Supplementary Materials for "Efficient Unlearning with Privacy Guarantees"

Author information scrubbed for double-blind reviewing

No Institute Given

The content herein provides additional experimental setup, deeper analysis of hyperparameter impacts, and detailed runtime measurements that support the findings presented in the main paper.

1 Experimental setup details

The experiments were conducted on a system running Windows 11 Home OS, equipped with a 12$^{\text{th}}$ Gen Intel®Core™i7-12700 12-core CPU, 32 GB RAM, and an NVIDIA GeForce RTX 4080 with 16 GB GPU.

1.1 Datasets

We used four publicly available datasets, each one representing a classification problem from a different domain. Three of these datasets are tabular and were chosen because of their privacy relevance, as they contain records describing personal data of individuals. Also, we included an image classification dataset to evaluate the generality of our approach across diverse data types.

- *Adult income* [1]: it comprises 32,561 training records and 16,281 testing records of demographic and financial data, with six numerical and eight categorical attributes. The class attribute indicates whether an individual makes more than 50K dollars a year.
- *Heart disease* [2]: it contains 55,869 training records and 14,131 testing records of patient data, with five numerical and six categorical measurements related to cardiovascular diseases. The class attribute denotes the presence of a heart disease.
- *Credit information* [3]: it includes 96,215 training records and 24,054 testing records of financial information, with ten numerical attributes. The class attribute indicates whether an individual has experienced financial distress.
- *CIFAR-10* [4]: it is a widely-used image classification dataset containing 60,000 32x32 pixel images across ten classes, with three RGB channels. The dataset is split into 50,000 training samples and 10,000 testing samples.

[1] `https://archive.ics.uci.edu/ml/datasets/Adult`
[2] `https://www.kaggle.com/sulianova/cardiovascular-disease-dataset`
[3] `https://www.kaggle.com/c/GiveMeSomeCredit`
[4] `https://www.cs.toronto.edu/~kriz/cifar.html`

1.2 ML models

For each tabular dataset, we built two ML classification models: a multi-layer perceptron (MLP) and an XGBoost classifier. We implemented the MLP models using PyTorch with an input layer, a hidden layer, and an output layer. As for the XGBoost classifier, we utilized the implementation provided by XGBoost [3], which can be found in the official XGBoost website [5]. For CIFAR-10, we utilized the DenseNet12 [8] deep model, which is known for its densely connected layers that enhance feature reuse and improve gradient flow.

1.3 Privacy models

We employed the two privacy models described in Section 4 of the paper for the tabular datasets. To enable a fair comparison, we applied both privacy models to all attributes except the class attribute. To implement k-anonymity, we used the MDAV microaggregation algorithm [4] with one-hot encoded categorical attributes and the parameter $k \in \{3, 5, 10, 20, 80\}$.

For DP, we applied the Laplace mechanism [6] for numerical attributes and the exponential mechanism [9] for categorical ones, with $\epsilon \in \{0.5, 2.5, 5, 25, 50, 100\}$. For DP on the Adult dataset, which contains several non-ordinal categorical attributes, we leveraged the unsupervised training capabilities of TabNet [1] to generate embeddings for these attributes, which allowed us to encode each non-ordinal categorical attribute into a 10-dimensional embedding vector. Then, we used the cosine similarity between the embeddings of attributes as a utility function for the exponential mechanism.

For the CIFAR-10 dataset we enforced DP via the DP-Pix methodology described in [7]. DP-Pix first pixelizes the image by averaging pixel values in blocks of size $b \times b$ to reduce sensitivity, and then adds noise to the pixels by using the Laplace mechanism based on the global sensitivity $255m/b^2$. The m parameter defines the number of different pixels between neighboring images. We used $m = 16$, as suggested by the DP-Pix author, and $b = 4$, which is consistent with the image resolution [7]. We did not enforce k-anonymity on CIFAR-10 images because aggregating or generalizing images in order to make them indistinguishable (as required by k-anonymity) produces meaningless images for any safe-enough k. As a matter of fact, the very few works employing k-anonymity on images use $k = 2$ [2,11,10].

