Interpretable and Differentially Private Predictions

Frederik Harder[*,†], **Matthias Bauer**[*,‡], **Mijung Park**[*,†]
[*]Max Planck Institute for Intelligent Systems, Tübingen, Germany
[†]University of Tübingen, Tübingen, Germany
[‡]Department of Engineering, University of Cambridge, Cambridge, UK
{fharder|bauer|mpark}@tue.mpg.de

Abstract

Interpretable predictions, where it is clear why a machine learning model has made a particular decision, can compromise privacy by revealing the characteristics of individual data points. This raises the central question addressed in this paper: *Can models be interpretable without compromising privacy?* For complex "big" data fit by correspondingly rich models, balancing privacy and explainability is particularly challenging, such that this question has remained largely unexplored. In this paper, we propose a family of simple models in the aim of approximating complex models using several *locally linear maps* per class to provide high classification accuracy, as well as differentially private explanations on the classification. We illustrate the usefulness of our approach on several image benchmark datasets as well as a medical dataset.

1 Introduction

The *General Data Protection Regulation (GDPR)* by the European Union imposes two important requirements on algorithmic design, *interpretability* and *privacy* [27]. These requirements introduce new standards on future algorithmic techniques, making them of particular concern to the machine learning community [9]. This paper addresses these two requirements in the context of classification, and studies the trade-off between privacy, accuracy and interpretability, see Fig. 1.

Broadly speaking, there are two options to take for gaining interpretability: (i) rely on *inherently interpretable models*; and (ii) rely on *post-processing schemes* to probe trained complex models. Inherently interpretable models are often relatively simple and their predictions can be easily analyzed in terms of their respective input features. For instance, in logistic regression classifiers and sparse linear models the coefficients represent the importance of each input feature. However, modern "big" data typically exhibit complex patterns, such that these relatively simplistic models often have lower accuracy than more complex ones. To address this trade-off between interpretability and accuracy (Fig. 1 Ⓑ) there are many attempts to use more complex models such as deep neural networks and post-process these models to gain insights [2, 3, 16, 21, 22, 24, 26].

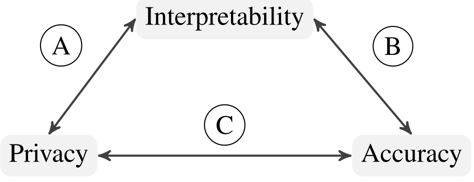

Figure 1: Modern machine learning systems need to trade off accuracy, privacy, and interpretability.

On the other hand, many recent papers address the concern that complex models with outstanding predictive performance can expose sensitive information from the dataset they were trained on [5, 7, 23, 25]. To quantify privacy, many recent approaches adopt the notion of *differential privacy* (DP), which provides a mathematically provable definition of privacy, and can quantify the level of privacy an algorithm or a model provides [6]. In plain English, an algorithm is called

Preprint. Under review.

differentially private, if its output is random enough to obscure the participation of any single individual in the data. The randomness is typically achieved by injecting noise into the algorithm. The amount of noise is determined by the level of privacy the algorithm guarantees and the sensitivity, a maximum difference in its output depending on a single individual's participation or non-participation in the data (See Sec. 2 for a mathematical definition of DP).

There is, however, a natural trade-off between privacy and accuracy (Fig. 1 Ⓒ): while a large amount of added noise provides a high level of privacy but also harms prediction accuracy. When the number of parameters is high, like in deep neural network models, juggling this trade-off is very challenging, as privatizing high dimensional parameters results in a high privacy loss to meet a good prediction level [1]. Therefore, most existing work considers small networks or assumes that some relevant data are publicly available to train a significant part of the network without privacy violation (See Sec. 5 for details).

In this paper, we study the trade-off between interpretability, privacy, and accuracy. Taking into account privacy *and* interpretability (Fig. 1 Ⓐ), we propose a family of inherently interpretable models that do not require post-processing for interpretability and can be trained privately. These models approximate the mapping of a complex model from the input data to class score functions, using *locally linear maps* (LLM) per class. Our formulation for LLM is inspired by the fact that a differentiable function can be well approximated as a collection of piece-wise linear functions, i.e., the first-order Taylor expansion of the function at a sufficiently large number of input locations. Indeed, our local models with a sufficiently large number of linear maps permit a relatively slight loss in accuracy compared to complex model counterparts[1]. In addition, the learned linear maps for each class provide insights on the key features for a classification task at hand.

Our locally linear maps, however, often introduce many linear maps (i.e., many parameters) to reach the classification performance of complex model counterparts. In terms of privacy, the high dimensionality introduces a challenge as mentioned earlier. We adopt the *Johnson-Lindenstrauss transform*, a.k.a., *random projection* [10], to decrease the dimensionality of each locally linear maps to an intermediate level and privatize the resulting lower dimensional quantities. We study the interplay between interpretability, privacy, and random projection in Sec. 4. To the best our knowledge, this is the first attempt to study the trade-off between privacy, accuracy, *and* interpretability. We hope that this work sparks a conversation in the machine learning community for algorithmic designs which consider all three simultaneously.

2 Background on Differential Privacy

We start by providing background information on differential privacy and a composition method that we will use in our algorithm, as well as random projections.

Differential privacy. Consider an algorithm $\mathcal{M}$ and neighbouring datasets $\mathcal{D}$ and $\mathcal{D}'$ differing by a single entry, where the dataset $\mathcal{D}'$ is obtained by excluding one datapoint from the dataset $\mathcal{D}$. In differential privacy [6], the quantity of interest is *privacy loss* , defined by

$$L^{(o)} = \log \frac{\Pr(\mathcal{M}_{(\mathcal{D})} = o)}{\Pr(\mathcal{M}_{(\mathcal{D}')} = o)}, \tag{1}$$

where $\mathcal{M}_{(\mathcal{D})}$ and $\mathcal{M}_{(\mathcal{D}')}$ denote the outputs of the algorithm given $\mathcal{D}$ and $\mathcal{D}'$, respectively. $\Pr(\mathcal{M}_{(\mathcal{D})} = o)$ denotes the probability that $\mathcal{M}$ returns a specific output o. When the two probabilities in Eq. (1) are very similar, even a strong adversary, who knows all the datapoints in $\mathcal{D}$ except for one, could not discern the one datapoint by which $\mathcal{D}$ and $\mathcal{D}'$ differ, based on the output of the algorithm alone. On the other hand, when the probabilities are very different, it would be easy to identify the exclusion of the single datapoint in $\mathcal{D}'$. Hence, the privacy loss quantifies how revealing an algorithm's output is about the single entry's participation to the dataset $\mathcal{D}$. Formally, an algorithm $\mathcal{M}$ is called ϵ-DP if and only if $|L^{(o)}| \leq \epsilon, \forall o$. A weaker version of the above is (ϵ, δ)-DP, if and only if $|L^{(o)}| \leq \epsilon$, with probability at least $1 - \delta$.

