Differentially Private Lifelong Learning

NhatHai Phan
New Jersey Institute of Technology
Newark, New Jersey, USA
phan@njit.edu

My T. Thai
University of Florida
Gainesville, Florida, USA
mythai@cise.ufl.edu

Devu M. Shila
Unknot.id
Orlando, Florida, USA
devums@unknot.id

Ruoming Jin
Kent State University
Kent, Ohio, USA
rjin1@kent.edu

Abstract

In this paper, we aim to develop a novel mechanism to preserve differential privacy (**DP**) in lifelong learning (**L2M**) for deep neural networks. Our key idea is to employ functional perturbation approaches in an original algorithm to preserve DP in both learning new tasks and memorizing acquired tasks in the past. Theoretical analysis shows that our mechanism significantly tighten the privacy loss, by avoiding the privacy budget accumulated in the continual learning and memorizing processes. Thorough evaluations show the effectiveness of our mechanism in preserving DP in L2M. Our study opens a new research avenue by uncovering the trade-off among privacy loss, model utility, and computational efficiency in L2M.

1 Introduction

Lifelong learning (**L2M**) is crucial for machine learning to acquire new skills quickly through the continual learning, pushing machine learning toward to a more human learning in reality. In this setting, a deep neural network (**DNN**) is trained with a stream of different tasks given streaming data. The DNN can quickly learn a new task, if the training algorithm can leverage the acquired knowledge after learning previous tasks. As in [1], to be practical, the model needs to be constrained in terms of amount of compute and memory required, as the goal is to quickly learn from a stream of data. As a result, it is quite challenging to train a L2M model with a high utility.

In addition to these challenges, L2M models are also vulnerable to adversarial model attacks, such as privacy model attacks [2–6], when the DNNs are trained on personal and highly sensitive data, such as clinical records [7–10], user profiles [11, 12], and bio-medical images [13, 14]. However, there is still a lack of scientific study to provide privacy protection to the private training data in L2M algorithms. To address this problem, we propose to preserve differential privacy (**DP**) [15], a rigorous cryptography-based formulation of privacy in probabilistic terms, in L2M.

However, this is a non-trivial task. In the continual learning, L2M models keep reading new coming data; while, in the memorizing process, the models randomly access a set of seen data samples stored in an episodic memory from the previous tasks. Compared with single trained models: **(1)** This continual learning and memorizing processes potentially cause a significant large privacy budget consumption, since the privacy budget can be accumulated in both learning and memorizing across tasks; and **(2)** Existing approaches [16–22] have not been developed to protect the training data in L2M. Thus preserving DP in L2M remains a largely open challenge.

Our Contributions. Motivated by this open problem, we propose to develop a novel *differentially private lifelong learning* (**DPL2M**) mechanism to preserve DP in L2M. In our mechanism, we incorporate the DP-preserving objective function (**DPAL**) [23] into the Average Gradient Episodic Memory (**A-gem**) [1] algorithm in a new approach, called **DPL2M**, to preserve DP in L2M. DPAL and A-gem are among state-of-the-art DP-preserving and L2M algorithms.

Preprint. Work in progress.

First, our episodic memory is a set of *fixed* batches, called *episodic memory batches*, each of which is a batch created from the learning process in each task. Note that, for each task, there is only one episodic memory batch used to memorize that task. Second, we apply the DPAL mechanism to perturb the objective functions in our DNNs. Finally, at each training step, the A-gem algorithm is applied to optimize the perturbed objective functions, given: 1) A batch of unseen data samples in the current task (for the continual learning); and 2) A number of randomly selected episodic memory batches (for the memorizing process). By doing this, batches are considered disjoint datasets in our algorithm. As a result, by applying the parallel composition property in DP [15], **(a)** we can avoid the privacy budget accumulation in both learning and memorizing processes across training steps, and **(b)** there is no extra privacy budget needed in the memorizing process, by avoiding the randomness leakage in the episodic memory batches.

