \mathcal{D}_i$ as inputs. All samples in $D_{\mathcal{T}}^{train}$ are used to compute the proposed gradient update g on task $\mathcal{T}$ with a sampling rate p (Line 6). We clip g so that its l_2-norm is bounded by a predefined gradient clipping bound β. Then we add a random Gaussian noise $\mathcal{N}(0, \sigma^2\beta^2 I)$ into g with a predefined noise scale σ (Line 9). Note that after training the task $\mathcal{T}$, samples in $D_{\mathcal{T}}^{ref}$ are added to the episodic memory $\mathcal{M}$ as a mini-memory block $\mathcal{M}_{\mathcal{T}}$ (Lines 17, 24–26). To reduce the privacy budget accumulated over the number of tasks, we limit the access to seen data of previous tasks by using a randomly selected mini-memory block $\mathcal{M}_i$ ($i < \mathcal{T}$) from $\mathcal{M}$ to compute g_{ref} (Lines 20–23). We clip g_{ref} by the gradient clipping bound β and then add a random Gaussian noise $\mathcal{N}(0, \sigma^2\beta^2 I)$ to g_{ref} (Line 14). The updated gradient $\tilde{g}$ is computed by Eq. 2 (Line 15). Then $\tilde{g}$ is used to update the model parameters θ (Line 16). The privacy budgets in our DP-CL can be bounded in the following lemma.

Lemma 2. *Until the task $\mathcal{T}$, 1) the privacy budget used for a specific and previously learned task $i \in [1, \mathcal{T}]$ is: $\epsilon_i(\mathcal{T}) = \epsilon_i + \epsilon'_i$, where ϵ'_i is the privacy budget used for a randomly chosen mini-memory block from $\mathcal{M}$ to compute g_{ref} at task i, and 2) the total privacy budget $\epsilon_{\mathcal{T}}^{all}$ of DP-CL is: $\epsilon_{\mathcal{T}}^{all} = \sum_{i=1}^{\mathcal{T}} \epsilon_i(\mathcal{T})$.*

Proof. Similar to the proof of Lemma 1 with using induction. Here, we need to show that it is true for $\mathcal{T} = k+1$. We have: $\epsilon_i(k+1) = \epsilon_i + \epsilon'_i$, and $\epsilon_{k+1}^{all} = \sum_{i=1}^{k} \epsilon_i(k) + \epsilon_{k+1} + \epsilon'_{k+1} = \sum_{i=1}^{k+1} \epsilon_i(k+1)$. Consequently, Lemma 2 hold.

It is obvious that our DP-CL algorithm significantly reduces the privacy consumption to $O(\mathcal{T})$, which is linear to the number of training tasks. In addition, our sampling approach to compute g_{ref} is unbiased, since the expectation for any data sample selected to compute g_{ref} is the same: $\forall x \in \mathcal{M}, \mathbb{E}(x \in \mathcal{M}_i) = q/(\mathcal{T}-1)$. In our experiment, we will show that DP-CL outperforms DP-AGEM.