A popular way of designing differentially private algorithms is by introducing a noise addition step to the algorithm. The *output perturbation* method achieves a DP output by adding noise to the output

[1] The level of loss in prediction accuracy depends on the complexity of data.

h, where the noise is calibrated to h's *sensitivity*, denoted by S_h. A common form of sensitivity is the L2-sensitivity, which is the maximum difference in terms of L2-norm, under the one datapoint's difference in $\mathcal{D}$ and $\mathcal{D}'$, $S_h = \max_{\mathcal{D},\mathcal{D}',\mathcal{D}\setminus\mathcal{D}'=1} \|h(\mathcal{D}) - h(\mathcal{D})\|_2$. With the sensitivity, we can privatize the output using the *Gaussian mechanism*, which simply adds Gaussian noise of the form: $\tilde{h}(\mathcal{D}) = h(\mathcal{D}) + \mathcal{N}(0, S_h^2\sigma^2\mathbf{I}_p)$, where $\mathcal{N}(0, S_h^2\sigma^2\mathbf{I}_p)$ means the Gaussian distribution with mean 0 and covariance $S_h^2\sigma^2\mathbf{I}_p$. The resulting quantity $\tilde{h}(\mathcal{D})$ is (ϵ, δ)-DP, where $\sigma \geq \sqrt{2\log(1.25/\delta)}/\epsilon$ (see [6, Appendix] for a proof). In this paper, we use the Gaussian mechanism to achieve differentially private locally linear maps.

Properties of differential privacy. There are two important properties of differential privacy. The first one is *post-processing invariance*, which states that applying any *data-independent* mechanism to a differentially private quantity does not alter the privacy level of the resulting quantity. Formally,

Proposition 1 (Proposition 2.1 [6]). *Let a mechanism that maps data where χ is the data universe to an output space, i.e., $\mathcal{M} : \mathbb{N}^{|\chi|} \mapsto \mathcal{R}$ be a randomized algorithm that is (ϵ, δ)-differentially private. Let $f : \mathcal{R} \mapsto \mathcal{R}'$ be an arbitrary, data-independent, randomized mapping. Then $f \circ \mathcal{M} : \mathbb{N}^{|\chi|} \mapsto \mathcal{R}'$ is also (ϵ, δ)- differentially private.*

The second one is *composability*, which states that combining differentially private quantities degrades privacy. For instance, if one computes a statistic given a dataset and adds Gaussian noise, and repeats this routine multiple times, combining these privatized quantities, e.g., the average of these quantities will be quite close to the true statistic. Hence, one needs to increase the noise level to keep the strength of the privacy guarantee after the repeated use of data. Existing composition theorems show, how exactly the privacy parameters ϵ and δ compose, when differentially private subroutines are combined. The most naïve way to compose these parameters is the *linear* composition (Theorem 3.14 in [6]), where the resulting parameter, which is often called *cumulative privacy loss* (cumulative ϵ and δ), are linearly summed up, $\epsilon = \sum_{t=1}^T \epsilon_t$ and $\delta = \sum_{t=1}^T \delta_t$ after the repeated use of data T times.

Recently, [1] proposed the *moments accountant* method, which provides a clever way of combining ϵ and δ such that the resulting total privacy loss is significantly smaller than that by other composition methods. The moments accountant method takes advantage of the fact that when adding Gaussian noise in each training step, the privacy loss in Eq. (1) also follows a Gaussian distribution. Hence, by observing the tail behaviour of the Gaussian random variable, one can obtain a tight moments bound which provides a better utility (i.e., smaller noise yields the same privacy guarantee compared to other composition methods) in the resulting (ϵ, δ) guarantee. See Appendix Sec. A for details.

Random projections in the context of differential privacy Our method involves projecting each input onto a lower-dimensional space using a *Johnson-Lindenstrauss transform* (a.k.a., *random projection*) [10]. We construct the projection matrix $\mathbf{R}$ by drawing each entry from $\mathcal{N}(0, 1/D')$ where D' is the dimension of the projected space. This projection nearly preserves the distances between two points in the data space and in the embedding space, as this projection guarantees low-distortion embeddings. Random projections have previously been used to ensure differential privacy [4]. However, here we only utilize them as a convenient method to reduce input dimension to our learnable linear maps. Since the random filters are data-independent, they do not need to be privatized. Now as we covered all relevant background information, in the next section we introduce our method.

3 Locally Linear Maps (LLM)

Suppose we trained a neural network model on a K-class classification problem, where the network maps a high dimensional input $\mathbf{x} \in \mathbb{R}^D$ to a class score function $\mathbf{s}(\mathbf{x})$, i.e., the pre-activation before the final softmax, where $\mathbf{s}(\mathbf{x})$ is a K-dimensional vector with entries s_k. Denote the mapping $\boldsymbol{\phi} : \mathbf{x} \mapsto \mathbf{s}(\mathbf{x})$ and the parameters of the network by $\boldsymbol{\theta}$. *Can we find the best approximation to the function $\boldsymbol{\phi}$, which presents interpretable features for classification and also guarantees a certain level of privacy?* To answer this question, we propose an approximation by *locally linear maps (LLM)*, inspired by gradient-based attribution methods for deep neural networks [2].

The gradient-based attribution methods assume that there is a set of attributions, at which the gradients of a classifier with respect to the input are maximized, *and* that the gradient information provides interpretability as to why the classifier makes a certain prediction. More specifically, they consider a first order Taylor approximation of $\boldsymbol{\phi}$

$$\boldsymbol{\phi}(\mathbf{x}) \approx \boldsymbol{\phi}(\mathbf{x}_0) + \boldsymbol{\phi}'(\mathbf{x}_0)^\top(\mathbf{x} - \mathbf{x}_0) = \boldsymbol{\phi}'(\mathbf{x}_0)^\top \mathbf{x} + \text{shift term}, \tag{2}$$

where $\boldsymbol{\phi}'(\mathbf{x}_0) = \left[\frac{\partial}{\partial \mathbf{x}} \boldsymbol{\phi}(\mathbf{x})\right]_{\mathbf{x}=\mathbf{x}_0}$, and the shift term is $\boldsymbol{\phi}(\mathbf{x}_0) - \boldsymbol{\phi}'(\mathbf{x}_0)^\top \mathbf{x}_0$. Notice that, (a) the first order approximation is only accurate locally at $\mathbf{x}_0$; (b) a good location $\mathbf{x}_0$ maximizes the gradient, because that point is intuitively where a tiny change in the input space would make the large change in the classification. Therefore, finding a good location $\mathbf{x}_0$ and its gradient would reveal the most discriminative features for a given classification.

