Differentially Private Graph Neural Networks for Whole-Graph Classification

Tamara T. Mueller, Johannes C. Paetzold, Chinmay Prabhakar, Dmitrii Usynin, Daniel Rueckert, *Fellow, IEEE*, and Georgios Kaissis

***Abstract*—Graph Neural Networks (GNNs) have established themselves as state-of-the-art for many machine learning applications such as the analysis of social and medical networks. Several among these datasets contain privacy-sensitive data. Machine learning with differential privacy is a promising technique to allow deriving insight from sensitive data while offering formal guarantees of privacy protection. However, the differentially private training of GNNs has so far remained under-explored due to the challenges presented by the intrinsic structural connectivity of graphs. In this work, we introduce a framework for differential private graph-level classification. Our method is applicable to graph deep learning on multi-graph datasets and relies on differentially private stochastic gradient descent (DP-SGD). We show results on a variety of datasets and evaluate the impact of different GNN architectures and training hyperparameters on model performance for differentially private graph classification, as well as the scalability of the method on a large medical dataset. Our experiments show that DP-SGD can be applied to graph classification tasks with reasonable utility losses. Furthermore, we apply explainability techniques to assess whether similar representations are learned in the private and non-private settings. Our results can also function as robust baselines for future work in this area.**

Manuscript received 25 February 2022; revised 29 September 2022; accepted 24 November 2022. Date of publication 12 December 2022; date of current version 5 May 2023. Recommended for acceptance by L. Wang. *(Corresponding author: Tamara T. Mueller.)*

Tamara T. Mueller and Dmitrii Usynin are with the Chair for AI in Medicine and Healthcare, Department of Computer Science, Technical University of Munich, 80333 Munchen, Germany, and also with the Department of Diagnostic and Interventional Radiology, School of Medicine, Technical University of Munich, 80333 Munchen, Germany (e-mail: tamara.mueller@tum.de; dmitrii.usynin@tum.de).

Johannes C. Paetzold is with the Department of Informatics, Technical University of Munich, 80333 Munchen, Germany, and also with the Institute for Tissue Engineering and Regenerative Medicine, Helmholtz Zentrum Munchen, 85764 Oberschleißheim, Germany (e-mail: johannes.paetzold@tum.de).

Chinmay Prabhakar is with the Department of Quantitative Bio Medicine, University of Zurich, 8006 Zurich, Switzerland (e-mail: chinmay.prabhakar@uzh.ch).

Daniel Rueckert is with the Chair for AI in Medicine and Healthcare, Department of Computer Science, Technical University of Munich, 80333 Munchen, Germany, and also with the Department of Computing, Imperial College London, SW7 2BX London, U.K. (e-mail: daniel.rueckert@tum.de).

Georgios Kaissis is with the Chair for AI in Medicine and Healthcare, Department of Computer Science, Technical University of Munich, 80333 Munchen, Germany, also with the Department of Diagnostic and Interventional Radiology, School of Medicine, Technical University of Munich, 80333 Munchen, Germany, also with the Department of Computing, Imperial College London, SW7 2BX London, U.K., and also with the Institute for Machine Learning in Biomedical Imaging, Helmholtz Zentrum Munich, 85764 Munich, Germany (e-mail: g.kaissis@tum.de).

The source code is available at https://github.com/tamaramueller/DP-GNNs.

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2022.3228315, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2022.3228315

***Index Terms*—Differential privacy, graph neural networks.**

I. Introduction

THE introduction of geometric deep learning, and more specifically Graph Neural Networks (GNNs) [1], [2], has enabled training ML models on data in non-Euclidean spaces with state-of-the-art performance in many applications. GNNs are able to directly leverage the graph structure of the data and propagate the information stored in nodes of the graph along the edges connecting nodes with each other. Thus, the information flow through the network respects the underlying topology of the graph.

In general, GNNs have been employed in three types of problem areas: node classification, edge prediction, and graph classification. In this work, we focus on graph classification tasks. In the setting of graph classification (also termed graph property prediction), the dataset consists of multiple independent graphs and a GNN is trained to predict one label for each individual graph, predicting a specific property of the whole graph. Application areas of geometric deep learning range from social networks [3] to medical applications [4], [5], drug discovery or molecule classification [6], spatial biological networks [7] and shape analysis [8]. Drawing meaningful insights in many of these application areas fundamentally relies upon the utilisation of privacy-sensitive, often scarce, training data belonging to individuals. For example when using functional magnetic resonance imaging (fMRI) for identifying disease-specific biomarkers of brain connectivity like in [4] and [9], the graph data encodes sensitive, patient-specific medical data.

The reliance on sensitive data in machine learning holds potential for misuse and can therefore be associated with the risks to individual participants' privacy. Various machine learning contexts have been shown vulnerable to be exploited by malicious actors, resulting in a leakage of private attributes [10], of membership information [11] or even in full dataset reconstruction [12], [13]. In graph machine learning, the data and the models trained on that data are *by design* more vulnerable to adversarial attacks targeting privacy of the data owners. This is attributed to the fact that graphs incorporate additional information that is absent from typical Euclidean training contexts, such as the relational information about the nodes in the graph. This auxiliary, highly descriptive information can be leveraged by an adversary to assist them in sensitive information extraction, which has been demonstrated in a number of prior works [14],

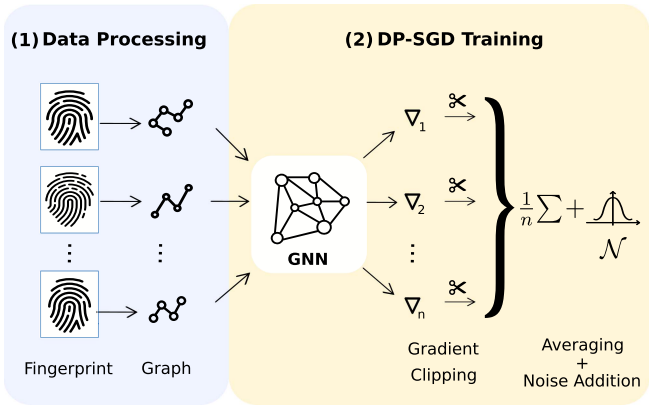

Fig. 1. Overview of our differentially private training method for graph classification on a fingerprint dataset. In step (1) the fingerprint images are converted into graphs, which are then in step (2) passed to a GNN model, which is trained with differentially private stochastic gradient descent (DP-SGD). The individual gradients are clipped, then averaged and Gaussian noise is added.

[15], [16]. Such attacks can also be facilitated by the choice of learning context in cases the model is trained collaboratively. For instance, transductive collaborative learning renders attacks aimed at disclosing the membership of individual training points trivial [15]. Of note, such additional information embedded in graphs is often essential for effective GNN training and is, thus, non-trivial to privatise or remove, as it would be highly detrimental to the performance of the model.

It is thus apparent that the implementation of privacy-enhancing techniques is required to facilitate the training of models of sensitive graph-structured data, but such techniques must also respect the particularities of graph machine learning. Our work utilises a formal method of privacy preservation termed differential privacy (DP) [17] which, when applied to machine learning training, is able to objectively quantify the privacy loss for individual input data points. DP methods have been successfully applied to numerous problems such as medical image analysis [18], [19], natural language processing (NLP) [20], reinforcement learning [21] or generative models [22] and have shown promising results. DP guarantees that the information gain from observing the output of an algorithm trained on datasets differing in one individual is (sometimes with high probability), bounded by a (typically small) constant.

In this work, motivated by the above-mentioned requirements for objective privacy guarantees in machine learning tasks involving graph-structured data, we study the problem of efficient differentially private graph neural network training for graph classification tasks (Fig. 1). To the best of our knowledge, our is the first work that demonstrates the application of differential privacy to whole graph classification tasks. We investigate and evaluate privacy-utility trade-offs on several datasets and compare the learned representations between DP and non-DP trained models using explianability methods for GNNs. This comparison can offer insights into differences regarding model parameters, which are considered as important for the decision making, under different training conditions. In our work, we extend the utilisation of differentially private stochastic gradient descent (DP-SGD) [23], a technique designed for the training of regular neural networks. Due to its compatibility with existent deep learning workflows, it can be seamlessly adapted to GNN use cases and therefore offers high generalisability to new model architectures and problem spaces. We demonstrate that DP-SGD can be applied to graph learning and evaluate our results with respect to privacy budgets and network performance on five different datasets. Combined with our investigation of the explainability technique *GNNExplainer* to determine differences between DP and non-DP models, this work can serve as a baseline for future work in this area. Our contributions can be summarised as follows:

1) We formally extend the application of DP-SGD to graph classification tasks with GNNs;
2) To demonstrate its utility, we apply our method to commonly utilised graph neural networks on a number of benchmark and real-world datasets and investigate the effects of DP training on model utility and privacy guarantees;
3) To assess how similar the representations between privately and non-privately trained models are, we apply GNNExplainer, a state-of-the-art explainability technique tailored to graph neural networks.