Algorithm 1. DP in Continual Learning (DP-CL) Algorithm

```
**Input:** Number of tasks $N$, dataset $\mathcal{D} = \cup_{i=1}^{N}\mathcal{D}_i$, gradient clipping bound $\beta$
Initialize model $\theta$, episodic memory $\mathcal{M} = \emptyset$, moments accountant $\mathbb{M}$
**for** $\mathcal{T} = \{1, ..., N\}$ **do**
    $D_{\mathcal{T}}^{train} \sim \mathcal{D}_{\mathcal{T}}, D_{\mathcal{T}}^{ref} \sim \mathcal{D}_{\mathcal{T}}$ s.t. $D_{\mathcal{T}}^{train} \cup D_{\mathcal{T}}^{ref} = \mathcal{D}_{\mathcal{T}}$, $D_{\mathcal{T}}^{train} \cap D_{\mathcal{T}}^{ref} = \emptyset$
    **for** each iteration $e = 0, 1, 2, \ldots$ **do**
        $\mathcal{D}_{\mathcal{T}}^{e} \leftarrow$ Take random samples in $\mathcal{D}_{\mathcal{T}}^{train}$ with a sampling rate $p$
        **for** $(x, y) \in D_{\mathcal{T}}^{e}$ **do**
            $g \leftarrow \text{ClipGrad}(\nabla_\theta \mathcal{L}(f_\theta(x), y), \beta) + \mathcal{N}(0, \sigma^2\beta^2 I)$
            **if** $\mathcal{T} = 1$ **then**
                $\tilde{g} \leftarrow g$
            **else**
                $g_{ref} \leftarrow \text{ClipGrad}(\text{CalGref}(\mathcal{M}, \mathcal{T}), \beta) + \mathcal{N}(0, \sigma^2\beta^2 I)$
                Compute $\tilde{g}$ with Eq. 2
            $\theta \leftarrow \theta - \alpha\tilde{g}$
    $\mathcal{M} \leftarrow$ **UpdateEpsMem**$(\mathcal{M}, D_{\mathcal{T}}^{ref}, \mathcal{T})$
**Output:** $(\epsilon, \delta)$-DP-CL $\theta$, $\mathbb{M}$ (from $\mathbb{M}$.get_priv_spent())
**CalGref**$(\mathcal{M}, \mathcal{T})$:
    Randomly choose $\mathcal{M}_i$ from $\mathcal{M}$, where $i < \mathcal{T}$
    $(x^{ref}, y^{ref}) \sim \mathcal{M}_i$ ($\mathcal{M}_i$ is randomly chosen from $\mathcal{M}$, where $i < \mathcal{T}$ )
    return $g_{ref} = \nabla_\theta \mathcal{L}(f_\theta(x^{ref}), y^{ref})$
**UpdateEpsMem**$(\mathcal{M}, D_{\mathcal{T}}^{ref}, \mathcal{T})$:
    $\mathcal{M}_{\mathcal{T}} \leftarrow D_{\mathcal{T}}^{ref}$
    return $\mathcal{M} \cup \mathcal{M}_{\mathcal{T}}$
**ClipGrad**$(g, \beta)$: return $\pi(g, \beta) = g \cdot \min\left(1, \frac{\beta}{\|g\|}\right)$
```

4 Experimental Results

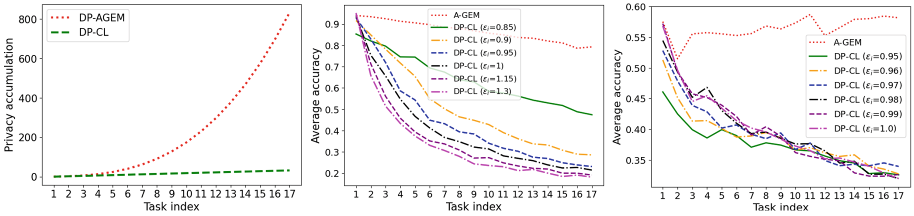
We have conducted experiments on the permuted MNIST dataset [8] and the Split CIFAR dataset [19]. Our validation focuses on shedding light on the interplay between model utility and privacy loss of preserving DP in CL. Our code, datasets, and model configurations are available on Github[1].

Baseline Approaches. We evaluate our DP-CL algorithm and compare it with A-GEM [3], one of the state-of-the-art CL algorithms. Since A-GEM does not preserve DP, we only use A-GEM to show the upper-bound performance. We use the *average accuracy*, the *average forgetting* (F), the *worst-case forgetting* (worst-case F), and the *learning curve area* (LCA) [3] for evaluation.

[1] https://github.com/PhungLai728/DP-CL.

- **Comparing Privacy Accumulation.** Since the number of data samples and the sampling rate remain the same for every task, ϵ_i and ϵ'_i can be the same for every task. Therefore, for the sake of clarity without loss of generality, we draw different random Gaussian values (mean = 1, std = 0.02) and assign the generated values as the privacy budget ϵ_i and ϵ'_i for 17 tasks.

Figure 2a illustrates how privacy loss accumulates over 17 tasks in DP-AGEM and our DP-CL. Our algorithm achieves a notably tighter privacy budget compared with DP-AGEM, which accesses data samples from the whole episodic memory to compute g_{ref}. When the number of tasks increases, DP-AGEM's privacy budget exponentially increases. In contrast, our approach's privacy budget slightly increases and is linear to the number of tasks or training steps.



(a) Privacy Accumulation (b) Permuted MNIST (c) Split CIFAR

Fig. 2. Theoretical analysis for privacy accumulation (a); and Average accuracy over 17 tasks of A-GEM and DP-CL algorithms with varying ϵ_i.