1.4 Training settings

For each tabular dataset, we trained an MLP model and an XGBoost model from scratch on the whole training set $\mathbf{D}$. We used the cross-entropy loss and the Adam optimizer to train all MLP models. The MLP hidden layer was configured with 128 neurons for all benchmarks, with the exception of the Credit benchmark, which was configured with 256 neurons. For CIFAR-10, we trained

[5] `https://xgboost.ai/`

the DenseNet12 model using the cross-entropy loss and the SGD optimizer. The specific hyperparameters used during the training of these benchmarks are detailed in Table 1.

For SISA, we split the training set into 5 disjoint shards (each containing 10 slices) and applied the SISA training procedure using the training hyperparameters presented in Table 1.

For our EUPG with tabular data, we obtained the base private models M^k and M^ϵ by training them on the $\mathbf{D}^k$ and $\mathbf{D}^\epsilon$ protected datasets, respectively, using the same training hyperparameters in Table 1. For EUPG on CIFAR-10, we trained on $\mathbf{D}^\epsilon$ only. We then fine-tuned M^k and M^ϵ on $\mathbf{D}$ for some epochs (see details below) to obtain $M^k_{\mathbf{D}}$ and $M^\epsilon_{\mathbf{D}}$ with the learning rates and batch size presented in Table 1.

Table 1: Training hyperparameters

Dataset	Model	Hyperparameters
Adult income	MLP	BS:512, LR:1e-2, Epochs:100
	XGBoost	Estimators:300, Depth:10, LR:0.5, λ:5
Heart disease	MLP	BS:512, LR:1e-2, Epochs:200
	XGBoost	Estimators:200, Depth:7, LR:0.5, λ:5
Credit	MLP	BS:256, LR:1e-3, Epochs:200
	XGBoost	Estimators:200, Depth:9, LR:0.5, λ:5
CIFAR-10	DenseNet12	BS:64, Epochs:100 LR:(1e-1 for train., 1e-2 for finetun.)

2 Detailed analysis of hyperparameter impact

In this section, we discuss the influence of key hyperparameters. For each of them, we report results on the datasets and models that best illustrate the effect of varying its value.

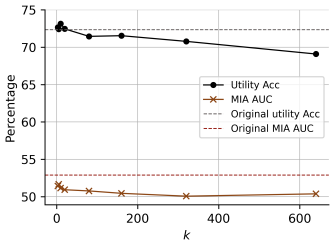
2.1 Impact of parameter k

Figure 1a depicts the impact of parameter k (when using k-anonymity to protect training data) on utility and forgetting. The figure reports the performance of the MLP model trained on the anonymized Heart dataset (without fine-tuning, M^k) across various values of k in terms of accuracy (utility) and MIA AUC scores (forgetting). The results display a noticeable trend, where the accuracy peaks at $k = 10$, and then gradually decreases as k increases. This suggests that a moderate level of anonymity (with $k = 10$) strikes an optimal balance for utility due to effective generalization of the training data without losing

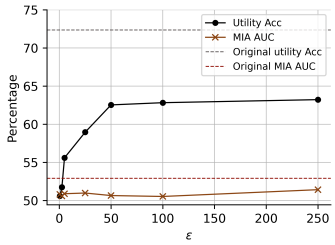
important features. At lower values of k ($k = 3$), the accuracy is slightly lower than at $k = 10$, which might be due to insufficient ML model generalization. As k increases beyond 10, the utility diminishes gradually. This is indicative of over-generalization, where the ML model loses useful information to make accurate predictions.

The MIA AUC score is slightly higher with $k = 3, 5$, but the improvement in forgetting does not linearly correlate with the increase of k. However, in general, forgetting improves as k increases.

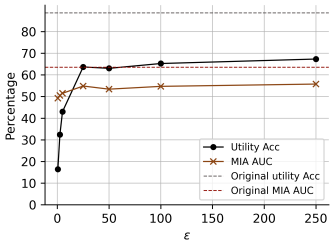
In summary, the value of k defines a trade-off between the utility of the model and the privacy (or forgetting) guarantees, in line with the typical use of k-anonymity to protect microdata. Therefore, it is crucial to choose a value of k that maximizes the model utility while providing enough privacy. Given the results in Figure 1a, we took $k = 10$ in the rest of our experiments.