II. Related Work

Specific facets of differentially private graph analysis have been addressed in prior work: Since the introduction of differentially private computation on graph data in 2007 by Nissim et al. [24], *node-level* and *edge-level* DP have been established as the two DP formalisms on graphs [25]. As discussed in the *Theory* section below, the definition of DP relies on the notion of adjacent datasets, that is, datasets differing in the data of one individual. In the setting of tabular data for example, two datasets are adjacent if they differ in one row. In node-level DP, two graph datasets are interpreted as adjacent if one node and its incident edges is inserted or removed. For edge-level DP, on the other hand, two datasets are regarded as adjacent if they differ in exactly one edge. As real-world graphs are prevalently sparse, the removal of a single node can severely alter the graph's structure [26], whereas removal of an edge typically has a less severe impact on the resulting graph structure.

Implementations of the aforementioned techniques have been presented in the context of graph neural network training. For instance, Igamberdiev et al. [27] explore the application of DP on Graph Convolutional Networks (GCNs) [28] for node classification. They evaluate privacy guarantees for text classification on benchmark datasets and achieve rigorous privacy guarantees while maintaining high model performance. Daigavan et al. [29] formalise the notion of node-level DP on one-layer GNNs with an extension of privacy amplification by sampling to GNNs and evaluate their method on several benchmark datasets in node classification tasks. Different approaches to the here introduced application of differential privacy have been explored in the context of federated learning on graphs and locally private graph neural network training. Zhou et al. [30], for example, introduce

a vertically federated GNN for node classification tasks and Sajadmanesh et al. [31] introduce a framework to train locally private GNNs. These works stand in contrast to the notion of graph-level DP, which ensures data privacy of a graph as a whole.

DP is one of the most frequently used methods in deep learning that offer privacy guarantees. Furthermore, it is the only approach that gives formal guarantees for privacy as well as a quantification of the guaranteed privacy. However, there exist other empirical methods next to differential privacy that allow to privatise sensitive data of individuals, which have also been applied to GNN training in node classification and edge prediction tasks. Liao et al. [32] introduce a method to filter specific node feature attributes using adversarial training of GNNs and therefore achieve a strong defence against inference attacks. Their method is in parallel to our work, since they do not ensure differential privacy guarantees for each graph as a whole, but instead address an information obfuscation problem where the goal of an adversary is to infer specific node attributes in a graph. Other works like the privacy-preserving network embedding introduced by Han et al. [33] and the privacy-preserving GCN model by Hu et al. [34] also do not give differential privacy guarantees. They show other methods for protecting private links in graph-structured data [33] and user-specific sensitive node features [34], respectively.

However, to our knowledge, the application of DP algorithms specifically to graph property prediction has neither been formalised nor evaluated.

III. Theoretical Preliminaries

In this section, we introduce and formalise the theory to train graph neural networks for graph property prediction using the concept of differentially private stochastic gradient descent (DP-SGD).

A. GNNs for Graph Property Prediction

The objective of graph classification (also known as graph property prediction) is to predict a specific property of interest for an entire graph $\mathcal{G}$. In our examples, $\mathcal{G}$ represents an unweighted and undirected graph with $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of edges. The nodes $\mathcal{V}$ are represented by a vector or a matrix of node features. Graph classification aims to predict a property for each graph $\mathcal{G}_i$, $i \in [1, \dots, N]$ in a multi-graph dataset $D = \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_N\}$ with N graphs. A GNN used for graph property prediction needs to map the embedded node features into a unified representation of the whole graph using a readout layer (e.g. global max pooling). This single unified embedded graph representation allows to learn a prediction for the whole graph.

B. Differential Privacy

Differential Privacy (DP) [17] is a theoretical framework and collection of techniques aimed at enabling analysts to draw conclusions from datasets while safeguarding individual privacy. Intuitively, an algorithm preserves DP if its outputs are approximately invariant to the inclusion or exclusion of a single individual in the dataset over which the algorithm is executed. The DP guarantee is given in terms of probability mass/density of the algorithm's outputs.

In the current study, we assume that an analyst $\mathcal{A}$ is entrusted with a multi-graph database D of cardinality N containing privacy-sensitive graphs $\mathcal{G}_i \in D, i \in [1, \dots, N]$ by a group of individuals. We assume that each individual's graph is only present in the database once. From D, an *adjacent* database D' of cardinality $N \pm 1$ can be constructed by adding or removing a single individual's graph. We denote adjacency by $D \simeq D'$. The set (*universe*) of all adjacent databases forms a metric space X with associated metric d_X, in our case, the Hamming metric.

We additionally assume that $\mathcal{A}$ executes a *query function* f over an element of X. In our study, the application of f represents a sequential composition of the forward pass, loss calculation and gradient computation of a graph neural network for each individual input (training example) to f. We then define the global L_2-sensitivity of f as follows:

Definition III.1 (Global L_2-sensitivity of f). Let f, X and d_X be defined as above. Additionally, let Y be the metric space of f's outputs with associated metric d_Y. When Y is the Euclidean space and d_Y the L_2 metric, we define the (global) L_2-sensitivity Δ of f as:

$$\Delta := \max_{D, D' \in X, D \simeq D'} \frac{d_Y(f(D), f(D'))}{d_X(D, D')}. \tag{1}$$

We remark that the maximum is taken over all adjacent database pairs in X. Moreover, Δ describes a Lipschitz condition on f, implying that $\Delta \equiv K_f$, where K_f is the Lipschitz constant of f. This in turn implies that $\Delta = \sup \|\nabla f\|_2$. In our case, the L_2-sensitivity of the loss function therefore corresponds to the upper bound on its gradient.

We can now define the Gaussian Mechanism on f:

Definition III.2 (Gaussian Mechanism). Let f, Δ be defined as above. The Gaussian mechanism $\mathcal{M}$ operates on the outputs of f, $\mathbf{y} = f(x)$, where $\mathbf{y} \in \mathbb{R}^n$ as follows:

$$\mathcal{M}(\mathbf{y}) = \mathbf{y} + \xi, \tag{2}$$

where $\xi \sim \mathcal{N}(0, \sigma \mathbb{I}^n)$, σ is calibrated to Δ, and $\mathbb{I}^n$ is the identity matrix with n diagonal elements.

When σ is appropriately calibrated to Δ, $\mathcal{M}$ preserves (ε, δ)-DP:

Definition III.3 ((ε, δ)-DP). $\mathcal{M}$ preserves (ε, δ)-DP if, $\forall\, S \subseteq \mathrm{Range}(\mathcal{M})$ and all adjacent databases D, D' in X:

$$\mathbb{P}(\mathcal{M}(D) \in S) \leq e^{\varepsilon} \mathbb{P}(\mathcal{M}(D') \in S) + \delta. \tag{3}$$

We remark that the definition is symmetric.

C. DP-SGD

Abadi et al. [23] introduced an extension to stochastic gradient descent (SGD), termed DP-SGD to enable the differentially private training of neural networks. Here, at each training step, the Gaussian Mechanism is used to privatise the individual gradients of each training example before the model parameters are updated. However, since the sensitivity of the loss function in deep neural networks is – in general – unbounded, the gradient

L_2-norm of each individual training example is *clipped*, that is, projected to an L_2-ball of a pre-defined radius to artificially induce a bounded sensitivity condition before noise is applied. Tracking the privacy expenditure over the course of training (*privacy accounting*) is enabled through the *composition* property of DP, stating that repeated application of DP algorithms over the same data predictably degrades the privacy guarantees. In our study, a relaxation of DP termed Rényi DP (RDP) [35] is used for privacy accounting, due to its favourable compositional properties. RDP guarantees can be converted to (ε, δ)-DP.