However, directly using $\boldsymbol{\phi}$ and its gradients violates privacy, as $\boldsymbol{\phi}$ contains sensitive information about individuals from the training dataset. In practise, privatizing $\boldsymbol{\phi}$ often proves to be difficult because of its unknown sensitivity, *in other words*, we do not know how much noise to add to privatize $\boldsymbol{\phi}$. Thus, we cannot use $\boldsymbol{\phi}$ and its gradients, unless we privatize the parameters of $\boldsymbol{\phi}$ or we train $\boldsymbol{\phi}$ using public data. The first – if at all possible – typically incurs a high privacy loss to meet a certain level of classification accuracy as $\boldsymbol{\theta}$ is typically high-dimensional; and the latter is often infeasible as public data is often simply not available. Here, we present a different approach than post-processing a trained neural network to simultaneously obtain high classification accuracy and interpretability while preserving privacy.

3.1 Locally Linear Maps

We introduce a set of local functions f_k to approximate the score function at each class s_k, and parameterize each f_k by a combination of M linear maps denoted by g_m^k, *as if* we directly learned the gradient directions of $\boldsymbol{\phi}$ at M input points per class:

$$f_k(\mathbf{x}) = \sum_{m=1}^{M} \sigma_m^k \; g_m^k(\mathbf{x}), \tag{3}$$

$$\text{where } g_m^k(\mathbf{x}) = \mathbf{w}_m^{k\,\top}\mathbf{x} + \mathbf{b}_m^k, \quad \text{and} \quad \sigma_m^k(\mathbf{x}) = \frac{\exp\left[\beta \cdot g_m^k(\mathbf{x})\right]}{\sum_{m=1}^{M} \exp\left[\beta \cdot g_m^k(\mathbf{x})\right]}. \tag{4}$$

The M linear maps are weighted separately for each class using the weighting coefficients σ_m^k, which determine how *important* each linear map is for classifying a given input. One way to choose the weighting coefficients is by assigning a probability to each linear map using the softmax function as in Eq. (4). We introduce a global inverse temperature parameter β in the softmax to tune the sensitivity of the relative weighting – large β (small temperature) favours single filters; small β (high temperature) favours several filters. The softmax weighting is a heuristic we chose to avoid the non-identifiability issues of parameters in mixture models.

Although we motivated our method from the point of view of a first-order Taylor approximation, we *cannot* identify the gradients of $\boldsymbol{\phi}$ at various input points from the learned linear maps $\mathbf{w}_m^k$, as there can be many input points that produce the same gradients[2]. Hence, we only show the difference and similarity in the features learned from the neural-net-based classifier and the locally linear maps in Sec. 4, where we visualize the qualitative differences and similarities tested on several datasets.

We train the LLM by optimizing the following (standard) cross-entropy loss:

$$\mathcal{L}(\mathbf{W}, \mathcal{D}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} y_{n,k} \log \hat{y}_{n,k}(\mathbf{W}), \tag{5}$$

where we denote the parameters of LLM collectively by $\mathbf{W}$, and we define the predictive class label by the mapping from the pre-activation through another softmax function, $\hat{y}_{n,k}(\mathbf{W}) = \exp(f_k(\mathbf{x}_n))/[\sum_{k'=1}^{K} \exp(f_{k'}(\mathbf{x}_n))]$.

3.2 Differentially private LLM

To produce differentially private locally linear map parameters $\tilde{\mathbf{W}}$, we adopt the moments accountant method combined with the gradient-perturbation technique. This involves (a) perturbing gradients at

[2] However, we *can* identify the linear maps from the gradients of $\boldsymbol{\phi}$, though this violates privacy.

each learning step when optimizing Eq. (5) for all locally linear map parameters $\mathbf{W}$; and (b) using the moments accountant method to compute the cumulative privacy loss after the training is over.

When we perturb the gradient, we need to ensure to add the right amount of noise. As there is no way of knowing how much change a single datapoint would make in the gradient's L2-norm, we rescale all the datapoint-wise gradients, $\mathbf{h}_t(\mathbf{x}_n) := \nabla_{\mathbf{W}}\mathcal{L}(\mathbf{W}, \mathcal{D}_n)$ for all $n = \{1, \cdots, N\}$, by a pre-defined norm clipping threshold, C, as used in [1], i.e.,

$$\bar{\mathbf{h}}_t(\mathbf{x}_n) \leftarrow \mathbf{h}_t(\mathbf{x}_n)/\text{max}(0, \|\mathbf{h}_t(\mathbf{x}_i)\|_2/C). \tag{6}$$

Algorithm 1 summarizes this procedure, Now we formally state that the resulting locally linear maps are differentially private in Theorem 3.1.

Algorithm 1 DP-LLM for interpretable classification

```
Require: Dataset $\mathcal{D}$, norm-clipping threshold $C$, privacy parameter $\sigma^2$, and learning rate $\eta_t$
Ensure: $(\epsilon, \delta)$-DP locally linear maps for all $K$ classes, $\tilde{\mathbf{W}}$
  for number of training steps $t \leq T$ do
    1: For each minibatch of size $L$, we noise up the gradient after clipping the norm of the datapoint-wise gradient given in Eq. (6) via $\tilde{\mathbf{h}}_t \leftarrow \frac{1}{L}\left[\sum_{n=1}^{L} \bar{\mathbf{h}}_t(\mathbf{x}_n) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right]$.
    2: Then, we make a step in the descending direction by $\tilde{\mathbf{W}}_{t+1} \leftarrow \tilde{\mathbf{W}}_t - \eta_t \tilde{\mathbf{h}}_t$.
  end for
  Calculate the cumulative privacy loss $(\epsilon, \delta)$ using the moments accountant.
```

Theorem 3.1. *Algorithm 1 produces (ϵ, δ)-DP locally linear maps for all K classes.*

The proof is provided in Appendix Sec. B.

For high-dimensional inputs such as images, we found that adding noise to the gradient corresponding to the full dimension of $\mathbf{W}$ lead to very low accuracies for private training. Therefore, we propose to incorporate the random projection matrix $\mathbf{R}_m \in \mathbb{R}^{D' \times D}$ with $D' \ll D$, which is shared among classes k, to first decrease the dimensionality of the parameters that need to be privatized. We now have the following parameterization for each locally linear maps, $\mathbf{w}_m^k = \mathbf{m}_m^k \mathbf{R}_m$, where the effective parameter for each local linear map is $\mathbf{m}_m^k \in \mathbb{R}^{D'}$. We perturb the gradient of $\mathbf{m}_m^k$ for all k and m in each training step in Algorithm 1 to produce differentially private linear maps, $\tilde{\mathbf{w}}_m^k = \tilde{\mathbf{m}}_m^k \mathbf{R}_m$.

Due to Prop. 1, we can use the differentially private locally linear maps to make predictions on test data. Here we focus on guarding the training data's privacy and assume that the test data do not need to be privatized, which is a common assumption in DP literature.

4 Experiments

In this section we evaluate the trade-off between accuracy, privacy, and interpretability for our LLM model on several datasets and compare to other methods where appropriate. Our implementation is available on GitHub[3].