To our knowledge, our mechanism establishes the first connection between *DP preservation* and *lifelong learning*. Such a mechanism will greatly extend the applicability of machine learning. Preliminary experiments conducted on permuted MNIST and permuted CIFAR-10 datasets [24] show promising results in preserving DP in L2M. In addition, we also explore the trade-off among privacy loss, model utility, and the amount of compute required to train a DPL2M model.

2 Background and Problem Definition

In this section, we revisit A-gem, DP, DPAL, and introduce our problem definition. In lifelong learning, we want to learn the sequence of tasks $\mathbf{T} = \{t_1, \ldots, t_M\}$ one by one, such that the learning of each new task will not forget the models learned for the previous tasks. Let D_τ is the dataset of the τ-th task. Each tuple contains data $x \in [-1, 1]^d$ and a *ground-truth label* $y \in \mathbb{Z}_K$, with K categorical outcomes. Each y is a one-hot vector of K categories $y = \{y_1, \ldots, y_K\}$. A single *true class label* $y_x \in y$ given $x \in D_\tau$ is assigned to only one of the K categories. On input x and parameters θ, a model outputs class scores $f : \mathbb{R}^d \to \mathbb{R}^K$ that maps inputs x to a vector of scores $f(x) = \{f_1(x), \ldots, f_K(x)\}$ s.t. $\forall k \in [1, K] : f_k(x) \in [0, 1]$ and $\sum_{k=1}^{K} f_k(x) = 1$. The class with the highest score is selected as the *predicted label* for x, denoted as $y(x) = \max_{k \in K} f_k(x)$. A loss function $L(f(\theta, x), y)$ presents the penalty for mismatching between the predicted values $f(\theta, x)$ and original values y, given the model parameters θ.

A-gem [1]. Let us denote $\mathbf{T}_l = \{t_1, \ldots, t_{\tau-1}\}$ s.t. $\tau < M$ is a set of tasks that have been learned. A-gem avoids catastrophic forgetting by storing an episodic memory $\mathbb{M}_t$ for each task t. When minimizing the loss on the current task τ, a typical approach is to treat the losses on the episodic memories of tasks $t < \tau$, given by $L(f(\theta, \mathbb{M}_t)) = \frac{1}{|\mathbb{M}_t|} \sum_{x \in \mathbb{M}_t} L(f(\theta, x), y)$, as inequality constraints. In A-gem, the objective function is as follows:

$$\theta^* = \arg\min_{\theta} L\big(f(\theta, D_\tau)\big) \text{ s.t. } L\big(f(\theta, \mathbb{M})\big) \leq L\big(f^{\tau-1}(\theta, \mathbb{M})\big) \text{ with } \mathbb{M} = \cup_{t<\tau} \mathbb{M}_t, \quad (1)$$

where $f^{\tau-1}$ is the model learned after training the task $\tau - 1$, indicating that the model will not forget previous learned tasks $\{t_1, \ldots, \tau - 1\}$ with the *memory replaying* constraint $L\big(f(\theta, \mathbb{M})\big) \leq L\big(f^{\tau-1}(\theta, \mathbb{M})\big)$. The constrained optimization problem of Eq. 1 can be solved quickly, when the updated gradient $\tilde{g}$ is as follows:

$$\tilde{g} = g - \frac{g^\top g_{ref}}{g_{ref}^\top g_{ref}} g_{ref} \quad (2)$$

where g is the the gradient update on the current task τ, g_{ref} is a gradient computed using a batch randomly sampled from the episodic memory $\mathbb{M}$.

Differential Privacy. The definition of DP is as follows:

Definition 1 *(ϵ, δ)-DP [15]. A randomized algorithm A fulfills (ϵ, δ)-DP, if for any two databases D and D' differing at most one tuple, and for all $O \subseteq Range(A)$, we have:*

$$Pr[A(D) = O] \leq e^\epsilon Pr[A(D') = O] + \delta \quad (3)$$

Here, ϵ controls the amount by which the distributions induced by D and D' may differ, δ is a broken probability. DP also applies to general metrics $\rho(D, D') \leq 1$, where ρ can be a Hamming metric as in Definition 1 and l_p-norms [25]. DP-preserving algorithms in deep learning can be categorized into two lines: 1) introducing noise into *gradients* of parameters [19–22], and 2) injecting noise into objective functions [16–18]. However, these existing mechanisms have not been designed to preserve DP in L2M. That is different from our goal in this study.