DP-SGD is widely regarded as the gold-standard for privacy preserving deep learning, as it is generically applicable to all types of gradient-based optimisation and protects both features and labels. It can be easily adapted to e.g. regression or generative modelling workflows. Other DP methods are substantially less flexible [23]. Private aggregation of teacher ensembles (PATE) [36] for example, is only usable for classification tasks and requires large public datasets, which, especially in the medical field, cannot be procured in many cases.

D. DP Notions on Graph-Structured Datasets

There exist three major tasks in the context of GNN training: node classification/regression, edge prediction, and graph classification/regression. Similar to the existence of multiple tasks in graph deep learning, there also exist different notions of DP on graph-structured datasets, that specifically relate to different notions of adjacent datasets. For *node-level DP*, two datasets are interpreted as adjacent, if they vary in one node and all its adjacent edges [25]. If the notion of adjacent datasets is based on the inclusion or exclusion of one edge, this notion of DP is called *edge-level DP* [37]. Node-level DP is a strictly stronger privacy guarantee in comparison to edge-level DP [26]. As real-world graphs are prevalently sparse, the removal of a single node can severely alter the graph's structure [26], whereas removal of an edge typically has a less severe impact on the resulting graph structure. However, in case of multi-graph datasets, a third notion of DP can come into play. Here, two datasets can be defined to be adjacent if they differ in one graph. The resulting DP-guarantee is then *graph-level DP* [38], which we utilise in this work. For more details we refer to [38].

IV. Experiments

A. Datasets

We evaluate the application of DP-SGD in the context of graph property prediction tasks on five datasets. We rely on three publicly available benchmark datasets, a dataset from the U.K. Biobank [39], and a synthetic dataset, generated to provide a reproducible and easy to control proof-of-concept. The three benchmark datasets tackle the problems of molecule classification (Molbace), fingerprint classification, and Left Bundle Branch Block (LBBB) detection on electrocardiogram (ECG) data. Table I provides an overview of the datasets and their characteristics and more detailed information about the datasets can be found in the Appendix, (available online).

TABLE I
Overview of the Utilised Datasets and Their Characteristics

Dataset	Mean num. nodes	Num. graphs	Num. node features	Num. classes
Synthetic	20	1,000	9	2
Fingerprints	7.6	1,900	2	4
Molbace	34	1,513	9	2
ECG	12	1,125	512	2
Organ Meshes	7546.7	151,910	3	5

We report the mean number of nodes, in case the dataset contains graphs of varying sizes.

Synthetic Dataset: In order to derive a proof-of-concept of the novel application of DP-SGD on graph classification tasks, we construct a synthetic dataset, in which parameters can be manually controlled to create an easily controllable dataset where high accuracy can be achieved in a non-private setting and we can evaluate how DP-SGD training at different strengths of privacy guarantee impacts utility. We generate 1,000 individual Erdős-Rényi graphs, equally distributed to two classes. Each graph consists of twenty nodes which contain nine features each. The node features are sampled from a normal distribution with different mean values and the same standard deviation, corresponding to the label class of the graph. The edge connection probabilities vary slightly between the two classes.

Fingerprints Dataset: Fingerprint classification aims to separate images of fingerprints into the different classes - arch, left, right, and whorl - from the Galton-Henry classification system [40], [41]. A large within-class variability and a small separation between classes makes fingerprint classification a challenging task [42]. We rely on the dataset introduced by Riesen et al. [43] and provided by TU Datasets [44] to perform differentially private graph classification on fingerprints. The graphs are extracted from the images based on directional variance and the task follows the Galton-Henry classification scheme of five classes. We merge the five classes into four classes following the approach described in [43]. Differentially private ML naturally befits this task, as it allows one to privatise the utilisation of the uniquely identifying fingerprint data for e.g. training machine learning models in tasks such as automated authentication.

Molbace Dataset: To perform molecule classification in a binary graph classification setting, we use the benchmark dataset *Molbace* from the OGB database [45], where the *Molbace* dataset is adapted from MoleculeNet [46]. It consists of 1,513 graphs, where each graph represents a molecule. Edges represent molecular bonds and nodes correspond to the atoms in the molecule. Each node contains 9 node features and the average number of nodes per graph is 34. We split the dataset into 1,210 training graphs, 152 test graphs and 151 validation graphs. Node features contain atom features; for example the atomic number, chirality, formal charge, or whether the atom is in a ring or not. The prediction task of this dataset is to correctly classify whether the molecule inhibits HIV virus replication [45]. Such a task is representative of federated learning workflows with per-site (local) DP application, in which e.g. several pharmaceutical companies wish to jointly train a model for molecule

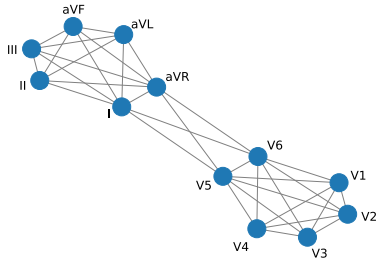

Fig. 2. Graph visualisation of ECG data. We connected the different signal channels based on the medical location of the leads as well as prior knowledge. Leads I, II, III, aVF, aVL, and aVR are located on the extremities and the remaining leads on the chest.

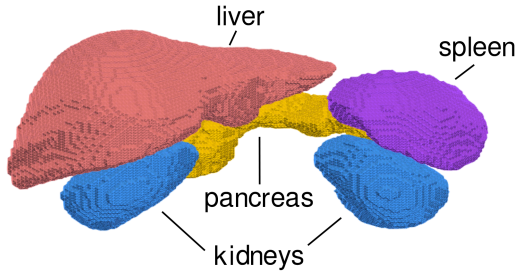

Fig. 3. Organ meshes extracted from segmenations of U.K. Biobank data [39]. The organs shown in this figure are the liver (coral), the spleen (purple), the left and right kidneys (blue) and the pancreas (yellow).

property prediction, while wishing to limit the disclosure of their (possibly proprietary) molecule structures from third parties.

ECG Dataset: For the task of electrocardiogram (ECG) classification, we use the publicly available ECG dataset from the China Physiological Signal Challenge (CPSC) 2018 challenge dataset [47]. We formulate a classification task between ECGs showing signs of a Left Bundle Branch Block and normal ECGs showing a sinus rhythm. The ECG data consists of twelve ECG signal channels (*leads*), recorded at different locations on the human torso and extremities. Leads affixed to the extremities constitute signal channels I, II, II, aVR, aVF and aVL. Leads affixed to the chest are used to derive signal channels V1 through V6. To construct a graph dataset from the ECG data, we utilise this medical motivation and divide the ECG extremity signal channels from the chest signal channels by fully connecting the extremity and chest subgraphs. In addition, we utilise prior knowledge about the leads which are typically used by physicians to delineate LBBB from sinus rhythm and thus connected channels I, aVR, V5, and V6. The structure of those graphs is visualised in Fig. 2. The dataset we use contains ECG data of 1,125 subjects. As ECG signals are periodic, we sub-sample the signals by only retaining the first 512 signal points of each channel, leading to 512 node features in the graphs. The binary classification dataset is highly imbalanced with 207 subjects showing signs of LBBB and 918 having normal ECG curves. Evidently, ECG data, like all medical data is highly sensitive, and thus requires formal methods of privacy protection.

Organ Meshes Dataset: To investigate the scalability of our method to large sensitive medical datasets, we perform an organ mesh classification task on 151,910 organ surface meshes extracted from 30,382 subjects from the U.K. Biobank database [39]. As a first step, the five organs liver, spleen, left and right kidney, and pancreas were segmented using the segmentation pipeline of [48]. Secondly, the organ meshes were extracted from those segmentations using the *marching cubes* algorithm [49] implementation by [50]. Fig. 3 shows an example visualisation of the surface meshes of one subject. Each organ is represented as an individual graph in the dataset and the task is to classify which of the five organs is represented by the surface mesh. Node features contain the three dimensional coordinates of the organs with respect to the original magnetic resonance imaging (MRI) scan of the subject.

B. GNN Models for Graph Classification and DP-SGD Training

Since the adoption of deep learning techniques to graph learning, most state-of-the-art methods for graph classification rely on a variant of *message passing* to aggregate information across the nodes [51], [52], [53], [54], [55].