(a) Impact of k for the MLP model trained on the anonymized Heart dataset.



(b) Impact of ϵ for the MLP model trained on the anonymized Heart dataset.



(c) Impact of ϵ for the DenseNet12 model trained on the anonymized CIFAR-10 dataset.

Fig. 1: Impact on utility and forgetting of privacy parameters.

2.2 Impact of parameter ϵ

Figure 1b shows the impact of the differential privacy parameter ϵ on utility and forgetting. The figure outlines the performance of MLP trained on the differentially private Heart dataset (without fine-tuning, M^ϵ) with various values of ϵ in terms of accuracy (utility) and MIA AUC scores (forgetting). As ϵ increases, which corresponds to a relaxation of the privacy guarantees, there is a notable improvement in model accuracy. This improvement is especially significant as ϵ goes from 0.5 to 25, suggesting that a moderate relaxation of privacy can substantially enhance the ML model's ability to learn from the data. However, the rate of accuracy improvement tapers off for very high ϵ values, approaching a plateau that hints at diminishing returns on utility gains. The MIA AUC scores remain relatively stable across the spectrum of ϵ values, which indicates that even a light perturbation of data (corresponding to a weak DP guarantee) can significantly reduce the ML model's vulnerability to MIA.

Figure 1c shows a similar trend for CIFAR-10 with DenseNet12.

From Figure 1b and Figure 1c alone, it would seem that choosing, say, $\epsilon = 100$ would yield a better trade-off between utility and privacy than the value $\epsilon = 0.5$ we have taken in our experiments. The problem is that, although MIAs are the standard way to evaluate forgetting in the literature, they are rather weak attacks from the privacy point of view. In order to provide a fair comparison with the exact forgetting of SISA or retraining from scratch, which achieve top privacy (equivalent to taking $\epsilon = 0$ in DP), we could not take ϵ too large. We therefore chose $\epsilon < 1$ following the guidelines of [5], specifically $\epsilon = 0.5$. This level of privacy is comparable to exact forgetting and can protect against other attacks, such as reconstruction.

The above issue comes from the fact that the ϵ-DP guarantee on insensitivity to individual contributions becomes meaningless when ϵ is large. Notice that this does not happen with k-anonymity, whose privacy guarantee is meaningful for any $k \geq 2$: the re-identification probability is upper-bounded by $1/k$, and thus one can choose any value of k such that $1/k$ is considered to be an acceptable re-identification probability.

2.3 Impact of the number of fine-tuning epochs

Table 2 reports the impact of the number of fine-tuning epochs on utility and privacy for the MLP and XGBoost models trained on Adult with $\epsilon = 0.5$. We can see that fine-tuning significantly boosts the utility (accuracy) of both models, resulting in rapid gains in performance within the first 5 epochs. For MLP, utility jumps from 59.89% to approximately 84.92% and stabilizes with minimal changes beyond 5 epochs. For XGBoost, starting from a higher baseline of 73.25%, utility improves steadily up to 82.58% at 20 epochs.

Table 3 reports the same results for the DenseNet12 model trained on CIFAR-10. Here, the impact of fine-tuning epochs is more notable. Starting from a low baseline of 16.42% accuracy, the model shows a significant improvement, reaching

Table 2: Impact of the number of fine-tuning epochs on MLP and XGBoost trained on Adult with $\epsilon = 0.5$

Dataset	Epochs	MLP		XGBoost	
		Utility Acc(%)↑	MIA AUC(%)↓	Utility Acc(%)↑	MIA AUC(%)↓
Adult	0	59.89	51.42	73.25	50.97
	5	84.92	51.21	81.79	51.04
	10	84.97	51.32	82.34	51.14
	20	84.93	51.36	82.58	51.33

Table 3: Impact of the number of fine-tuning epochs on DenseNet12 trained on CIFAR-10 with $\epsilon = 0.5$

Dataset	Epochs	DenseNet12	
		Utility Acc(%)↑	MIA AUC(%)↓
CIFAR-10	0	16.42	49.23
	5	81.10	50.73
	10	85.72	50.58
	20	88.03	51.07

81.10% utility after just 5 epochs. This trend continues, with utility reaching 85.72% at 10 epochs and 88.03% at 20 epochs.