Differentially Private Adversarial Learning [23]. In DPAL, DNNs can be represented as: $f(x) = g(a(x, \theta_1), \theta_2)$, where $a(x, \theta_1)$ is a feature representation learning model with x as an input, and g will take the output of $a(x, \theta_1)$ and return the class scores $f(x)$. To preserve DP in training $f(x)$, DPAL leverage *Functional Mechanism* [26] to perturb the objective functions of $a(\cdot)$ and $g(\cdot)$, correspondingly denoted as $\mathcal{R}_{B_t}(\theta_1)$ and $L_{B_t}(\theta_2)$. In [23], $\mathcal{R}$ is the 1st-order polynomial function of the data reconstruction function (cross-entropy) in an auto-encoder given the batch B_t, and L is a 2nd-order polynomial objective function at the output layer with a tight sensitivity $\Delta_L \leq 2|\mathbf{h}_\pi|$, where $|\mathbf{h}_\pi|$ is the number of hidden neurons in the last hidden layer $\mathbf{h}_\pi$. The total budget to learn private parameters $\overline{\theta} = \{\overline{\theta}_1, \overline{\theta}_2\} = \arg\min_{\{\theta_1,\theta_2\}}(\overline{\mathcal{R}}_{B_t}(\theta_1) + \overline{L}_{B_t}(\theta_2))$ is $(\epsilon_1 + \epsilon_1/\gamma + \epsilon_2)$; where $\overline{\mathcal{R}}$ and $\overline{L}$ are the perturbed functions with the privacy budgets ϵ_1 and ϵ_2, given $\gamma = \frac{2\Delta_{\mathcal{R}}}{m\|\overline{\theta}_1\|_{1,1}}$ and $\|\overline{\theta}_1\|_{1,1}$ is the maximum 1-norm of θ_1's columns [27], the global sensitivity $\Delta_{\mathcal{R}}$ of $\mathcal{R}$, and the batch size m of B_t.

Privacy Budget Accumulation. The optimization of DPAL is repeated in T steps, without using additional information from the original data: **(1)** It only reads perturbed inputs and perturbed coefficients; **(2)** A single draw of noise is used during training; and **(3)** To avoid leaking information about the randomness, generated batches are not changed between epochs. Consequently, the privacy budget consumption is not accumulated at each training step, i.e., independent of T.

3 Differential Private Lifelong Learning

Our DP-preserving mechanism, called DPL2M, is presented in Alg. 3. Given the current learning task $\tau \in \mathbf{T}$, our algorithm will scan the data D_τ once in a batch mechanism (Lines 2-4). If τ is the first task, we optimize our parameters normally with perturbed objective functions $\overline{R}_B(\theta_1)$ and $\overline{L}_B(\theta_2)$ by applying DPAL mechanism on the batch $B \in \mathbf{B}$ (Lines 5-6). Then, a batch B, randomly selected from the set of batches $\mathbf{B}$, will be included in the episodic memory $\mathbb{M}$ (Lines 11-12). If τ is not the first task, we take an episodic memory batch B_e randomly from a set of batches in $\mathbb{M}$. Then, we compute $\tilde{g}_1$ with Eq. 2 given B, B_e, and $\overline{\mathcal{R}}$, and compute $\tilde{g}_2$ with Eq. 2 given B, B_e, and $\overline{L}$. Given $\tilde{g}_1$ and $\tilde{g}_2$, we update the model parameters θ_1 and θ_2 (Lines 8-10). Finally, we return $(\epsilon_1 + \epsilon_1/\gamma + \epsilon_2)$-DP parameters $\theta = \{\theta_1, \theta_2\}$.