For our experiments, we implement a variety of GNN models to compare performance and evaluate the impact of DP on different graph learning techniques. We use GraphSAGE [56], Graph Attention Networks (GATs) [57], Graph Convolutional Networks (GCNs) [28], and chebyshev spectral graph convolutions (Cheb) [58]. For each dataset, we perform hyperparameter searches, leading to different models for each application. The depth of the GNNs varies from two to three layers with/without Instance Normalisation layers and with/without dropout, depending on the problem space. We do not use Batch Normalisation because of its incompatibility with differentially private training; Batch Normalisation, by taking averages across the batch during the forward pass, leaks information over samples in a batch and precludes the computation of *per-sample* gradients necessary for DP-SGD. More details about the model architectures can be found in the supplementary material, available online.

When training graph classification models with DP-SGD, we follow the standard procedure of DP-SGD training. Firstly, a privacy budget is set in terms of ε, then the model is trained with a specific noise multiplier that defines the amount of Gaussian noise added to the gradients of the model and a L_2-sensitivity bound. The model can then be trained a certain number of iterations, until the privacy budget ε is reached. We then report the scores of the best-performing model out of the ones trained before the privacy budget is exhausted. For all differentially private training runs, we set $\delta = \frac{1}{N}$, where N is the cardinality of the dataset and monitor the performance of the algorithm with different privacy budgets ε. Across all experiments, we utilise the same model architectures for DP-SGD and SGD training with the removal of potential dropout layers for DP-SGD training. In Table II we report the mean performance as well as the standard

TABLE II
SUMMARY OF OUR EXPERIMENTAL EVALUATION ON FOUR DATASETS: *SYNTHETIC*, *FINGERPRINTS*, *ECG*, *MOLBACE*, AND *ORGAN MESHES* WITH DIFFERENT NETWORK TYPES

Data	Network	Training	ROC-AUC	Accuracy	Sensitivity	Specificity	F1-Score	Noise	L_2-Clip	ε
Synthetic	GCN	SGD	0.934 ± 0.01	0.934 ± 0.01	0.955 ± 0.03	0.913 ± 0.03	0.934 ± 0.01	-	-	-
		DP-SGD	0.918 ± 0.02	0.918 ± 0.02	0.897 ± 0.03	0.940 ± 0.02	0.917 ± 0.02	1.0	3.0	5.0
		DP-SGD	0.907 ± 0.02	0.910 ± 0.20	0.869 ± 0.04	0.946 ± 0.20	0.907 ± 0.02	2.0	3.0	1.0
		DP-SGD	0.757 ± 0.11	0.756 ± 0.10	0.936 ± 0.06	0.575 ± 0.28	0.756 ± 0.10	2.2	3.0	0.5
	GAT	SGD	0.912 ± 0.01	0.912 ± 0.01	0.940 ± 0.03	0.883 ± 0.06	0.912 ± 0.01	-	-	-
		DP-SGD	0.893 ± 0.01	0.893 ± 0.01	0.895 ± 0.03	0.891 ± 0.03	0.893 ± 0.01	1.0	3.0	5.0
		DP-SGD	0.872 ± 0.02	0.872 ± 0.02	0.827 ± 0.04	0.907 ± 0.07	0.872 ± 0.02	2.0	3.0	1.0
		DP-SGD	0.575 ± 0.07	0.575 ± 0.07	0.730 ± 0.35	0.419 ± 0.42	0.576 ± 0.08	2.2	3.0	0.5
	SAGE	SGD	0.903 ± 0.02	0.902 ± 0.02	0.913 ± 0.04	0.893 ± 0.08	0.903 ± 0.02	-	-	-
		DP-SGD	0.918 ± 0.01	0.918 ± 0.01	0.907 ± 0.03	0.933 ± 0.02	0.918 ± 0.01	1.0	3.0	5.0
		DP-SGD	0.893 ± 0.01	0.892 ± 0.01	0.872 ± 0.04	0.914 ± 0.03	0.893 ± 0.01	2.0	3.0	1.0
		DP-SGD	0.598 ± 0.10	0.598 ± 0.11	0.609 ± 0.47	0.587 ± 0.39	0.598 ± 0.10	2.2	3.0	0.5
Fingerprints	GCN	SGD	0.856 ± 0.01	0.785 ± 0.01	0.785 ± 0.01	0.928 ± 0.00	0.785 ± 0.01	-	-	-
		DP-SGD	0.863 ± 0.01	0.794 ± 0.01	0.771 ± 0.01	0.932 ± 0.00	0.794 ± 0.01	1.0	3.0	5.0
		DP-SGD	0.844 ± 0.02	0.766 ± 0.04	0.733 ± 0.05	0.921 ± 0.01	0.766 ± 0.04	1.8	3.0	1.0
		DP-SGD	0.796 ± 0.06	0.693 ± 0.10	0.658 ± 0.09	0.898 ± 0.03	0.693 ± 0.09	2.3	3.0	0.5
	GAT	SGD	0.857 ± 0.01	0.786 ± 0.01	0.764 ± 0.02	0.929 ± 0.01	0.786 ± 0.01	-	-	-
		DP-SGD	0.849 ± 0.02	0.774 ± 0.03	0.733 ± 0.04	0.924 ± 0.01	0.770 ± 0.03	1.0	3.0	5.0
		DP-SGD	0.812 ± 0.02	0.728 ± 0.03	0.661 ± 0.01	0.906 ± 0.01	0.730 ± 0.03	1.8	3.0	1.0
		DP-SGD	0.737 ± 0.05	0.605 ± 0.08	0.585 ± 0.08	0.871 ± 0.03	0.610 ± 0.08	2.3	3.0	0.5
	SAGE	SGD	0.876 ± 0.02	0.814 ± 0.02	0.802 ± 0.03	0.940 ± 0.01	0.814 ± 0.02	-	-	-
		DP-SGD	0.869 ± 0.01	0.804 ± 0.01	0.788 ± 0.02	0.935 ± 0.01	0.804 ± 0.01	1.0	3.0	5
		DP-SGD	0.861 ± 0.01	0.792 ± 0.01	0.776 ± 0.01	0.932 ± 0.00	0.791 ± 0.01	1.8	3.0	1
		DP-SGD	0.712 ± 0.06	0.568 ± 0.08	0.529 ± 0.09	0.853 ± 0.03	0.568 ± 0.08	2.3	3.0	0.5
ECG	GCN	SGD	0.979 ± 0.01	0.932 ± 0.01	0.744 ± 0.03	0.979 ± 0.01	0.845 ± 0.02	-	-	-
		DP-SGD	0.983 ± 0.01	0.904 ± 0.02	0.581 ± 0.07	0.983 ± 0.01	0.727 ± 0.06	0.6	5.0	10
		DP-SGD	0.983 ± 0.01	0.923 ± 0.01	0.644 ± 0.12	0.983 ± 0.01	0.772 ± 0.09	0.8	5.0	5.0
		DP-SGD	0.986 ± 0.02	0.824 ± 0.03	0.169 ± 0.23	0.986 ± 0.02	0.231 ± 0.28	1.5	5.0	1.0
	GAT	SGD	0.983 ± 0.01	0.922 ± 0.04	0.675 ± 0.19	0.983 ± 0.01	0.781 ± 0.17	-	-	-
		DP-SGD	0.968 ± 0.03	0.899 ± 0.01	0.637 ± 0.11	0.968 ± 0.03	0.762 ± 0.11	0.6	5.0	10
		DP-SGD	0.960 ± 0.01	0.909 ± 0.02	0.712 ± 0.12	0.960 ± 0.01	0.811 ± 0.08	0.8	5.0	5.0
		DP-SGD	0.991 ± 0.01	0.846 ± 0.01	0.200 ± 0.11	0.991 ± 0.01	0.319 ± 0.11	1.5	5.0	1.0
	SAGE	SGD	0.985 ± 0.01	0.946 ± 0.01	0.757 ± 0.04	0.985 ± 0.01	0.856 ± 0.02	-	-	-
		DP-SGD	0.972 ± 0.01	0.932 ± 0.02	0.767 ± 0.09	0.972 ± 0.01	0.854 ± 0.06	0.6	5.0	10
		DP-SGD	0.973 ± 0.02	0.928 ± 0.02	0.738 ± 0.09	0.973 ± 0.02	0.835 ± 0.06	0.8	5.0	5.0
		DP-SGD	0.951 ± 0.07	0.841 ± 0.02	0.402 ± 0.30	0.951 ± 0.07	0.493 ± 0.24	1.5	5.0	1.0
Molbace	GCN	SGD	0.743 ± 0.00	0.655 ± 0.02	0.511 ± 0.03	0.820 ± 0.01	0.629 ± 0.02	-	-	-
		DP-SGD	0.699 ± 0.01	0.670 ± 0.01	0.723 ± 0.02	0.608 ± 0.01	0.660 ± 0.01	0.5	5.0	20
		DP-SGD	0.688 ± 0.01	0.609 ± 0.01	0.412 ± 0.01	0.834 ± 0.01	0.552 ± 0.01	0.6	5.0	10
	GAT	SGD	0.781 ± 0.01	0.726 ± 0.02	0.691 ± 0.07	0.766 ± 0.06	0.721 ± 0.02	-	-	-
		DP-SGD	0.747 ± 0.02	0.580 ± 0.02	0.333 ± 0.07	0.862 ± 0.03	0.475 ± 0.07	0.5	5.0	20
		DP-SGD	0.692 ± 0.03	0.518 ± 0.04	0.153 ± 0.10	0.935 ± 0.04	0.248 ± 0.14	0.6	5.0	10
	SAGE	SGD	0.785 ± 0.00	0.654 ± 0.01	0.484 ± 0.02	0.848 ± 0.01	0.616 ± 0.01	-	-	-
		DP-SGD	0.717 ± 0.00	0.620 ± 0.01	0.901 ± 0.00	0.299 ± 0.01	0.448 ± 0.02	0.5	5.0	20
		DP-SGD	0.701 ± 0.00	0.550 ± 0.01	0.262 ± 0.00	0.879 ± 0.01	0.403 ± 0.01	0.6	5.0	10
Organ Meshes	GCN	SGD	0.997 ± 0.00	0.988 ± 0.00	0.988 ± 0.00	0.997 ± 0.00	0.988 ± 0.00	-	-	-
		DP-SGD	0.946 ± 0.00	0.940 ± 0.00	0.940 ± 0.00	0.985 ± 0.00	0.940 ± 0.00	0.791	2.0	1.0
		DP-SGD	0.946 ± 0.00	0.934 ± 0.00	0.934 ± 0.00	0.984 ± 0.00	0.934 ± 0.00	1.07	1.5	0.5
	Cheb	SGD	0.992 ± 0.00	0.983 ± 0.00	0.983 ± 0.00	0.996 ± 0.00	0.983 ± 0.00	-	-	-
		DP-SGD	0.978 ± 0.00	0.933 ± 0.00	0.933 ± 0.00	0.983 ± 0.00	0.933 ± 0.00	0.796	2.5	1.0
		DP-SGD	0.982 ± 0.00	0.925 ± 0.00	0.924 ± 0.00	0.981 ± 0.00	0.925 ± 0.00	1.094	2.5	0.5
	SAGE	SGD	0.996 ± 0.00	0.992 ± 0.00	0.991 ± 0.00	0.998 ± 0.00	0.991 ± 0.00	-	-	-
		DP-SGD	0.981 ± 0.00	0.938 ± 0.00	0.937 ± 0.00	0.984 ± 0.00	0.938 ± 0.00	0.796	1.0	1.0
		DP-SGD	0.988 ± 0.00	0.937 ± 0.00	0.937 ± 0.00	0.984 ± 0.00	0.937 ± 0.00	1.06	2.5	0.5

