UAE-PUPET: AN UNCERTAINTY-AUTOENCODER-BASED PRIVACY AND UTILITY PRESERVING END-TO-END TRANSFORMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a new framework that deals with the privacy-utility tradeoff problem under two centralized settings: a dynamic setting and a constant setting. The dynamic setting corresponds to the min-max two-player game whereas the constant setting corresponds to a generator which tries to outperform an adversary already trained using ground truth data. In both settings, we use the same architecture consisting of a generator and a discriminator, where the generator consists of an encoder-decoder pair, and the discriminator consists of an adversary and a utility provider. Unlike previous research considering this kind of architecture, which leverage variational autoencoders (VAEs) based on learning a latent representation which is forced into a Gaussian assumption, our proposed technique removes the Gaussian assumption restriction on the latent variables, and only focuses on the end-to-end stochastic mapping of the input to privatized data. We also show that testing the privacy mechanism against a single adversary is usually not sufficient to capture the leakage of private information, as better adversaries can always be created by training under different conditions. Therefore, we test our proposed mechanism under five different types of adversary models. To compare privacy mechanisms under a fair framework, we propose a new metric called the Utility-Privacy Tradeoff (UPT) curve, obtained by using the upper convex hull of the utility-privacy tradeoff operation points achievable under the most powerful of the five adversary models. Finally, we test our framework on four different datasets: MNIST, Fashion MNIST, UCI Adult and US Census Demographic Data, providing a wide range of possible private and utility attributes. Through comparative analysis, our results show better privacy and utility guarantees, under our more rigorous adversary model, than the existing works, even when the latter are considered under their original restrictive single-adversary models.

1 INTRODUCTION

With the recent surge in the usage of online services, we share an increasingly large amount of data with different third-party service providers in order to receive some form of utility. Even when we pay special attention to avoid disclosing what we deem to be private information, such as our identity, age, race, location, gender, income, medical conditions, political views etc., the data that we do share may still contain an uncomfortably large amount of information about these attributes – an amount that may be just enough for intruders to infer our private data. It is therefore important to be aware of such potential data correlations, and to employ a privacy mechanism that can essentially destroy the information between the shared data and the private data, while preserving the useful information – that required to achieve the desired level of utility. We call such a mechanism a *privacy and utility preserving end-to-end transformation (PUPET)*.

Ideally, a PUPET should minimize the leakage of information about the private attributes, and maximize the information that the shared data contains about the utility attributes. The operational point of the mechanism, establishing a point in some private-information-utility-information plane, is said to achieve a certain privacy-utility tradeoff. Most notions of privacy, such as Differential Privacy (Dwork (2006); Dwork & Roth (2014)), or k-anonymity (Sweeney, 2002) are achieved by using some sort of distortion, such as adding random noise to the data, or performing data su-

pression or generalization. The privacy-utility tradeoff is the subject of a large body of research, with many works focused on producing application-specific solutions for the problem (Rajagopalan et al. (2011); Alvim et al. (2012); Makhdoumi & Fawaz (2013); Sankar et al. (2013); Sharma et al. (2021)). (Domingo-Ferrer & Torra, 2008) shows the drawbacks of k-anonymity and its variant. Similarly, it is known that computing the optimal noise addition in higher dimensional data for differential privacy can potentially be infeasible. Therefore, a wide majority of recent techniques use neural networks, which can deal with the high dimensional data and can provide an approximation of the underlying functions. In particular, research works such as (Edwards & Storkey (2016); Huang et al. (2017); Madras et al. (2018); Huang et al. (2019); Chen et al. (2019); Erdemir et al. (2021)) have been carried out leveraging variational autoencoder (VAEs) (Kingma & Welling, 2014) and/or generative adversarial network (GAN) type training (Goodfellow et al., 2014). VAEs are used to create compressed latent representations capturing maximum and minimum information about the private and utility features, respectively. However, VAEs need to model explicitly the prior distribution over the latent variable. Their loss functions include a regularization term to minimize the KL divergence between the variational posterior over the latent variable and some prior distribution over the latent variable (which, for the convenience of generating data in the absence of an encoder, is usually chosen to be white Gaussian). GANs, on the other hand, involve a min-max game between the generator and discriminator. Similarly, recent obfuscation mechanisms (Ilanchezian et al. (2019); Hsu et al. (2020)) demonstrate privacy preservation by first estimating each feature's information density relative to the private data, and then selectively distorting the most information-leaking features. All these strategies require either training a filter that adds noise, or adding noise directly into the input stream while training, or realizing distortions based on the loss function or some other form of masking technique. Although the strategies around generating private data are diverse, the approaches used to test such mechanisms still remain trivial i.e. they are tested against a single and particular type of an adversary network, which is usually trained using ground truth data, or dynamically trained as in GAN-type training.