Since generated batches are disjoint and fixed in the training $\mathbf{B}$ and in the episodic memory $\mathbb{M}$, our algorithm closely follows the privacy budget accumulation property of DPAL [23]. In fact, each batch is considered a disjoint dataset and the total privacy budget is the maximum privacy budget applied in each of the batch; following the parallel composition in DP [28].

Algorithm 1 DP Lifelong Learning

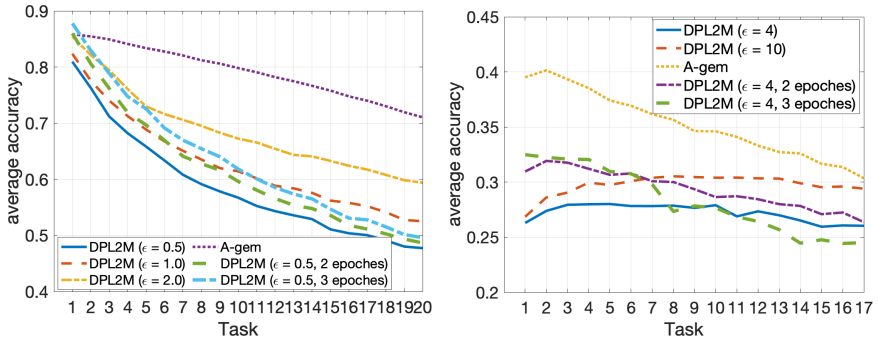
Input: A sequence of tasks $\mathbf{T} = \{t_1, \ldots, t_M\}$, databases $\{D_1, \ldots, D_M\}$, batch size m, privacy budgets: ϵ_1 and ϵ_2, learning rate ϱ

1: **Randomly Initialize** $\theta = \{\theta_1, \theta_2\}$, $\mathbb{M} = \emptyset$
2: **for** $\tau \in \mathbf{T}$ **do**
3: $\quad\mathbf{B} = \{B_1, \ldots, B_{N/m}\}$ s.t. $\forall B \in \mathbf{B}$: B is a random batch with the size m, $B_1 \cap \ldots \cap B_{N/m} = \emptyset$, and $B_1 \cup \ldots \cup B_{N/m} = D_\tau$
4: $\quad$**for** $B \in \mathbf{B}$ **do**
5: $\quad\quad$**if** $\tau == 0$ **then**
6: $\quad\quad\quad$**Descent:** $\theta_1 \leftarrow \theta_1 - \varrho\nabla_{\theta_1}\overline{\mathcal{R}}_B(\theta_1)$; $\theta_2 \leftarrow \theta_2 - \varrho\nabla_{\theta_2}\overline{L}_B(\theta_2)$
7: $\quad\quad$**else**
8: $\quad\quad\quad$Take an episodic memory batch B_e randomly from a set of batches in $\mathbb{M}$
9: $\quad\quad\quad$Compute $\tilde{g}_1$ with Eq. 2 given B, B_e, and $\overline{\mathcal{R}}$; Compute $\tilde{g}_2$ with Eq. 2 given B, B_e, and $\overline{L}$
10: $\quad\quad\quad$**Descent:** $\theta_1 \leftarrow \theta_1 - \varrho\tilde{g}_1$; $\theta_2 \leftarrow \theta_2 - \varrho\tilde{g}_2$
11: $\quad$Take a random batch $B \in \mathbf{B}$
12: $\quad\mathbb{M} = \mathbb{M} \cup \{B\}$

Output: $(\epsilon_1 + \epsilon_1/\gamma + \epsilon_2)$-DP parameters $\theta = \{\theta_1, \theta_2\}$

4 Experimental Results

We have carried out an experiment on permuted MNIST [24] and permuted CIFAR-10 datasets. Permuted MNIST is a variant of MNIST [29] dataset of hand-written digits where each task has a random permutation of the input pixels which is applied to all the images of that task. We adapt



(a) Permuted MNIST (20 tasks) (b) Permuted CIFAR-10 (17 tasks)

Figure 1: Average accuracy on the permuted MNIST and CIFAR-10 datasets.

Table 1: Average forgetting measure.