4.1 MNIST Classification

We consider the classification of MNIST [11] and Fashion-MNIST [28] images with the usual train/test splits and train a CNN[4] as a baseline model, which has two convolutional layers with 5x5 filters and first 20, then 50 channels each followed by max-pooling and finally a fully connected layer with 500 units. The model achieves 99% test accuracy on MNIST and 87% on Fashion-MNIST.

We train several LLMs in the private and non-private setting. By default, we use LLM models with $M = 30$ predictions per class and random projections to $D' = 300$ dimensions, which are optimized for 20 epochs using the Adam optimizer with learning rate 0.001, decreasing by 20% every 5 epochs.

[3] `https://github.com/frhrdr/dp-llm`

[4] `https://github.com/pytorch/examples/blob/master/mnist/main.py`

On MNIST the model benefits from an increased inverse softmax temperature $\beta = 1/30$, while $\beta = 1$ is optimal for Fashion-MNIST. We choose a large batch size of 500, as this improves the signal-to-noise ratio of our algorithm. In the private setting we clip the per-sample gradient norm to $C = 0.001$ and train with $\sigma = 1.3$, which gives this model an $(\epsilon = 2, \delta = 10^{-5})$-DP guarantee via the moments accountant. For the low privacy regime $\epsilon \geq 4$ we train with a batch size of 1500 and for 60 epochs.

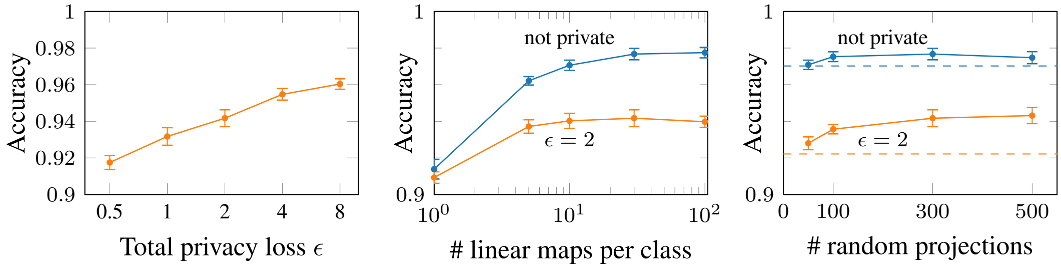

Figure 2: Accuracy of our LLM model on the MNIST testset for different levels of privacy and different model configurations in the private (—) and non-private (—) setting. Errorbars are 2 stdev from 10 random restarts; dashed lines on the right (- - -) denote no random projections.

First, we consider the trade-off between privacy and accuracy (Fig. 1 Ⓒ) in Fig. 2 (left). Note that current privatized network methods [1, 19] achieve an accuracy of 95% for $\epsilon = 2$ and up to 92% for $\epsilon = 0.5$, which is comparable to our mean accuracy of $94.2 \pm 0.4\%$ and $91.8 \pm 0.4\%$ respectively (on Fashion-MNIST we achieve $80.7 \pm 0.6\%$ and $83.2 \pm 0.4\%$). However, such a privatized network does not provide transparent explanations as opposed to our approach. In the remainder of Fig. 2 we study the impact of varying the number of filters per class M (center) and the output dimensionality of the random projections D' (right) in private and non-private LLM models. Private LLMs deteriorate beyond a certain number of linear maps due to the increased noise needed to privatize them, whereas non-private models continue to benefit from additional filters. Increasing the dimensionality of the random projections benefits private training.

Next, we show the trade-offs with interpretability (Fig. 1 Ⓐ and Ⓑ). For this, we investigate the learned LLM filters under increasing privacy guarantees and increased private utility as shown in Fig. 3. In the unconstrained setting, filter selection assigns high weights to single filters, which is why in the low privacy settings top and bottom row are hardly different. In the high privacy setting this evens out and we see that while the individual filters become hard to distinguish as intermediate dimension D' and privacy loss ϵ are decreased, the weighted average of filters used for classification still provide a good local explanation of what parts of the given input the model is sensitive to. Additional private filter visualizations for Fashion-MNIST are shown in Fig. 10 in the appendix.

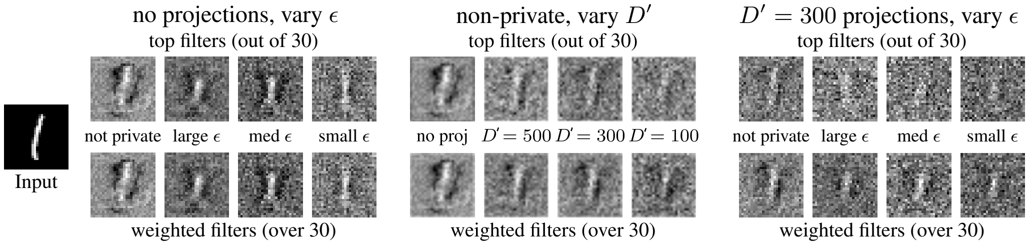

Figure 3: Decay of filter interpretability in an LLM in three different settings. *left*: increasing privacy guarantee, *center*: reducing the dimensionality D' of random projections, *right*: increasing privacy at fixed 300 random projections. Top row: highest activated filters in target class, Bottom row: sum of filters in target class weighted by contribution

In order to highlight the advantage of the simplistic LLM architecture in terms of immediate interpretability, we compare the learned filters our model uses to classify certain images to two attribution methods for a neural network trained on the same data. We train a simple CNN and an LLM on Fashion-MNIST to matching 87% test accuracy. We then use SmoothGrad [24] and integrated gradients [26] to visualize the CNN's sensitivity to test images and compare these methods to LLM filters in Fig. 4 (and Fig. 8). Note that we did not multiply the integrated gradient with the input image, as Fashion-MNIST images have a mask-like effect which occludes the partial output of the method. We observe that both alternative attribution methods produce similar outputs, which are nonetheless hard to interpret, whereas the LLM filters show simplistic prototype images of the classes they are associated with. This is further illustrated in Fig. 5 where we show the three highest weighted filters for test images from three classes. The diversity of filters varies a lot for different class labels, as some are more varied and harder to discriminate than others. For instance, while the sandal class (center) has filters which distinguish between different types of heels, the ankle boot filters (right) show similar coarse features, which are sufficient for classifying a majority of the inputs correctly. The coat (left) filters are mostly selective in the shoulder region and general silhouette, but some filters also track other features like arms, collar and zipper.