In this paper, we show that adversaries employing training techniques different from the ones used by the privacy mechanism can possibly infer private data in a much more efficient way than adversaries using the same training technique as the ones used to train the PUPET. Therefore, existing privacy mechanisms that claim a certain effectiveness for their model against a single adversary might not be robust enough against other adversaries – hence leading to more leaked private information than originally expected. To overcome this shortcoming, we propose a new privacy mechanism and test it on a total of five *differently trained adversary models*. In our proposed privacy mechanism, we remove the Gaussian assumption on the prior distribution of the latent variable (like in the case of VAEs) and only focus on the end-to-end stochastic mapping to transform data that preserves privacy and utility. The Gaussian assumption used by VAEs is useful for ancestral sampling, which is not required in the generation of private data. Instead, we require a Markov chain $(X \rightarrow Z \rightarrow \hat{X})$ where X is the input data, Z is the latent variable and $\hat{X}$ is the private data. To implement our privacy mechanism, we leverage Uncertainty Autoencoders (UAEs) (Grover & Ermon, 2019), which define an implicit generative model without specifying a prior on the latent representation. The use of a Markov chain to generate private data is necessary regardless of the generative model (VAE or UAE) and thus, the use of UAE doesn't make the process any more computationally expensive than previous approaches. In our setting, the UAE behaves as a generator, and the discriminators (adversary and utility provider) provide a regularization term that reflects the cost of ensuring protection of private features and disclosure of utility ones, respectively.

To compare privacy mechanisms under a fair framework, we propose a new metric called the Utility-Privacy Tradeoff (UPT) curve. The UPT curve is a graph plotting the operational points of accuracy in inferring the utility feature vs. accuracy in inferring the private feature, for different system parameter settings, and then finding the upper convex hull (Andrew, 1979) of these operational points. It is worth noting that the accuracy in inferring the private feature is the one achievable under the most powerful of the five adversary models considered.

To showcase the effectiveness of our proposed privacy mechanism as well as the necessity of testing against multiple adversary models, we performed comprehensive experiments on four datasets viz. MNIST handwritten digits (LeCun & Cortes, 2010), Fashion MNIST (Xiao et al., 2017), UCI Adult (Dua & Graff, 2017), and US Census Demographic Data (MuonNeutrino, 2017) for both the constant and dynamic settings. We demonstrate that our constant setting can be used in areas where the desired goal is to outperform a given adversary. This kind of setting may be very limited in real life

and therefore, the majority of our work focuses on the dynamic setting where our proposed model considers five different types of adversary models and is shown to provide us better privacy and utility guarantees than any of the previously existing methods.

For each dataset, we pick different sets of private and utility features. In particular, we directly compare our proposed UAE-based PUPET (or *UAE-PUPET*) mechanism to that of (Chen et al., 2019), in the same setting as in (Chen et al., 2019), using the MNIST Case2 (variant) database, where the private attribute encodes whether a number is odd or even, while utility attribute encodes whether a number is ≥ 5. We show that our mechanism attains 4.2% lesser accuracy for the private feature, and 11.5% higher accuracy for the utility attribute, thus clearly outperforming the previously existing mechanism. It is worth mentioning that this result was obtained by our UAE-PUPET under the best performing adversary out of five, which shows that our mechanism is more robust than previous works considering their original restrictive single adversary model.

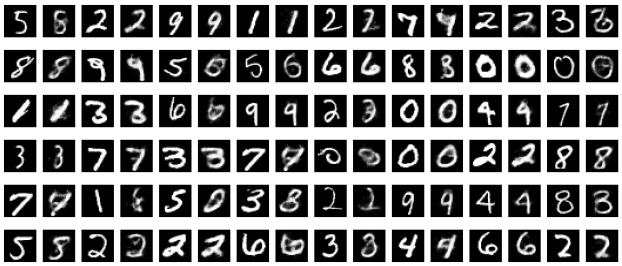


Figure 1: **MNIST Case 2**: Odd and even adjacent columns show original and privatized versions respectively. For most images, numbers are still in the same category (utility attribute: ≥ 5 or < 5) while being switched from odd to even (private attribute). Some digits change from odd to even but also switch from ≥ 5 to < 5, and some remain unchanged.