- **Privacy Loss and Model Utility.** From our theoretical analysis, DP-AGEM suffers from a huge privacy budget accumulation over tasks. Therefore, we only compare our DP-CL and A-GEM for the sake of simplicity.

As shown in Fig. 2b and 2c, our proposed method achieves a comparable average accuracy with the noiseless A-GEM model at the first task. In the permuted MNIST dataset, when the number of tasks increases, the average accuracy of our DP-CL drops faster than the average accuracy of the A-GEM model. For example, at task 17-th, A-GEM's average accuracy drops to 79.3%, while DP-CL's average accuracy drops to 47.5% with a tight privacy budget $\epsilon_i = 0.85$. When the privacy budget increases, the average accuracy gap between our model and the noiseless A-GEM is larger, indicating that preserving DP in CL may increase the catastrophic forgetting. This phenomenon is further clarified by F, worst-case F, and LCA (Table 1). At $\epsilon_i = 0.85$, the values of F, worst-case F, and LCA are 0.401, 0.586, and 0.146 respectively in DP-CL. After that, the F and worst-case F significantly increase, and LCA moderately decreases in DP-CL.

In the Split CIFAR dataset, when the number of tasks increases, the average accuracy of DP-CL drops quickly while the average accuracy of the A-GEM model fluctuates. For instances, A-GEM's average accuracy is 57.5% at the first task, drops to 51.5% at the second task, and is 58.1% at the last task. Meanwhile, DP-CL's average accuracy is 56.8% at the first task, and gradually drops to

Table 1. Forgetting measure (F), worst-case F, and LCA results for the MNIST dataset. The lower F, worst-case F, and the higher LCA the better.

			Forgetting (F)	Worst-case F	LCA
MNIST	**A-GEM**		0.166 ± 0.0070	0.272 ± 0.0086	0.481 ± 0.0051
	DP-CL ($\epsilon_i' = 1.47$ and $\delta = 10^{-4}$ for all tasks)	$\epsilon_i = 0.85$	0.401 ± 0.0070	0.586 ± 0.0191	0.146 ± 0.0077
		$\epsilon_i = 0.9$	0.657 ± 0.0099	0.809 ± 0.0110	0.123 ± 0.0039
		$\epsilon_i = 0.95$	0.713 ± 0.0060	0.840 ± 0.0186	0.120 ± 0.0038
		$\epsilon_i = 1.0$	0.750 ± 0.0017	0.851 ± 0.0081	0.119 ± 0.0115
		$\epsilon_i = 1.15$	0.782 ± 0.0017	0.863 ± 0.0061	0.124 ± 0.0013
		$\epsilon_i = 1.30$	0.796 ± 0.0023	0.864 ± 0.0077	0.121 ± 0.0021
CIFAR	**A-GEM**		0.089 ± 0.0163	0.188 ± 0.0317	0.348 ± 0.0111
	DP-CL ($\epsilon_i' = 1.47$ and $\delta = 10^{-4}$ for all tasks)	$\epsilon_i = 0.95$	0.149 ± 0.0123	0.314 ± 0.0057	0.262 ± 0.0058
		$\epsilon_i = 0.96$	0.181 ± 0.0193	0.335 ± 0.0421	0.259 ± 0.0130
		$\epsilon_i = 0.97$	0.196 ± 0.0194	0.377 ± 0.0174	0.266 ± 0.0111
		$\epsilon_i = 0.98$	0.239 ± 0.0162	0.428 ± 0.0701	0.266 ± 0.0008
		$\epsilon_i = 0.99$	0.249 ± 0.0097	0.435 ± 0.0432	0.259 ± 0.0053
		$\epsilon_i = 1.0$	0.262 ± 0.031	0.455 ± 0.0452	0.263 ± 0.0096

31.9% at the last task with a tight privacy budget $\epsilon_i = 1.0$. The fluctuation phenomenon in the A-GEM model is probably due to the curse of dimension in which there are $2,500$ training examples, which is much smaller than the number of trainable parameters in the ResNet-18, i.e., 11 million. Different from the permuted MNIST dataset, in the Split CIFAR dataset, when the privacy budget increases, the average accuracy gap between DP-CL and the noiseless A-GEM is smaller, especially at the first task. For instance, at the first task, the gaps are 11.4%, 6.3%, 4.7%, 3.1%, 0.3%, and 0.7% when the values of $\epsilon_i \in [0.95, 1.0]$. This shows the trade-off between privacy budget and model utility in which when we spend more privacy budget, the model accuracy improves. The gap between DP-CL's and A-GEM's average accuracy are significantly bigger when the number of tasks increases, but the difference among different privacy budgets decreases. For instance, at the last task, the gaps are $[24.2\%, 26.2\%]$ when $\epsilon_i \in [0.95, 1.0]$. As shown in Table 1, when the privacy budget increases, the F and worst-case F significantly increase, while the LCA slightly fluctuates around $[0.259, 0.266]$.