	DPL2M ($\epsilon = 0.5$)	DPL2M ($\epsilon = 1$)	DPL2M ($\epsilon = 2$)	A-gem
Permuted MNIST	0.305 ± 0.00886	0.278 ± 0.00907	0.237 ± 0.00586	0.162 ± 0.01096
	DPL2M ($\epsilon = 4$)	DPL2M ($\epsilon = 10$)	A-gem	
Permuted CIFAR-10	0.057 ± 0.002	0.0369 ± 0.00389	0.133 ± 0.00859	

this approach to permute the CIFAR-10 dataset, in which a random permutation of the input pixels including three color channels is applied to all the images of that task. We call this dataset a *permuted CIFAR-10* dataset. Our **DPL2M** mechanism is evaluated in comparison with **A-gem** [1], which is one of the state-of-the-art L2M algorithm. *Note that A-gem does not preserve DP.* We apply two well-applied metrics, including the *average accuracy* and the *average forgetting measure* in [30] to evaluate our proposed algorithm.

Model Configuration. In the permuted MNIST dataset, we used three convolutional layers (32 64, and 96 features). Each hidden neuron connects with a 5x5 unit patch. A fully-connected layer has 512 units. The batch size m was set to 2,500, and learning rate $\varrho = 0.1$. In the permuted CIFAR-10, we used a Resnet-18 network (64, 64, 128, 128, and 160 features) with kernels (4, 3, 3, 3, and 3). One fully-connected layer has 256 neurons. The batch size m was set to 500, and learning rate $\varrho = 0.2$. The number of runs for each experiment is 5.

MNIST. Figure 1a and Table 1 illustrate the average accuracy and forgetting measure of each model as a function of the privacy budget $\epsilon = (\epsilon_1 + \epsilon_1/\gamma + \epsilon_2)$ on the MNIST dataset. It is clear that there is a gap in terms of average accuracy between the A-gem (i.e., a noiseless model) and our DPL2M models given a small number of tasks. However, the gap is significantly increased when the number of tasks increased (23.3% at $\epsilon = 0.5$ with 20 tasks). In addition, the larger privacy budget (i.e., $\epsilon = 2.0$), the higher average accuracy we can achieve, compared with smaller privacy budgets (i.e., $\epsilon = 0.5$). Even though, our DPL2M model can achieve a relative high average accuracy given a small number of tasks, the result clearly illustrates that it is challenging to preserve DP in L2M while retaining a high model utility. The average forgetting measure further strengthens our observation, since our DPL2M models have a relative high average forgetting value, compared with the noiseless model A-gem. One potential solution is to increase the number of training epochs on each task. When we train our DPL2M model with 2 or 3 epochs per task (denoted DPL2M with $\epsilon = 0.5$, 2 or 3 epochs), the average accuracy is significantly improved. However, this will come with a computational cost.

CIFAR-10. Results on the CIFAR-10 dataset strengthen our observations (Figures 1b and Table 1). The only small difference is that the average forgetting values in our DPL2M are better than the A-gem algorithm. However, this may be caused by the low average accuracy of our DPL2M models in the permuted CIFAR-10 dataset. In fact, permuted CIFAR-10 tasks are difficult to classify, even with the noiseless model A-gem. Further investigation is needed to shed light into this problem.

5 Conclusion

In this paper, we established the first connection among DP preservation to protect the training data and lifelong learning. Our proposed mechanism combines DPAL and A-gem in a holistic way, in order to preserve DP in L2M. Our model shows promising results and opens a long-term avenue to achieve better model utility and computational efficiency under strong privacy guarantees in L2M.

References

[1] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with a-GEM," in *International Conference on Learning Representations*, 2019.

[2] R. Shokri, M. Stronati, and V. Shmatikov, "Membership Inference Attacks against Machine Learning Models," *ArXiv e-prints*, 2016.

[3] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15, 2015, pp. 1322–1333.

[4] Y. Wang, C. Si, and X. Wu, "Regression model fitting under differential privacy and model inversion attack," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, 2015, pp. 1003–1009.