We report results with SGD and DP-SGD training as well as varying privacy budgets ε. The scores are evaluated on the test sets with a standard deviation based on five independent runs. We find that our models achieve high performance when using our proposed DP-SGD training method. The performance decreases gradually when increasing privacy guarantees.

deviation of five independent runs for each experiment. We evaluate different scores for each model: ROC AUC, Accuracy, Sensitivity, Specificity and F1 Score. Hereby sensitivity reports the rate between the true positives and the sum of the true positives and false negatives. Specificity is the rate between the true negatives and the sum of the true negatives and false positives. The ROC AUC score is the Compute Area Under the Receiver Operating Characteristic Curve with a *micro* average for multi-class datasets. Accuracy is the rate between the true positives and all samples and the F1 Score reports the harmonic mean of the precision and recall, also using a *micro* averaging strategy for multi-class datasets.

V. Experimental Results

In this section, we evaluate our results, compare DP-SGD training with standard SGD training and show the impact of different privacy budgets on model performances. The results achieved on the four datasets are summarised in Table II.

Summary of Results: For all datasets, we observe similar behaviour, namely a correlation between stronger privacy budgets and diminished model performance. Although this phenomenon is – in general – an unavoidable, information-theoretic consequence of the trade-off between privacy and utility, the individual models exhibit different behaviour with regards to their individual tolerances towards the amount of Gaussian noise added for DP-SGD, as well as the tolerances towards gradient clipping. For instance, for the *synthetic* dataset, an ε value of 5 does not lead to accuracy loss, whereas for the *Molbace* dataset, a privacy budget of $\varepsilon = 10$ already results in diminished model accuracy. Interestingly, the performance of DP-SGD training is overall not substantially influenced by the choice of GNN architecture (GCN, GAT, GraphSAGE, or ChebNet). We observe high performance and similar convergence rates for all architectures, indicating the robust performance of DP-SGD training. For a comparison of the training behaviours please see our Figure in the supplementary material, available online.

For all models, we observe an increased inter-run variability with stronger privacy guarantees. This behaviour is reflected in the higher standard deviations reported in Table II, and we attribute this phenomenon to the increased randomness injected by the DP mechanism.

Exemplarily, we visualise the impact of a stronger privacy guarantee on the performance on the *ECG* dataset in Fig. 4. Given that the dataset is highly imbalanced, a constant prediction (marked by the lower dashed green line in Fig. 4) would result in an approximate test accuracy of 81.6%. We examine the dependency of the results on the choice of ε and report the different performances. With a very strong privacy guarantee (corresponding to a low ε value), the performance of the network is barely better than a constant prediction. The looser the privacy guarantee (larger ε value) the better the performance; for a very loose ε the results reach non-DP performance. Interestingly, for some models we observe identical performance between DP-SGD and normal training, e.g. *Fingerprint-GCN*, where the DP-SGD model (privacy budget of $\varepsilon = 5$) reaches slightly higher performance then the normal training, see Table II; this

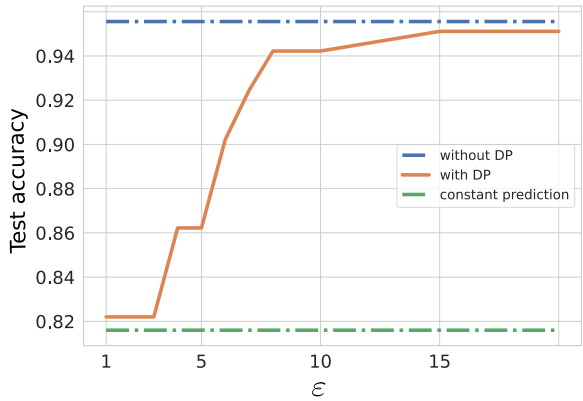

Fig. 4. Impact of ε on test accuracy on ECG dataset. The performance increases with larger ε values and looser privacy guarantees. The top dashed line (blue) indicates the performance without DP, the lower dashed line (green) a constant prediction and the solid line in the middle (orange) the model performance with different ε values: $\varepsilon \in \{1, 2, \ldots, 10, 15, 20\}$.

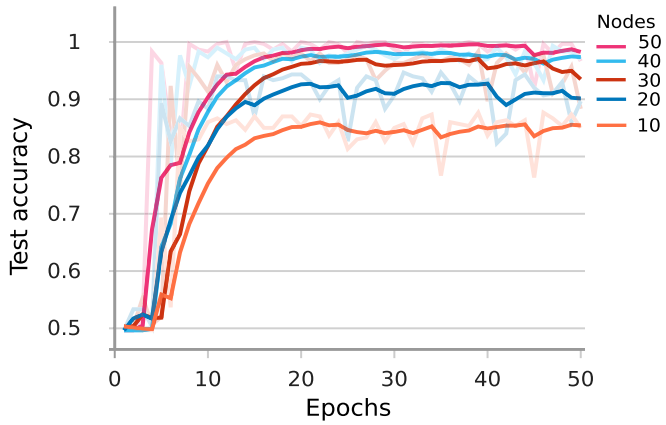

Fig. 5. Impact of graph size to performance under DP: Increasing graph sizes result in better performance and faster convergence. The privacy guarantees are set to $\varepsilon = 2.3$.

beneficial effect can be attributed to the regularising effects of gradient norm bounding and noise injection, indicating that – within certain constraints – DP training can go hand-in-hand with excellent overall model performance and generalisability.