The rest of the paper is organized as follows. The problem formulation is detailed in Section 2, experimental results are given in Section 3, and concluding remarks are drawn in Section 4.

2 Problem Formulation and Methodology

Consider a setting where a user wishes to release some data vector X with the intent to receive certain level utility, while maintaining a certain level of privacy about a specific feature or set of features. We represent the private feature vector as X_P and the utility feature vector as X_U, and expect that they are both correlated with X. Some examples of possible features include identity, age, race, location, gender, income, medical conditions, political views, like/dislike on a content etc. To ensure the desired privacy and utility guarantees, before publicly sharing their data X, users employ a PUPET that takes X as input, and generates $\hat{X}$, a distorted version of X which contains minimum information about X_P and maximum information about X_U. the data $\hat{X}$ is then shared publicly.

It is very important to note here that the disclosed data $\hat{X}$ is usually required to preserve in general the size and structure of X. That is, while it may be tempting to devise a compression mechanism – for example, through a simple arbitrary affine transformation – that produces a shorter $\hat{X}$, each component of which is some affine transformation of the components of X, such a mechanism has very little applicability in practice. This is because usually the disclosure of $\hat{X}$ has to take place over a pre-existing platform, outside of the data owner's control, which is designed specifically for X. This platform usually has very specific fields that the user needs to fill out, so that the structure of X is enforced on the PUPET's output. It is also worth noting that most currently existing platforms (such as social media platforms) would provide their utility by taking the values of $\hat{X}$ at face value, as if it was the original X that was being disclosed, which motivates some of the existing privacy-utility tradeoff works to use the distance between $\hat{X}$ and X as a measure of utility Asoodeh et al. (2015); Erdogdu & Fawaz (2015); Wang & Calmon (2017); Kalantari et al. (2017); Basciftci et al. (2016); Wang et al. (2018); Rassouli & Gündüz (2019); Diaz et al. (2019). However, unlike these works, we consider a smart and informed utility provider, who is aware of the privacy mechanism employed by the user – of course, without knowing the exact realizations of the randomness it uses, and can make a competent inference about the utility feature, using a neural-network-based architecture. Relating our setting to real world privacy problems, one of the many possible examples

could be a setting where users are required to fill in certain information when creating e-commerce accounts. These e-commerce accounts gather the information we share and may want to develop implicit models to recognize our gender and income. However, users might not be comfortable knowing that these implicit models learn about their income, but at the same time would also want to get better recommendation of products based on their gender. In such a case, the users can use a PUPET which generates $\hat{X}$, and then use $\hat{X}$ to fill up the information to create an account. The training of the PUPET requires multiple tuples (X, X_P, X_U), collected from users who do not mind disclosing $X_P,\ X_U$, and can be handled either by the data owner, or by a trusted service provider.

Formally, let $X^j = \{x_1^j, x_2^j, x_3^j, \cdots x_{n^j}^j\}$, where the components x_i^j are all correlated random variables denoting n distinct features of the user U^j, which the user wishes to release to the public. In addition, the user U^j consists of private features X_P^j, and utility features X_U^j with n_p^j and n_u^j component random variables respectively. Note that, X_P^j and X_U^j are both correlated with X^j and no private and utility feature is in X^j i.e. $X_P^j \notin X^j$ and $X_U^j \notin X^j$. For different users, the choice of the data they wish to share, private features and utility features differ, and thus our privacy mechanism needs to be trained differently for different sets of users. For simplicity we drop the user-specific indices and refer to the random vectors directly as X, X_P and X_U. We represent the PUPET in its most general form as a function f (which could be a randomized mapping) that takes input (X, X_P, X_U) and generates $\hat{X}$ i.e. $\hat{X} = f(X, X_P, X_U)$. The adversary builds a learning algorithm a_p that takes privatized data $\hat{X}$ to infer the private attributes X_P' which is an estimate of X_P i.e. $X_P' = a_p(\hat{X})$. The goal of adversary is to minimize loss between X_P' and X_P i.e. $l_P(X_P', X_P) = l_P(a_p(f(X, X_P, X_U)), X_P)$. Correspondingly, the utility provider builds a learning algorithm a_u to infer X_U' which is an estimate of X_U and desires to minimize the loss $l_U(X_U', X_U) = l_U(a_u(f(X, X_P, X_U)), X_U)$. The privacy mechanism f is now chosen to maximize the inference loss l_P and minimize the inference loss l_U. This setting refers to the min-max game and can be expressed as follows:

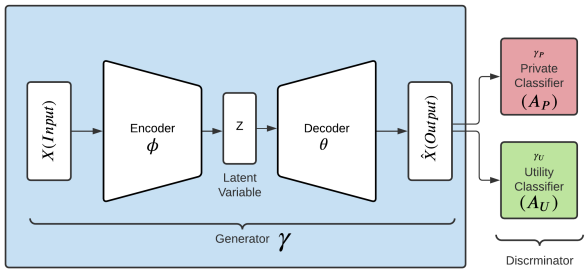


Figure 2: **UAE-PUPET** architecture. This architecture supports both dynamic and constant setting. After the completion of training, we detach the discriminator, and use the generator to generate private data.

$$\max_{f \in \mathcal{F}} \left\{ \lambda_P \min_{a_p \in \mathcal{A}_p} \mathbb{E}\left[l_P\left(a_p\left(f\left(X, X_P, X_U\right)\right), X_P\right)\right] - \min_{a_u \in \mathcal{A}_u} \mathbb{E}\left[l_U\left(a_u\left(f\left(X, X_P, X_U\right)\right), X_U\right)\right] \right\},$$

where λ_P is a hyperparameter that controls the tradeoff between adversary loss and utility provider loss, and the expectation is taken over all samples of the dataset, and $\mathcal{F}$, $\mathcal{A}_p$ and $\mathcal{A}_u$ are the sets of functions from which the privacy mechanism, the adversary and the utility provider select their corresponding operators, respectively. In this paper we shall use neural networks to optimize over different functions f, a_p, a_u and the loss l_P is taken as the cross-entropy (de Boer et al., 2004). This is standard for most classification problems. We also note that cross-entropy loss affects the mutual information (Chen et al., 2019) between two random variables, and this supports our original goal to reduce correlation with the private features and maintain similar correlation with the utility features. To solve the optimization problem, we propose two settings: a dynamic setting and a constant setting, which are both described in the subsections below.

2.1 Dynamic Setting (Joint Training)

In order to solve the optimization problem above, we leverage an uncertainty autoencoder (UAE), which serves as the generator for our privacy mechanism. The objective of UAE is given by $\max_{\theta,\phi} \mathbb{E}_{Q_\phi(X,Z)}\left[\log p_\theta(x|z)\right]$, where X is the input data distribution, Z is the latent variable, ϕ, θ are parameters of encoder and decoder, respectively, $Q_\phi(X, Z)$ is the true joint distribution of the

input and latent representation, and $p_\theta(X|Z)$ is the posterior distribution produced by the decoder, which aims to emulate $Q_\phi(X|Z)$ as closely as possible.

Notice that we don't force a Gaussian assumption on the latent variable prior, but instead focus on end-to-end stochastic mapping. In our joint setting, we introduce the parameter γ, which represents parameters of generator (encoder-decoder pair) with function f (basically, $\gamma = (\phi, \theta)$). Additionally, the output of the generator is attached to the discriminator, which consists of an adversary and a utility provider as shown in Figure 2.

The adversary learns the function a_p with parameters γ_P to minimize the privacy-specific loss $l_P(X'_P, X_P)$. Similarly, the utility provider learns the a_u with parameters γ_U to minimize the utility-specific loss $l_U(X'_U, X_U)$. Conversely, the generator function f with parameter γ learns to maximize $l_P(X'_P, X_P)$ and minimize $l_U(X'_U, X_U)$ respectively along with the objective of UAE. In this setting, the neural network parameters $\gamma, \gamma_P, \gamma_U$ all are trained together to solve the following optimization problem:

$$\max_{\gamma} \left\{ \mathbb{E}_{Q_\phi(X,Z)} \left[\log p_\theta(x|z) \right] + \lambda_P \min_{\gamma_P} \left\{ \mathbb{E} \left[l_P \left(X'_P, X_P \right) \right] \right\} - \min_{\gamma_U} \left\{ \mathbb{E} \left[l_U \left(X'_U, X_U \right) \right] \right\} \right\},$$

where $X'_P = a_p(f(X, X_P, X_U))$ and $X'_U = a_u(f(X, X_P, X_U))$.