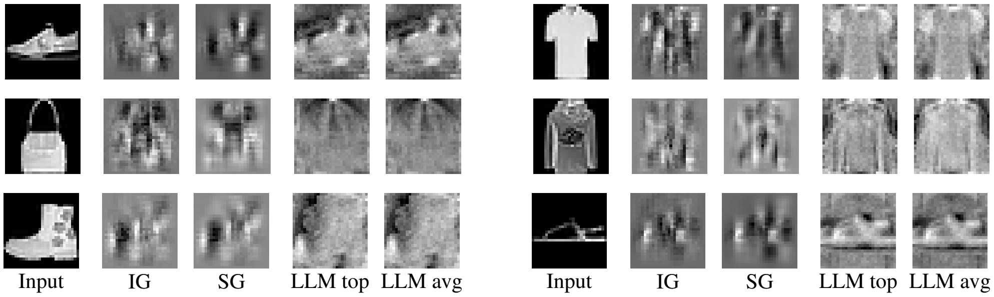

Figure 4: Comparison of interpretability CNN and LLM. *left to right*: input image, integrated gradient (IG) of CNN, smoothed gradient (SG) of CNN, top filter of LLM, weighted target class filters of LLM

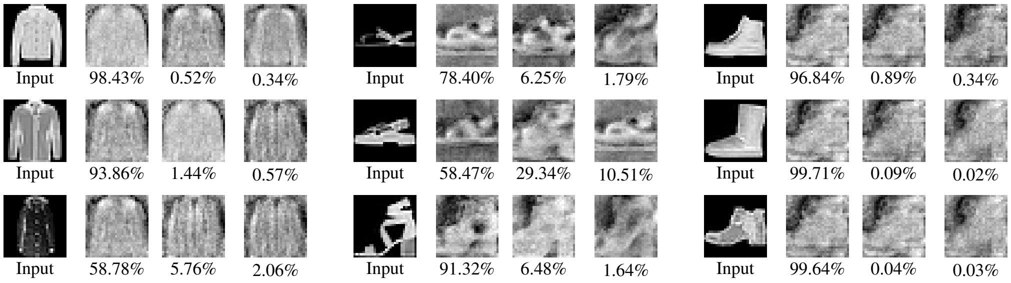

Figure 5: Top 3 filters with associated weightings for test images from three classes.

4.2 Disease classification in a medical dataset

As a second task we consider disease classification in the Henan Renmin Hospital Data [12, 14][5]. It contains 110,300 medical records with 62 input features and 3 binary outputs. The input features are 4 basic examinations (sex, BMI, distolic, systolic), 26 items from blood examinations, 12 items from urine examinations, and 20 items from liver function tests. The three binary outputs denote three medical conditions – hypertension, diabetes, and fatty liver – which can also co-occur. Following [14] we transform this multi-label task into a multi-class problem by considering the powerset of the three binary choices as eight independent classes. Because these classes are highly imbalanced, we only retained the four most common classes, leaving us with 100,140 records.

By default, we use an LLM model with $M = 2$ predictions per class and no random projections, which is optimized for 20 epochs using the Adam optimizer with learning rate 0.01, decreasing by 20% every 5 epochs. We choose a batch size of 256. In the private setting we clip the per-sample gradient norm to 0.001 and train with $\sigma = 1.25$, which gives this model an $(\epsilon \approx 1.5, \delta = 2 \cdot 10^{-5})$-DP guarantee via the moments accountant.

We train a baseline DNN (3 fully connected hidden layers with 128 units each) as well as several LLMs with varying number of linear filters per class in private and non-private settings. In Fig. 6 we

[5] The dataset is provided by [14] and was available at `http://pinfish.cs.usm.edu/dnn/`

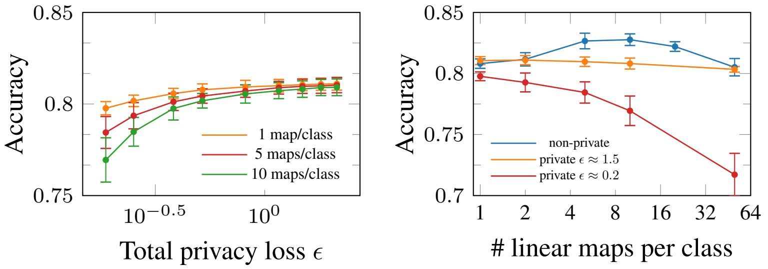

Figure 6: Accuracy of our LLM model on the Henan Renmin Hospital testset for different levels of privacy and different model configurations in the private and non-private setting. Errorbars are 2 stdev for 10 random restarts.

visualize the trade-off between accuracy and privacy for varying privacy losses as well as numbers of linear maps. Like before, the accuracy deteriorates as we decrease the privacy loss (Fig. 6 *left*). As the number of linear maps per class is increased (Fig. 6 *right*), the accuracy for the private models also drops due to the privacy budget being spread across more parameters. We attribute the drop in performance for the non-private LLM with number of maps to optimization difficulties and local minima as well as higher sensitivity to hyperparameters. A small number of maps (between 2 and 5) is sufficient for this datasets, especially in the private setting. Our LLMs attain $82.8 \pm 0.5\%$ (non-private), $82.0 \pm 0.4\%$ ($\epsilon \approx 1.5$), and $79.8 \pm 0.4\%$ ($\epsilon \approx 0.2$) compared to $84 \pm 0.5\%$ for a non-private DNN.

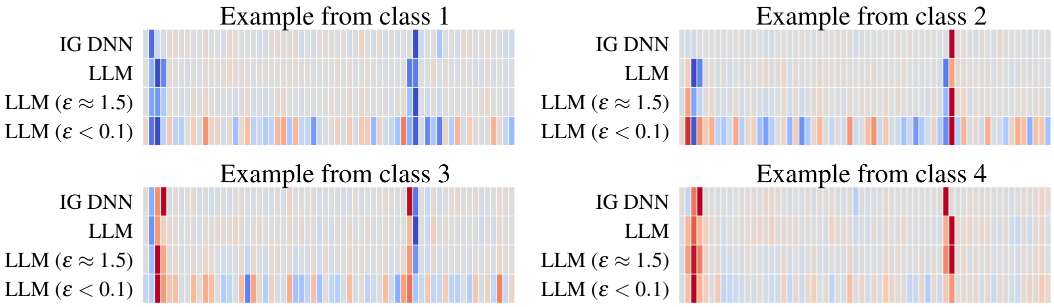

Figure 7: Integrated gradient (IG) and weighted linear filters (LLM; our method) for all 62 feature for one example from each class from the Henan Renmin dataset. For LMM we consider the non-private case (LLM) as well as two private cases with strong ($\epsilon < 0.1$) and weaker ($\epsilon \approx 1.5$) privacy. Entries are normalized and colorcoded between 🟦 $= -1$, ⬜ $= 0$, and 🟥 $= 1$.

In Fig. 7 we consider an example from each class and show the weighted linear maps by the LLM for each example as well as its integrated gradients (IGs) [26]. For our LLM we consider the non-private and two private cases. In general, there is good agreement between all attribution methods; they are relatively sparse and focus on a small set of features. We found that IGs varied much more between examples from the same class than our LLMs (see Fig. 9 in Appendix C.2). For strong privacy ($\epsilon < 0.1$), the linear maps are much less sparse, highlighting the trade-off between interpretability and privacy.