Scalability: In order to investigate the scalability of our approach, we vary the size of the created Erdős-Rényi graphs in the *synthetic* dataset between 10 and 500 nodes per graph. Fig. 5 shows the impact of the graph size on the performance under DP using a three-layer GCN and $\varepsilon = 2.3$. We visualised the performances of graph sizes between 10 and 50 nodes and find that performance improves with increasing graph size in these ranges. Beyond 50 nodes, the performance remains consistently high, which is why these plots were not included in Fig. 5. This behaviour indicates a strong performance of our model across varying graph sizes, i.e. robust scalability. Furthermore, with the utilisation of the large organ mesh dataset, we could show that our method also performs excellently for graphs with a large number of nodes and edges as well as large datasets with more than 100,000 graphs. In this dataset, we observe

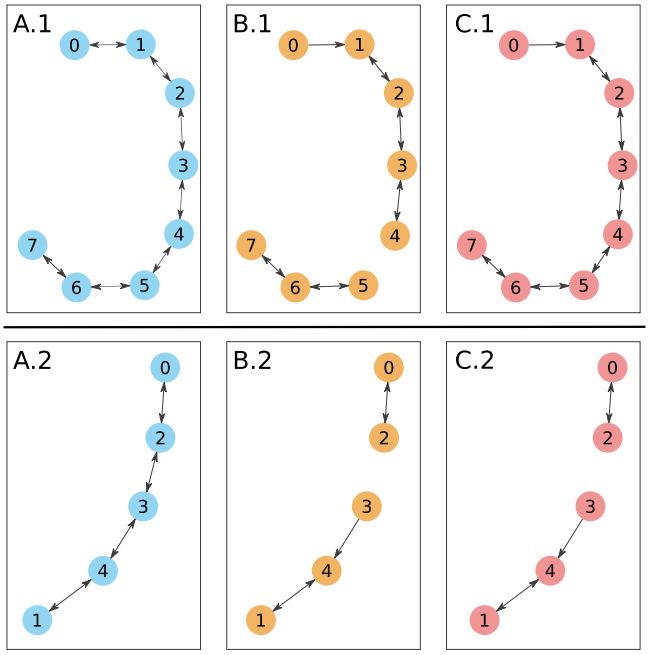

Fig. 6. Visualisation of two GNNExplainer examples. The original graph (**A**) is shown in blue, the resulting graph from the GNNExplainer and the model trained with SGD in orange (**B**) and with DP-SGD in red (**C**). In the example in the upper row (**1**) the two graphs (**B**) and (**C**) differ slightly, whereas in the lower example (**2**) both GNNExplainer graphs (**B**) and (**C**) are equal, meaning that the two models consider the same edges to be relevant. The privacy budget for the models trained with DP-SGD was set to ε=5.

low utility loss in the range of 10^{-3} even in a very high privacy regime of $\varepsilon = 0.5$. In comparison, many deep learning networks require a more loose privacy guarantee to achieve high performance [59].

Explainability: The interpretability of GNNs is a challenging and frequently discussed task in research. Recently, approaches like the GNNExplainer [60] formalised methods which can be used to interpret the results of trained GNNs. We make use of this method to interpret the differences in learned representations between models trained with DP-SGD and non-private SGD and visualise the results in Fig. 6. The GNNExplainer is an approach for post-hoc interpretation of predictions generated by a trained GNN. It is used to identify which edges in the graph represent essential connections for a prediction of the network, thus indicating nodes important for the final prediction. GNNExplainer prunes the original graph to only contain the nodes and edges with the highest impact on the model prediction. We apply the GNNExplainer to our results on the *Fingerprints* dataset, comparing a GCN model trained with standard SGD and three GCN models trained with DP-SGD with $\varepsilon = 5$, $\varepsilon = 1$ and $\varepsilon = 0.5$. We set the GNNExplainer threshold for edge importance to 0.2. Qualitatively, we observe that the GNNExplainer results of the DP models and the standard models appear very similar, if not identical for some examples, see Fig. 6 and supplementary material, available online. In these Figures, (**A**) visualises an example of an original graph from the *Fingerprints* dataset, containing all edges. Figures (**B**) and (**C**) show the pruned graphs for SGD and DP-SGD training,

TABLE III
MEAN IOU SCORES OF TEN TEST SAMPLES FROM THE *FINGERPRINT* DATASET FOR COMPARING EDGES BETWEEN (||) THE ORIGINAL GRAPH, THE GNNEXPLAINER GRAPH OF THE MODEL TRAINED WITH SGD, AND THE GNNEXPLAINER GRAPH OF THE MODEL TRAINED WITH DP-SGD. THE IOU BETWEEN THE ORIGINAL GRAPH AND THE NON-DP GRAPH IS **0.739**

ε	ROC-AUC	IoU (orig. || DP)	IoU (DP || non-DP)
5.0	0.863	0.652	**0.765**
1.0	0.844	0.592	**0.736**
0.5	0.796	0.617	**0.609**

The IoU between the original graph and the non-DP graph is 0.739. The IoU between the DP and the non-DP graphs decreases with a smaller ε value which corresponds to smaller ROC AUC results.

respectively. In the lower example (**2**) in Fig. 6, both GNNExplainer graphs are identical (almost identical in the upper row), showing that in both models the same edges and nodes have a high impact on the models' predictions. This indicates that the feature importance is the same (or almost the same) between both models and that the feature importance is not compromised by the privacy guarantees achieved through DP training.

To provide a quantitative estimation of GNNExplainer similarity of our results, we propose and use an Intersection over Union (IoU) score, measuring the pair-wise overlap of edges in the three resulting graphs. The IoU score of two graphs A and B is defined as follows:

$$\frac{|\mathcal{E}_A \cap \mathcal{E}_B|}{|\mathcal{E}_A \cup \mathcal{E}_B|}, \tag{4}$$

where $\mathcal{E}_X$ represents the set of all edges in Graph X and $|\cdot|$ denotes the cardinality of a set. Table III summarises the results of the mean IoU values between the original graph and the GNNExplainer graph based on training with DP, and the two resulting GNNExplainer graphs from DP-SGD and SGD training. The IoU score of the original graph and the GNNExplainer graph of the model trained with standard SGD is 0.739 for all graphs. We compare the overlap between the graphs with the model performance, reported by the ROC AUC score. We find a high IoU score for DP vs. non-DP models, which is in line with the GNNExplainer plots we observe in Fig. 6. Moreover, we observe that our GNNExplainer IoU score of the DP and the non-DP models slightly decreases with a smaller ε and smaller ROC AUC scores, see Table III. The increase in the IoU score between the original model and the DP model with $\varepsilon = 0.5$ most likely only indicates that the DP trained model with $\varepsilon = 0.5$ considers more edges as relevant than the model trained with $\varepsilon = 1.0$. These qualitative and quantitative GNNExplainer results indicate that our proposed DP graph classification models exhibit strong and similar inductive biases compared to "normal" GNNs while preserving privacy guarantees.

VI. DISCUSSION, CONCLUSION, AND FUTURE WORK

Our work introduces and evaluates differentially private graph classification, a formal method to offer quantifiable privacy guarantees in applications where sensitive data of individuals is represented as a whole graph. Such contexts include medical data (as shown in our ECG classification example), where DP

can enable training of machine learning models while maintaining both regulatory compliance and adherence to ethical standards mandating the protection of health-related data.

GNN Training is Possible With Strong Privacy Guarantees and Excellent Utility: Our experiments on benchmark and real-world datasets demonstrate that the training of GNNs for graph classification is viable with high utility and tight privacy guarantees. Especially the large scale mesh classification dataset achieved almost perfect accuracy even with very tight privacy bounds of $\varepsilon = 0.5$. Expectedly, we observe a privacy-performance trade-off for all datasets, whereby a decrease in the value of ε results in a decline in the accuracy of the model, as demonstrated in Fig. 4. The amount of performance loss is task and dataset dependent.