For all benchmarks, MIA AUC scores stay similar to those of M^{ϵ}, thereby indicating almost no impact of fine-tuning epochs on the forgetting performance.

2.4 Impact of the forgetting ratio

Table 4 reports the impact of the forgetting ratio on utility and runtime. Regarding utility, increasing the forgetting ratio led to a decrease of the AUC of the ML model retrained from scratch ($M^{O}_{\mathbf{D}_r}$) for both MLP and XGBoost. For SISA with MLP, there was almost no impact on utility, whereas utility slightly decreased for SISA with XGBoost. EUPG with MLP ($k = 10$ and $\epsilon = 0.5$) showed remarkable resilience, maintaining high utility scores even when the forgetting ratio reached 50%. With XGBoost and $k = 10$ the utility slightly decreased as the forgetting ratio increased, whereas with $\epsilon = 0.5$, performance was poor already for low forgetting ratios, and it deteriorated further as the forgetting ratio increased.

The runtime tended to decrease with increasing forgetting ratios for retraining from scratch and SISA, due to the lower training data size requiring less computational effort. EUPG also showed large reductions of its already very short runtime with increasing forgetting ratios.

Table 4: Impact of the forgetting ratio with the Credit dataset

Method	Forgetting ratio	MLP		XGBoost	
		Utility	AUC(%)↑ RT (s)↓	Utility	AUC(%)↑ RT (s)↓
$M^{O}_{\mathbf{D}_r}$	5%	71.96	448.61	80.24	3.68
	10%	71.46	402.03	80.12	3.60
	20%	70.92	383.03	79.76	3.52
	50%	67.11	236.01	79.55	3.25
$M^{SISA}_{\mathbf{D}_r}$	5%	76.12	179.81	83.33	7.19
	10%	76.20	160.09	81.37	7.60
	20%	76.52	146.06	80.91	7.59
	50%	76.20	94.38	79.81	7.24
$M^{k=10}_{\mathbf{D}_r}$	5%	81.78	10.56	82.24	0.33
	10%	81.87	10.32	82.01	0.30
	20%	81.95	9.73	81.97	0.26
	50%	81.61	6.17	81.74	0.19
$M^{\epsilon=0.5}_{\mathbf{D}_r}$	5%	81.66	11.45	60.71	0.32
	10%	81.89	10.32	60.34	0.29
	20%	81.89	9.53	56.84	0.26
	50%	81.92	6.21	52.52	0.18

3 Detailed runtime analysis

Table 5 reports the runtimes of the k-anonymization process to obtain $\mathbf{D}^k$, train M^k, and fine-tune M^k on $\mathbf{D}$ with different values of k. Note that the anonymization time decreases almost linearly as k increases, demonstrating that larger values of k reduce the anonymization effort due to the more uniform (aggregated) training data. On the other hand, the training and fine-tuning times for both MLP and XGBoost models remain relatively stable across k values. This indicates that the dominant factor in runtime is the initial k-anonymization process rather than the subsequent training or fine-tuning steps.

Tables 6 and 7 report runtimes for DP anonymization on tabular and image data, respectively. Runtimes are given for the embedding process (for categorical attributes), for creating $\mathbf{D}^\epsilon$, for training M^ϵ, and for fine-tuning M^ϵ on $\mathbf{D}$ with different values of ϵ. In our experiments, we only required embeddings for the Adult dataset. As expected, the runtime for generating ϵ-differentially private datasets ($\mathbf{D}^\epsilon$) remained stable across different ϵ values, as did the runtimes for training and fine-tuning all models. This stability suggests that the computational cost of implementing differential privacy through ϵ adjustments is almost invariant to the choice of ϵ. An important cost is the initial data embedding or transformation process needed for datasets with semantically-rich categorical attributes.