5 Related Work

Interpretability. The *saliency* map and *gradient-based attribution* methods are one of the most popular explanation methods that identify relevant regions and assign importance to each input feature (e.g., pixel for image data) [2, 3, 16, 21, 22, 24, 26]. These methods typically use first-order gradient information of a complex model with respect to inputs, to produce maps that indicate the relative importance of the different input features for the classification. An obvious downside of these approaches is that they provide explanations conditioned on only *a single* input and hence it is necessary to manually assess each input of interest in order to draw a class-wide conclusion. In contrast, our approach can draw class-wide conclusions without manually assessing each input, because it outputs the most relevant explanations in terms of a collection of linear maps for each class. For explanations conditioned on any specific input, our model can provide an input-dependent weighted collection of these features related to that specific input.

Privacy. To privatize complex models, such as deep neural networks, a popular approach is to add noise to the gradients in the stochastic gradient descent (SGD) algorithm [1, 15, 18]. An alternative approach is to directly perturb the objective with additive noise [19, 20, 29]. In these works, the objective function is approximated by the Taylor expansion, and the resulting coefficients of the polynomials are perturbed before training. We found the latter approach less practical than the former, as we need to choose which order of polynomial degree to use. Typically, adding more layers introduces a more nested-ness in the objective function in which case using a higher order approximation is more suitable to approximate the loss function accurately. A high degree of polynomial approximation, however, increases the privacy loss as the dimensionality of the coefficients grow. From our perspective, the gradient perturbation method is simple to use and model agnostic, and there are many successful examples of the gradient perturbation methods used in slightly different settings than privatizing a whole model from scratch, such as knowledge transfer from teacher models to student models in [1, 17, 18]. However, none of these methods took interpretability into account, and some of the work assume the availability of public data to train a significant part of their model to decrease the necessary privacy budget to train the entire model. In our method, no access to public data is assumed and interpretability through linear maps is a key component of the trained model.

Mixtures of Experts. Our LLMs are reminiscent of *Mixture of experts* (ME) models. A ME assigns different specialized linear models to different parts of input space in a discriminative task (see [13] for a broad overview of existing ME models). In our case, each local expert model is class specific and contributes to a weighted linear map for that class. The weighting provides an input-dependent *significance* for each linear map, and considering more than one map per class brings in the flexibility to fit the data better. Another relevant model is the *Mixture of factor analyzer* (MFA), which also has a very similar flavour as the ME models, but developed for density estimation of high-dimensional real-valued data [8].

6 Conclusion and Discussion

We proposed a family of simple models that aim to approximate neural-net-based models using several *locally linear maps* (LLM) per class to provide interpretable features in a privacy-preserving manner while maintaining high classification accuracy. Results on two image benchmark datasets as well as a medical dataset indicate that a reasonable trade-off between classification accuracy, privacy *and* interpretability can indeed be struck and tuned by varying the number of linear maps. Nevertheless, several open questions for future research remain. First, the datasets in this paper are still relatively simple, such that it would be intriguing to see the limits of complexity the LLM model can model with a sufficiently high accuracy. Second, the current model does not interact with a larger and richer counterpart, such as a neural network, due to privacy constraints. It would be interesting to investigate if gaining gradient information of a more flexible model at particularly important input points in a differentially private way would be possible, in order to combine benefits of both models.

Acknowledgements

The work of M. Park and F. Harder is supported by the Max Planck Society, as well as the Gibs Schüle Foundation and the Institutional Strategy of the University of Tübingen (ZUK63). M. Bauer gratefully acknowledges partial funding through a Qualcomm studentship in technology as well as by the EPSRC. We would like to thank Maryam Ahmed for useful comments on the manuscript, Patrick Putzky and Wittawat Jitkritum for their scientific insights, and Been Kim and Isabel Valera for their inputs on interpretability research. Lastly, M. Park is grateful to Gilles Barthe for the inspiration of studying the trade-off between interpretability and privacy.

References

[1] M. Abadi, A. Chu, I. Goodfellow, H. Brendan McMahan, I. Mironov, K. Talwar, and L. Zhang. "Deep learning with differential privacy". In: *ArXiv e-prints* (2016).

[2] M. Ancona, E. Ceolini, A. C. Öztireli, and M. H. Gross. "A unified view of gradient-based attribution methods for Deep Neural Networks". In: *CoRR* (2017).

[3] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". In: *PLOS ONE* 7 (2015).

[4] J. Blocki, A. Blum, A. Datta, and O. Sheffet. "The Johnson-Lindenstrauss Transform Itself Preserves Differential Privacy". In: *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science* (2012).

[5] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D. Song. "The Secret Sharer: Measuring Unintended Neural Network Memorization & Extracting Secrets". In: *CoRR* (2018).

[6] C. Dwork and A. Roth. "The Algorithmic Foundations of Differential Privacy". In: *Found. Trends Theor. Comput. Sci.* (2014).

[7] 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*. 2015.

[8] Z. Ghahramani and G. E. Hinton. *The EM Algorithm for Mixtures of Factor Analyzers*. Tech. rep. University of Toronto, 1997.

[9] B. Goodman and S. Flaxman. "European Union regulations on algorithmic decision-making and a "right to explanation"". In: *arXiv e-prints*, arXiv:1606.08813 (2016).

[10] K. Kenthapadi, A. Korolova, I. Mironov, and N. Mishra. "Privacy via the Johnson-Lindenstrauss Transform". In: *Journal of Privacy and Confidentiality* 1 (2013).

[11] Y. LeCun and C. Cortes. "MNIST handwritten digit database". In: (2010).

[12] R. Li, W. Liu, Y. Lin, H. Zhao, and C. Zhang. "An Ensemble Multilabel Classification for Disease Risk Prediction". In: *Journal of healthcare engineering* (2017).

[13] S. Masoudnia and R. Ebrahimpour. "Mixture of experts: a literature survey". In: *Artificial Intelligence Review* 2 (2014).

[14] A. Maxwell, R. Li, B. Yang, H. Weng, A. Ou, H. Hong, Z. Zhou, P. Gong, and C. Zhang. "Deep learning architectures for multi-label classification of intelligent health risk prediction". In: *BMC Bioinformatics* 14 (2017).

[15] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang. "Learning Differentially Private Language Models Without Losing Accuracy". In: *CoRR* (2017).

[16] G. Montavon, S. Bach, A. Binder, W. Samek, and K. Müller. "Explaining NonLinear Classification Decisions with Deep Taylor Decomposition". In: *CoRR* (2015).

[17] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and U. Erlingsson. "Scalable Private Learning with PATE". In: *International Conference on Learning Representations*. 2018.

[18] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar. "Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2017.

[19] N. Phan, X. Wu, and D. Dou. "Preserving Differential Privacy in Convolutional Deep Belief Networks". In: *ArXiv e-prints* (2017).

[20] N. Phan, Y. Wang, X. Wu, and D. Dou. *Differential Privacy Preservation for Deep Auto-Encoders: an Application of Human Behavior Prediction*. 2016.

[21] M. T. Ribeiro, S. Singh, and C. Guestrin. “"Why Should I Trust You?": Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016.