GNNs Learn Similar Features in the Private and Non-Private Scenarios: Additionally, we investigate the utilisation of explainability techniques to compare the representations learned by models trained with SGD and DP-SGD. The application of the GNNExplainer indicates that models trained with DP-SGD learn similar relevant representations to the non-privately trained models. To quantitatively demonstrate the results of the GNNExplainer, we calculated an IoU score on the edges considered important by the technique between the resulting graphs. We observe an overall high IoU with a slight decline in overlap with tighter privacy guarantees, indicating that – as expected – the high levels of noise required to achieve such guarantees eventually become detrimental to learning.

Private GNN Training Can Help Alleviate Social Impacts of Machine Learning: We strongly believe that the implementation of formal techniques for privacy preservation like DP in the setting of GNN training will mitigate the risks of using sensitive data in ML tasks. In the case of medical data (as in the *ECG* dataset example), we believe the utilisation of privacy preserving methods to also hold positive effects in terms of encouraging data owners (such as patients) to make their data accessible for research purposes. Evidently, such implementations must go hand in hand with educating potential stakeholders in the correct application of DP mechanisms, including the appropriate choice of parameters like ε. In this work, we rely exclusively on public datasets collected with informed consent or with approval of institutional review boards wherever applicable.

Limitations: Inherent to the concept of differential privacy in machine learning is a performance-to-privacy trade-off. While our experiments visually illustrate the implications of the trade-off and provide insight into its practical importance in the context of machine learning on graphs, the actual relationship between privacy and accuracy is highly task- and user-specific [61], [62]. Therefore, we note that one can interpret the value of ε as an additional design-parameter that needs to be optimised for in order to minimise the adverse effects that DP can have on performance in the context of graph classification (or most other learning tasks in general).

Future Work: In our experiments we utilise a limited set of standard model architectures (GCN, GraphSAGE, GAT, ChebNet). Evidently, more sophisticated architectures have been designed and deployed to real world problems. As our proposed approach is general, we assume that an extension to such advanced graph learning models is natural and should exhibit similar behaviour, and we intend to expand our purview to such models in future investigations.

While the GNNExplainer concept can provide initial clues to interpret and explain GNN training and the intrinsic differences between models trained with SGD and DP-SGD, it is only an initial step towards full explainability and interpretability. We consider this to be a highly relevant and an interesting direction for future research. In particular, we aim to investigate the effects of differentially private GNN learning on adversarial robustness of the model. We hypothesise that – similarly to Euclidean settings – [63], [64] DP should have a mitigating effect against attacks that diminish the utility of the trained model in the context of machine learning on graphs. Furthermore, we believe that a comparison of different explainability techniques like [65], [66], [67], [68] will provide even more insight into the differences between DP and non-DP training, which we also intend to investigate in future work.

References

[1] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2005, pp. 729–734.

[2] M. Michael, J. Bronstein, Y. Bruna, A. LeCunSzlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.

[3] W. Fan et al., "Graph neural networks for social recommendation," in *Proc. World Wide Web Conf.*, 2019, pp. 417–426.

[4] X. Li et al., "Graph neural network for interpreting task-fMRI biomarkers," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Intervention*, 2019, pp. 485–493.

[5] C. Mao, L. Yao, and Y. Luo, "MedGCN: Graph convolutional networks for multiple medical tasks," 2019, *arXiv:1904.00326*.

[6] D. Duvenaud et al., "Convolutional networks on graphs for learning molecular fingerprints," 2015, *arXiv:1509.09292*.

[7] J. C. Paetzold et al., "Whole brain vessel graphs: A dataset and benchmark for graph learning and neuroscience," in *Proc. 35th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track*, 2021. [Online]. Available: https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/c9f0f895fb98ab9159f51fd0297e236d-Abstract-round2.html

[8] X. Wei, R. Yu, and J. Sun, "View-GCN: View-based graph convolutional network for 3D shape analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1850–1859.

[9] X. Li et al., "BrainGNN: Interpretable brain graph neural network for fMRI analysis," *Med. Image Anal.*, vol. 74, 2021, Art. no. 102233.

[10] K. Ganju, Q. Wang, W. Yang, Carl A. Gunter, and N. Borisov, "Property inference attacks on fully connected neural networks using permutation invariant representations," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Toronto Canada, 2018, pp. 619–633.

[11] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 3–18.

[12] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," 2019, *arXiv:1911.07135*.

[13] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting Gradients–How easy is it to break privacy in federated learning?," 2020, *arXiv:2003.14053*.

[14] Z. Zhang et al., "Extracting private graph data from graph neural networks," 2021, *arXiv:2106.02820*.

[15] X. He, R. Wen, Y. Wu, M. Backes, Y. Shen, and Y. Zhang, "Node-level membership inference attacks against graph neural networks," 2021, *arXiv:2102.05429*.

[16] E. Iyiola, W. O. Nejdl, and M. Khosla, "Membership inference attack on graph neural networks," 2021, *arXiv:2101.06570*.

[17] C. Dwork et al., "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3/4, pp. 211–407, 2014.

[18] A. Ziller, D. Usynin, R. Braren, M. Makowski, D. Rueckert, and G. Kaissis, "Medical imaging deep learning with differential privacy," *Sci. Rep.*, vol. 11, no. 1, pp. 1–8, 2021.

[19] G. Kaissis et al., "End-to-end privacy preserving deep learning on multi-institutional medical imaging," *Nature Mach. Intell.*, vol. 3, no. 6, pp. 473–484, 2021.

[20] P. Basu, T.S. Roy, R. Naidu, Z. Muftuoglu, S. Singh, and F. Mireshghallah, "Benchmarking differential privacy and federated learning for BERT models," 2021, *arXiv:2106.13973*.

[21] H.H. Zhuo, W.Y. Feng, Q. LinXu, and Q. Yang, "Federated deep reinforcement learning," 2019, *arXiv:1901.08277*.

[22] L. Frigerio, A. S. de Oliveira, L. Gomez, and P. Duverger, "Differentially private generative adversarial networks for time series, continuous, and discrete open data," in *Proc. IFIP Int. Conf. ICT Syst. Secur. Privacy Protection*, 2019, pp. 151–164.

[23] M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 308–318.

[24] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *Proc. 39th Annu. ACM Symp. Theory Comput.*, 2007, pp. 75–84.

[25] S. Raskhodnikova and A. Smith, "Differentially private analysis of graphs. Encyclopedia of Algorithms," 2016.

[26] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith, "Analyzing graphs with node differential privacy," in *Proc. Theory Cryptogr. Conf.*, 2013, pp. 457–476.

[27] T. Igamberdiev and I. Habernal, "Privacy-preserving graph convolutional networks for text classification," *arXiv:2102.09604*, 2021.

[28] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv:1609.02907*, 2016.

[29] A. Daigavane, G. Madan, A. Sinha, A. G. Thakurta, G. Aggarwal, and P. Jain, "Node-level differentially private graph neural networks," 2021, *arXiv:2111.15521*.

[30] J. Zhou et al., "Vertically federated graph neural network for privacy-preserving node classification," 2020, *arXiv:2005.11903*.

[31] S. Sajadmanesh and D. Gatica-Perez, "Locally private graph neural networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2021, pp. 2130–2145.

[32] P. Liao et al., "Information obfuscation of graph neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 6600–6610 .

[33] X. Han, L. Wang, J. Wu, and Y. Yang, "Large-scale privacy-preserving network embedding against private link inference attacks," 2022, *arXiv:2205.14440*.

[34] H. Hu, L. Cheng, J. P. Vap, and M. Borowczak, "Learning privacy-preserving graph convolutional network with partially observed sensitive attributes," in *Proc. ACM Web Conf.*, 2022, pp. 3552–3561.

[35] I. Mironov, "Rényi differential privacy," in *Proc. IEEE 30th Comput. Secur. Foundations Symp.*, 2017, pp. 263–275.

[36] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," 2016, *arXiv:1610.05755*.

[37] V. Karwa, S. Raskhodnikova, A. Smith, and G. Yaroslavtsev, "Private analysis of graph structure,' *Proc. VLDB Endowment*, vol. 4, no. 11, pp. 1146–1157, 2011.