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

[6] N. Papernot, P. D. McDaniel, A. Sinha, and M. P. Wellman, "Towards the science of security and privacy in machine learning," *CoRR*, vol. abs/1611.03814, 2016.

[7] E. Choi, A. Schuetz, W. F. Stewart, and J. Sun, "Using recurrent neural network models for early detection of heart failure onset," *Journal of the American Medical Informatics Association*, 2016.

[8] H. Li, X. Li, M. Ramanathan, and A. Zhang, "Prediction and informative risk factor selection of bone diseases," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 12, no. 1, pp. 79–91, 2015.

[9] R. Miotto, L. Li, B. A. Kidd, and J. T. Dudley, "Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records," *Scientific Reports*, vol. 6, 2016.

[10] A. Perotte, R. Ranganath, J. S. Hirsch, D. Blei, and N. Elhadad, "Risk prediction for chronic kidney disease progression using heterogeneous electronic health record data and time series analysis," *Journal of the American Medical Informatics Association*, vol. 22, no. 4, pp. 872–880, 2015.

[11] M. Roumia and S. Steinhubl, "Improving cardiovascular outcomes using electronic health records," *Current Cardiology Reports*, vol. 16, no. 2, p. 451, 2014.

[12] J. Wu, J. Roy, and W. F. Stewart, "Prediction modeling using EHR data: challenges, strategies, and a comparison of machine learning approaches." *Medical care*, vol. 48, no. 6 Suppl, 2010.

[13] S. M. Plis, D. R. Hjelm, R. Salakhutdinov, E. A. Allen, H. J. Bockholt, J. D. Long, H. J. Johnson, J. S. Paulsen, J. A. Turner, and V. D. Calhoun, "Deep learning for neuroimaging: a validation study," *Frontiers in Neuroscience*, vol. 8, p. 229, 2014.

[14] M. Helmstaedter, K. L. Briggman, S. C. Turaga, V. Jain, H. S. Seung, and W. Denk, "Connectomic reconstruction of the inner plexiform layer in the mouse retina," *Nature*, vol. 500, no. 7461, pp. 168–174, 2013.

[15] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," *Theory of Cryptography*, pp. 265–284, 2006.

[16] N. Phan, Y. Wang, X. Wu, and D. Dou, "Differential privacy preservation for deep auto-encoders: an application of human behavior prediction," in *AAAI'16*, 2016, pp. 1309–1316.

[17] N. Phan, X. Wu, and D. Dou, "Preserving differential privacy in convolutional deep belief networks," *Machine Learning*, 2017.

[18] N. Phan, X. Wu, H. Hu, and D. Dou, "Adaptive laplace mechanism: Differential privacy preservation in deep learning," in *IEEE ICDM'17*, 2017.

[19] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," *arXiv:1607.00133*, 2016.

[20] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning." in *CCS'15*, 2015, pp. 1310–1321.

[21] L. Yu, L. Liu, C. Pu, M. Gursoy, and S. Truex, "Differentially private model publishing for deep learning," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 326–343.

[22] J. Lee and D. Kifer, "Concentrated differentially private gradient descent with adaptive per-iteration privacy budget," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1656–1665.

[23] N. Phan, R. Jin, M. T. Thai, H. Hu, and D. Dou, "Preserving differential privacy in adversarial learning with provable robustness," *CoRR*, vol. abs/1903.09822, 2019.

[24] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," vol. 114, no. 13, pp. 3521–3526, 2017.

[25] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi, "Broadening the scope of differential privacy using metrics," in *Privacy Enhancing Technologies*, E. De Cristofaro and M. Wright, Eds., 2013, pp. 82–102.

[26] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett, "Functional mechanism: regression analysis under differential privacy," *PVLDB*, vol. 5, no. 11, pp. 1364–1375, 2012.

[27] Operator norm, "Operator norm," 2018. [Online]. Available: https://en.wikipedia.org/wiki/Operator_norm

[28] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, pp. 211–407, Aug. 2014. [Online]. Available: http://dx.doi.org/10.1561/0400000042

[29] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[30] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. S. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," *CoRR*, vol. abs/1801.10112, 2018.