[22] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. “Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization”. In: *CoRR* (2016).

[23] R. Shokri and V. Shmatikov. “Privacy-preserving deep learning”. In: *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 2015.

[24] D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg. “SmoothGrad: removing noise by adding noise”. In: *CoRR* (2017).

[25] C. Song, T. Ristenpart, and V. Shmatikov. “Machine Learning Models That Remember Too Much”. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2017.

[26] M. Sundararajan, A. Taly, and Q. Yan. “Axiomatic Attribution for Deep Networks”. In: *CoRR* (2017).

[27] P. Voigt and A. v. d. Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. 1st. Springer Publishing Company, Incorporated, 2017.

[28] H. Xiao, K. Rasul, and R. Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. 28, 2017.

[29] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett. “Functional Mechanism: Regression Analysis under Differential Privacy”. In: *ArXiv e-prints* (2012).

Supplementary Material for Locally Linear Maps

A A short summary for moments accountant method

The moments accountant. The privacy loss in eq. 1 is a random variable once we add noise to the output of the algorithm. In fact, when we add Gaussian noise, the privacy loss random variable is also Gaussian distributed. Using the tail bound of Gaussian privacy loss random variable, the *moments accountant* method [1] provides a clever way of combining ϵ and δ such that the resulting total privacy loss is significantly smaller than other composition methods.

In the *moments accountant* method, the cumulative privacy loss is calculated by bounding the moments of the privacy loss random variable $L^{(o)}$. First off, each λ-th moment, where λ can be positive integers, is defined as the log of the moment generating function evaluated at λ, i.e., $\alpha_{\mathcal{M}}(\lambda; \mathcal{D}, \mathcal{D}') = \log \mathbb{E}_{o \sim \mathcal{M}(\mathcal{D})} \left[e^{\lambda L^{(o)}} \right]$. Then, by taking the maximum over the neighbouring datasets, we compute the worst case λ-th moment by, $\alpha_{\mathcal{M}}(\lambda) = \max_{\mathcal{D}, \mathcal{D}'} \alpha_{\mathcal{M}}(\lambda; \mathcal{D}, \mathcal{D}')$, where the form of $\alpha_{\mathcal{M}}(\lambda)$ is determined by the moment of a Gaussian random variable. The moments accountant then computes $\alpha_{\mathcal{M}}(\lambda)$ at each step. The composability theorem (Theorem 2.1 in [1]) states that the λ-th moment composes linearly if we add independent noise at each training step. So, we can simply sum up the upper bound on each $\alpha_{\mathcal{M}_t}$ to obtain an upper bound on the total λ-th moment after T compositions, $\alpha_{\mathcal{M}}(\lambda) \leq \sum_{t=1}^{T} \alpha_{\mathcal{M}_t}(\lambda)$. Finally, once the moment bound is computed, we can convert the λ-th moment to the (ϵ, δ)-DP guarantee by, $\delta = \min_{\lambda} \exp \left[\alpha_{\mathcal{M}}(\lambda) - \lambda\epsilon \right]$, for any $\epsilon > 0$. See Appendix A in [1] for the proof.

B Proof of Theorem 1

Proof. We first prove that one gradient step in Algorithm 1 produces differentially private locally linear maps, then generalize this result for the T number of gradient steps.

Given an initial *data-independent* value of $\mathbf{W}_0$, if we add Gaussian noise to the norm-clipped gradient evaluated on the subsampled data with the samping rate $q = L/N$, then due to the *Gaussian mechanism* (Theorem 3.22 in [2]) and Theorem 1 in [1], the resulting estimate $\tilde{\mathbf{W}}_1$ from a single gradient step (i.e., the step **2** in Algorithm 1) is (ϵ', δ')-differentially private, where $\sigma \geq c \cdot q\sqrt{\log(1/\delta')}/\epsilon'$ with some constant c. Now, as $\tilde{\mathbf{W}}_1$ is already privatized, we can make further gradient steps from $\tilde{\mathbf{W}}_1$, which makes $\tilde{\mathbf{W}}_2$ also (ϵ', δ')-differentially private, as the only part that depends on the data is the gradient which we perturb for privacy. Applying the same amount of Gaussian noise to the gradient in each step ensures each $\tilde{\mathbf{W}}_t$ for all t also (ϵ', δ')-differentially private.

Finally, the composibility and tail bound in Theorem 2 in [1] proves that the cumulative privacy loss after T training steps computed by the moments accountant method ensures (ϵ, δ)[6]-DP locally linear maps. □

[6] Note that we can identify the exact relationship between the cumulative loss (ϵ, δ) and (ϵ', δ') numerically only, due to the constant factor in $\sigma \geq c \cdot q\sqrt{\log(1/\delta')}/\epsilon'$. We use code published by [1] to compute these numerically.

C Additional Experimental Results

C.1 Comparison of attribution methods on MNIST

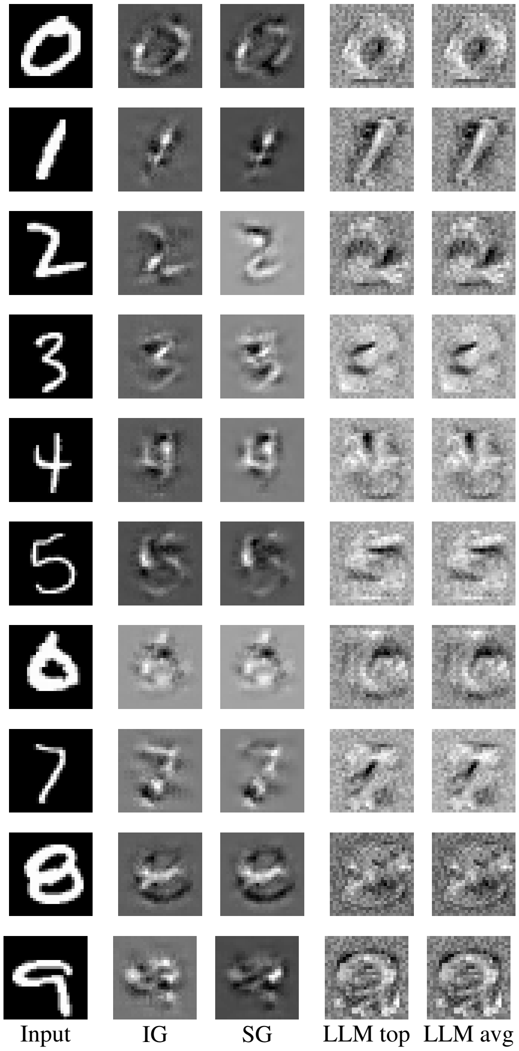

Figure 8: Comparison between different attribution methods, similar to Fig. 4 but for MNIST. On this dataset, the network attributions resemble more closely than on Fashion-MNIST, highlighting relevant edges. LLM filters exhibit the same kind of coarse prototypical images with pronounced edges as on Fashion-MNIST.