[38] T. T. Mueller, D. Usynin, J. C. Paetzold, D. Rueckert, and G. Kaissis, "SoK: Differential privacy on graph-structured data," 2022, *arXiv:2203.09205*.

[39] S. E. Petersen et al., "Imaging in population science: Cardiovascular magnetic resonance in 100,000 participants of UK biobank-rationale, challenges and approaches," *J. Cardiovasc. Magn. Reson.*, vol. 15, no. 1, pp. 1–10, 2013.

[40] F. Galton, *Finger Prints*. London, U.K.: Macmillan, 1892.

[41] ER Henry, *Classification and Uses of Fingerprints London*. Atlantic City, NJ, USA: George Rutledge and Sons, Limited, 1900.

[42] M. Neuhaus and H. Bunke, "A graph matching based approach to fingerprint classification using directional variance," in *Proc. Int. Conf. Audio-and Video-Based Biometric Person Authentication*, 2005, pp. 191–200 .

[43] K. Riesen and H. Bunke, "Iam graph database repository for graph based pattern recognition and machine learning," in *Proc. Joint IAPR Int. Workshops Statist. Techn. Pattern Recognit. Struct. Syntactic Pattern Recognit.*, 2008, pp. 287–297.

[44] C. Morris, Nils M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, "TUDataset: A collection of benchmark datasets for learning with graphs," in *Proc. Workshop Graph Representation Learn. Beyond*, 2020. [Online]. Available: https://github.com/chrsmrrs/tudataset

[45] W. Hu et al., "Open graph benchmark: Datasets for machine learning on graphs," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 22118–22133, 2020.

[46] Z. Wu et al., "MoleculeNet: A benchmark for molecular machine learning," *Chem. Sci.*, vol. 9, no. 2, pp. 513–530, 2018.

[47] F. Liu et al., "An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection," *J. Med. Imag. Health Informat.*, vol. 8, no. 7, pp. 1368–1373, 2018.

[48] T. Kart et al., "Deep learning-based automated abdominal organ segmentation in the UK biobank and German national cohort magnetic resonance imaging studies," *Invest. Radiol.*, vol. 56, no. 6, pp. 401–408, 2021.

[49] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *ACM Siggraph Comput. Graph.*, vol. 21, no. 4, pp. 163–169, 1987.

[50] S. V. der Walt et al., "scikit-image: Image processing in Python," *PeerJ*, vol. 2, 2014, Art. no. e453.

[51] J. Klicpera, A. Bojchevski, and S. Günnemann, "Predict then propagate: Graph neural networks meet personalized pagerank," 2018, *arXiv:1810.05997*.

[52] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh, "Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 257–266.

[53] K. Kong et al., "FLAG: Adversarial data augmentation for graph neural networks," 2020.

[54] Q. Huang, H. He, A. Singh, S.-N. Lim, and A. R. Benson, "Combining label propagation and simple models out-performs graph neural networks," 2020, *arXiv:2010.13993*.

[55] J. Klicpera, J. Groß, and S. Günnemann, "Directional message passing for molecular graphs," in *Proc. Int. Conf. Learn. Representations*, 2019. [Online]. Available: https://iclr.cc/virtual_2020/poster_B1eWbxStPH.html

[56] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.

[57] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[58] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[59] S. De, L. Berrada, J. Hayes, S. L. Smith, and B. Balle, "Unlocking high-accuracy differentially private image classification through scale," 2022, *arXiv:2204.13650*.

[60] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "GNN explainer: A tool for post-hoc explanation of graph neural networks," 2019, *arXiv:1903.03894*.

[61] C. Dwork, N. Kohli, and D. Mulligan, "Differential privacy in practice: Expose your epsilons!," *J. Privacy Confidentiality*, vol. 9, no. 2, 2019. [Online]. Available: https://journalprivacyconfidentiality.org/index.php/jpc/article/view/689

[62] R. Cummings, G. Kaptchuk, and E. M. Redmiles ., ""I need a better description": An investigation into user expectations for differential privacy," 2021, *arXiv:2110.06452*.

[63] M. Naseri, J. Hayes, and D. Cristofaro, "Toward robustness and privacy in federated learning: Experimenting with local and central differential privacy," 2020, *arXiv:2009.03561*.

[64] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," in *Proc. IEEE Symp. Secur. Privacy*, 2019, pp. 656–672.

[65] F. Baldassarre and H. Azizpour, "Explainability techniques for graph convolutional networks," 2019, *arXiv:1905.13686*.

[66] L. Faber, K. Moghaddam, and R. A. Wattenhofer, "Contrastive graph neural network explanation," 2020, *arXiv:2010.13663*.

[67] D. Luo et al., "Parameterized explainer for graph neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 19620–19631.

[68] Q. Huang, M. Yamada, Y. Tian, D. Singh, D. Yin, and Y. Chang, "Graphlime: Local interpretable model explanations for graph neural networks," 2020, *arXiv:2001.06216*.

[69] A. Paszke et al., "Automatic differentiation in pytorch," 2017.

[70] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch geometric," in *Proc. ICLR Workshop Representation Learn. Graphs Manifolds*, 2019.

[71] R. Zou and H. He, "functorch: JAX-like composable function transforms for pytorch," 2021. [Online]. Available: https://github.com/pytorch/functorch
[72] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. D. Alché-Buc, E. Fox, and R. Garnett, Eds., Red Hook, NY, USA: Curran Associates, Inc., 2019, pp. 8024–8035.
[73] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "GNNExplainer: Generating explanations for graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, Art. no. 9240.

Tamara T. Mueller received the MPhil degree in advanced computer science from the University of Cambridge, in 2018. She is currently working toward the PhD degree with the chair for AI in Medicine and Healthcare, Technical University of Munich. Her main research interests include the application of geometric deep learning to medical imaging tasks and differential privacy of deep learning models - as well as the intersection of both.

Johannes C. Paetzold is a postdoctoral researcher in computer science with Imperial College London. He is also the artificial intelligence team leader with the Institute for Tissue Engineering and Regenerative Medicine, Helmholtz Zentrum München. His main interests include development of deep learning and graph learning methods for large biological networks such as vessels and neurons. Further research interests include topology-aware machine learning and generative models.

Chinmay Prabhakar is currently working toward the PhD degree with the University of Zurich. His main research interest includes development of deep learning and graph learning methods on biological networks. Further research interest includes application of multi-modal learning specifically text and image data in medical domain.

Dmitrii Usynin currently working toward the PhD degree with a Joint Academy of Doctoral Studies (JADS) launched between Imperial College London and Technical University of Munich. His research interests include the domain of adversarial influence in collaborative machine learning, privacy-preserving machine learning and trustworthy artificial intelligence. He is also a privacy researcher with OpenMined, working on federated learning and differential privacy in healthcare. Dmitrii graduated from Imperial College London with an MEng in Computing and a distinguished project titled "Privacy-Preserving Machine Learning in a Medical Domain".

Daniel Rueckert (Fellow, IEEE) received the PhD degree from Imperial College, in 1997. He is Alexander von Humboldt professor for AI in Medicine and Healthcare with the Technical University of Munich. He is also professor of Visual Information Processing with the Department of Computing, Imperial College London. He has published more than 500 journal and conference articles in the area of medical image computing. He served as associate editor of *IEEE Transactions on Medical Imaging* and is a member of the editorial board of Medical Image Analysis. In 2014, he has been elected as a fellow with the MICCAI society, and in 2015 he was elected as a fellow with the Royal Academy of Engineering. More recently he has been elected as fellow with the Academy of Medical Sciences (2019) and as fellow with the American Institute for Medical and Biological Engineering (2021).

Georgios Kaissis received the medical and doctoral degrees from LMU Munich. He is an adjunct assistant professor and attending radiologist with the Technical University of Munich, where he leads the Trustworthy and Privacy-Preserving Artificial Intelligence group with the Institute of Artificial Intelligence in Medicine. He also leads the Reliable Artificial Intelligence group with the Institute for Machine Learning in Biomedical Imaging, Helmholtz Zentrum Munich. He master's in business administration for healthcare with FAU Nuremberg and conducted a postdoc with the Department of Computing, Imperial College London, where he remains as an honorary research associate. His research interests include focuses on next-generation privacy-preserving machine learning techniques and especially differential privacy and its applications to medical deep learning.