Key Observations. From our experiments, we obtain the following observations. **(1)** Merely incorporating the moments accountant into A-GEM causes a large privacy budget accumulation. **(2)** Although our DP-CL algorithm preserves DP in CL, optimizing the trade-off between model utility and privacy loss is an open problem as the privacy noise can worsen the catastrophic forgetting.

5 Conclusion and Future Work

In this paper, we established the first formal connection between DP and CL. We combine the moments accountant and A-GEM in a holistic approach to preserve DP in CL in a tightly accumulated privacy budget. Our model shows promising results under strong DP guarantees in CL and opens a new research line to optimize the model utility and privacy loss trade-off. One of the immediate questions is how to align the privacy noise with the catastrophic forgetting under the same privacy protection. We will examine our approach to a broader range of models and datasets, especially under attacks [2,18], and heterogeneous privacy-preserving mechanisms [15,16]. Our work further highlights an open direction of quantifying the privacy risk given a diverse correlation among tasks.

Acknowledgment. The authors gratefully acknowledge the support from the National Science Foundation grants NSF CNS-1935928/1935923, CNS-1850094, IIS-2041096/2041065.

References

1. Abadi, M., et al.: Deep learning with differential privacy. In: ACM SIGSAC, pp. 308–318 (2016)
2. Carlini, N., et al.: Extracting training data from large language models. In: USENIX Security Symposium (2021)
3. Chaudhry, A., Ranzato, M., Rohrbach, M., Elhoseiny, M.: Efficient lifelong learning with a-GEM. In: ICLR (2019)
4. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography Conference, pp. 265–284 (2006)
5. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014)
6. Farquhar, S., Gal, Y.: Differentially private continual learning. In: Privacy in Machine Learning and AI workshop at ICML (2018)
7. Fredrikson, M., Jha, S., Ristenpart, T.: Model inversion attacks that exploit confidence information and basic countermeasures. In: ACM SIGSAC (2015)
8. Goodfellow, I., Mirza, M., Xiao, D., Courville, A., Bengio, Y.: An empirical investigation of catastrophic forgetting in gradient-based neural networks. In: ICLR (2014)
9. Kartal, H., Liu, X., Li, X.: Differential privacy for the vast majority. ACM Trans. Manag. Inf. Syst. (TMIS) **10**(2), 1–15 (2019)
10. Lopez-Paz, D., Ranzato, M.: Gradient episodic memory for continual learning. In: Neural Information Processing Systems (NeurIPS) (2017)
11. Ostapenko, O., Puscas, M., Klein, T., Jahnichen, P., Nabi, M.: Learning to remember: a synaptic plasticity driven framework for continual learning. In: CVPR (2019)
12. Phan, H., Thai, M.T., Hu, H., Jin, R., Sun, T., Dou, D.: Scalable differential privacy with certified robustness in adversarial learning. In: ICML (2020)
13. Phan, N., Thai, M., Devu, M., Jin, R.: Differentially private lifelong learning. In: Privacy in Machine Learning (NeurIPS 2019 Workshop) (2019)
14. Phan, N., Jin, R., Thai, M.T., Hu, H., Dou, D.: Preserving differential privacy in adversarial learning with provable robustness. CoRR abs/1903.09822 (2019). http://arxiv.org/abs/1903.09822

15. Phan, N., et al.: Heterogeneous gaussian mechanism: preserving differential privacy in deep learning with provable robustness. In: IJCAI, pp. 4753–4759 (2019)
16. Phan, N., Wu, X., Hu, H., Dou, D.: Adaptive laplace mechanism: differential privacy preservation in deep learning. In: ICDM, pp. 385–394 (2017)
17. Schwarz, J., et al.: Progress & compress: a scalable framework for continual learning. In: ICML, pp. 4528–4537 (2018)
18. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE SP, pp. 3–18 (2017)
19. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: International Conference on Machine Learning, pp. 3987–3995 (2017)