C.2 Attribution methods on medical data

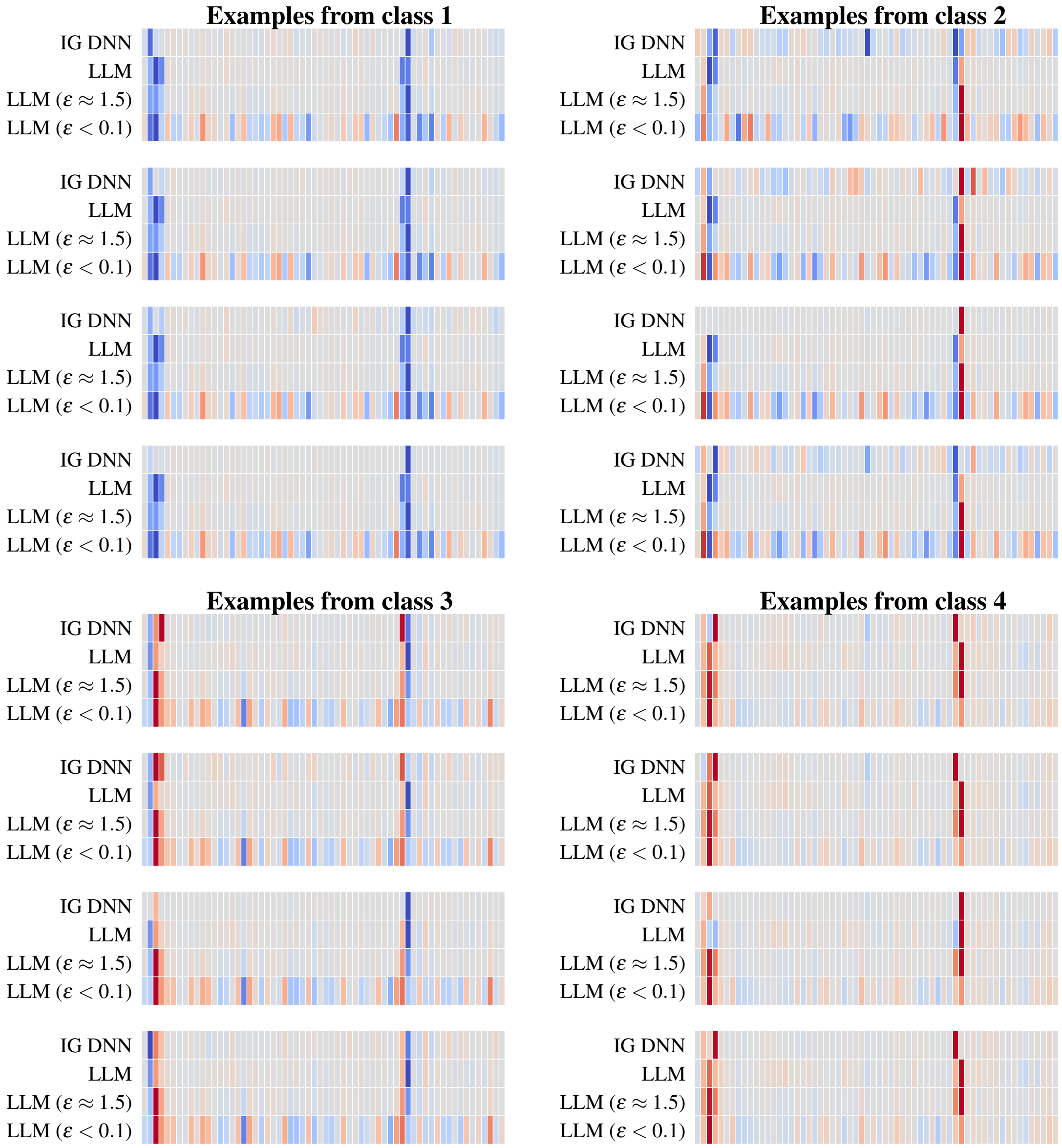

Figure 9: Integrated gradient (IG) and weighted linear filters (LLM; our method) for all 62 feature for four example from each class from the Henan Renmin dataset. For LMM we consider the non-private case (LLM) as well as two private cases with strong ($\epsilon < 0.1$) and weaker ($\epsilon \approx 1.5$) privacy. Entries are normalized and colorcoded between ■ $= -1$, □ $= 0$, and ■ $= 1$. This is an extended version of Fig. 7. Note that there is less variability between explanations/attributions for LLM (non-private) than there is for integrated gradients.

C.3 Private LLM Filters

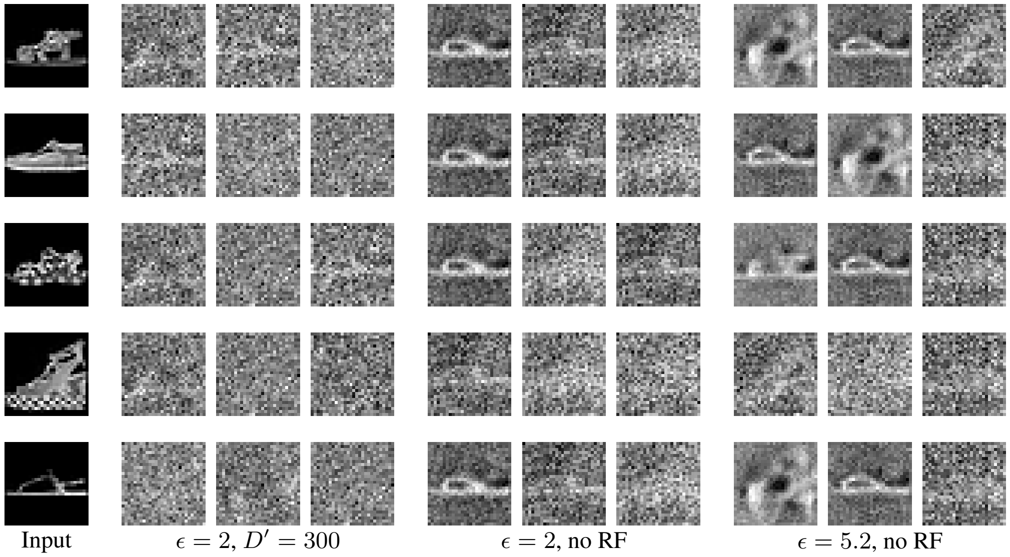

Figure 10: Highest activated filters for 5 test inputs under 3 differentially private setups, the leftmost filter having highest activation. We compare the default setting with random filters with $D' = 300$ (left), the same setting without random filters, but still $\epsilon = 2$, which incurs a loss in test accuracy but yields clearer filters (center), and a lower privacy setting trained with $\sigma = 1.0$ for 60 epochs, which amounts to $\epsilon = 5.2$ (right). As the level of noise is reduced, more interpretable filters are retrieved. When optimizing for accuracy in the high privacy case, however, we see that model interpretability suffers significantly.