Table 5: Runtime in seconds when using k-anonymity with different values of k

Dataset	k	k-Anonymizing $\mathbf{D}$	Training M^k		Fine-tuning M^k on $\mathbf{D}$	
			MLP	XGBoost	MLP	XGBoost
Adult income	3	161.97	93.67	5.06	4.93	0.41
	5	101.97	94.56	5.73	4.99	0.44
	10	50.59	94.13	5.13	4.86	0.42
	20	25.83	94.90	4.51	4.80	0.39
	80	6.94	94.31	2.88	4.92	0.38
Heart disease	3	90.69	195.12	2.33	5.08	0.23
	5	57.50	193.46	2.40	5.06	0.24
	10	28.90	193.18	2.33	4.98	0.24
	20	14.57	193.24	2.24	5.08	0.25
	80	3.67	195.13	1.26	5.07	0.19
Credit	3	356.47	440.86	3.48	11.44	0.47
	5	162.00	442.65	4.07	11.41	0.48
	10	102.83	442.26	3.33	11.17	0.51
	20	40.56	442.50	4.00	11.35	0.50
	80	10.24	442.58	3.34	11.30	0.50

Table 6: Runtime in seconds when using DP with different ϵ values on tabular data

Dataset	ϵ	Embedding Generating $\mathbf{D}^\epsilon$	Training M^ϵ		Fine-tuning M^ϵ on $\mathbf{D}$	
			MLP	XGBoost	MLP	XGBoost
Adult income	0.5	167.43	21.17	93.97	6.52	4.60	0.42
	2.5		20.92	93.88	6.65	4.79	0.43
	5		20.90	94.69	6.79	4.60	0.42
	25		21.05	94.56	6.42	4.67	0.43
	50		20.90	94.55	6.41	4.91	0.43
	100		19.88	94.51	6.51	5.11	0.43
Heart disease	0.5	0	28.92	193.39	2.16	5.08	0.25
	2.5		28.94	194.82	2.37	5.18	0.25
	5		29.15	193.69	2.23	5.19	0.26
	25		28.88	193.83	2.36	4.96	0.27
	50		29.11	194.05	2.35	5.10	0.25
	100		29.01	195.07	2.45	5.05	0.28
Credit	0.5	0	43.43	442.56	3.79	12.03	0.32
	2.5		43.21	441.41	3.37	11.80	0.44
	5		43.29	440.94	3.89	11.59	0.44
	25		43.91	442.30	4.16	11.58	0.49
	50		43.62	442.15	4.20	11.45	0.48
	100		43.49	442.03	3.91	11.35	0.46

Table 7: Runtime in seconds when using DP with different ϵ values on image data

Dataset	ϵ	Embedding Generating $\mathbf{D}^\epsilon$	Training M^ϵ DenseNet12	Fine-tuning M^ϵ on $\mathbf{D}$ DenseNet12
CIFAR-10	0.5	0	95.41	28979.72	1662.23
	2.5		96.63	29028.00	1662.07
	5		96.05	29021.83	1665.11
	25		96.50	28939.69	1658.16
	50		96.32	29066.90	1707.61
	100		96.02	30044.02	1678.08

References

1. Arik, S.Ö., Pfister, T.: Tabnet: Attentive interpretable tabular learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 6679–6687 (2021)
2. Cao, J., Liu, B., Wen, Y., Zhu, Y., Xie, R., Song, L., li, L., Yin, Y.: Hiding among your neighbors: face image privacy protection with differential private k-anonymity. In: Proceedings of the 2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting. pp. 1–6 (2022)
3. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794 (2016)
4. Domingo-Ferrer, J., Torra, V.: Ordinal, continuous and heterogeneous k-anonymity through microaggregation. Data Mining and Knowledge Discovery **11**, 195–212 (2005)
5. Dwork, C.: A firm foundation for private data analysis. Communications of the ACM **54**(1), 86–95 (2011)
6. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3. pp. 265–284. Springer (2006)
7. Fan, L.: Differential privacy for image publication. In: Theory and Practice of Differential Privacy (TPDP) Workshop. vol. 1, p. 6 (2019)
8. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
9. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07). pp. 94–103. IEEE (2007)
10. Mygdalis, V., Tefas, A., Pitas, I.: Introducing k-anonymity principles to adversarial attacks for privacy protection in image classification problems. In: In 2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP). pp. 1–6 (2021)
11. Nakamura, T., Sakuma, Y., Nishi, H.: Face-image anonymization as an application of multidimensional data k-anonymizer. International Journal of Networking and Computing **11**(1), 102–119 (2021)