2.2 Constant Setting (Individual Training)

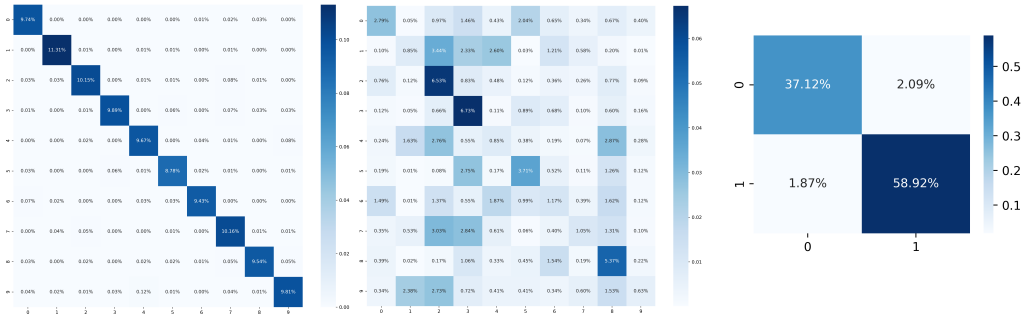
This setting considers a scenario where the desired goal is to outperform a given, a-priori established adversary. The architecture used for training the privacy mechanism is similar to the dynamic setting. Unlike joint training, the generator and discriminator are trained in two different phases. In the first phase, the adversary learns the function a_p with parameters γ_P, and utility provider learns the function a_u with parameters γ_U to minimize the private loss $l_P(X'_P, X_P)$, and utility loss $l_U(X'_U, X_U)$ respectively. It is important to note that a_p and a_u are trained using ground truth data and their respective labels. In the second phase of the privacy mechanism, the discriminator part is kept fixed, i.e., the parameters γ_P and γ_U are not updated. With respect to the fixed γ_P and γ_U, the generator's function f with parameters γ is then trained to solve the following optimization problem:

$$\max_{\gamma} \left\{ \mathbb{E}_{Q_\phi(X,Z)} \left[\log p_\theta(x|z) \right] + \lambda_P \left\{ \mathbb{E} \left[l_P \left(X'_P, X_P \right) \right] \right\} - \left\{ \mathbb{E} \left[l_U \left(X'_U, X_U \right) \right] \right\} \right\},$$

where $X'_P = a_p(f(X, X_P, X_U))$ and $X'_U = a_u(f(X, X_P, X_U))$.

3 Experiments and Results

We perform comprehensive experiments using the machine learning library Tensorflow (Abadi et al., 2015), Keras (Chollet et al., 2015) and optimizer Adam (Kingma & Ba, 2017) on four widely-used datasets such as MNIST, Fashion MNIST, UCI-adult, and US Census Demographic Data, to demonstrate the effectiveness of our privacy mechanism. In order to test our mechanism we first develop five different adversary and utility models for each of the different dataset experiments we conduct. The models are chosen as follows. (1) For the constant setting, the first phase is to train a_u and a_p using ground truth data. The pair (a_u, a_p) that is used to train the generator is used to define first attacker and utility provider model pair. (2) If we train another pair (a_u, a_p) using ground truth data similar to the first scenario above, we usually obtain different sets of weights (parameters) – this is because we allow random initialization of weights. This is how we produce the second attacker and utility provider model pair. (3) The third attacker and utility provider model pair is produced by training a (a_u, a_p) pair using the distorted data $(\hat{X})$ which is generated from the *constant setting* and a pre-defined λ_P value. The training labels used are still from the corresponding ground truth data. The pre-defined λ_P value is selected based on our experiments, such that the adversary and utility provider can capture the notion of distortion in the data. More details about the architecture of the discriminators is provided in the Appendix A. (4) The fourth attacker and utility provider model pair is produced by training a (a_u, a_p) pair using the distorted data $(\hat{X})$ which is generated from the *dynamic setting* and a pre-defined λ_P value. The training labels used are still from the corresponding ground truth data. (5) The fifth model pair is produced by a separate joint training process, managed by the attacker or utility provider, and using ground-truth data. The training uses



(a) private (raw) (b) private (UAE-PUPET) (c) utility (UAE-PUPET)

Figure 3: **MNIST Case 1:** Confusion matrix before and after privacy mechanism

a pre-defined λ_P value. The generator is only relevant to the training of the (a_u, a_p) pair, and is subsequently discarded. (6) The sixth model pair, which is only used when the UAE is trained in the joint setting, consists of the exact (a_u, a_p) pair that results from the joint training.

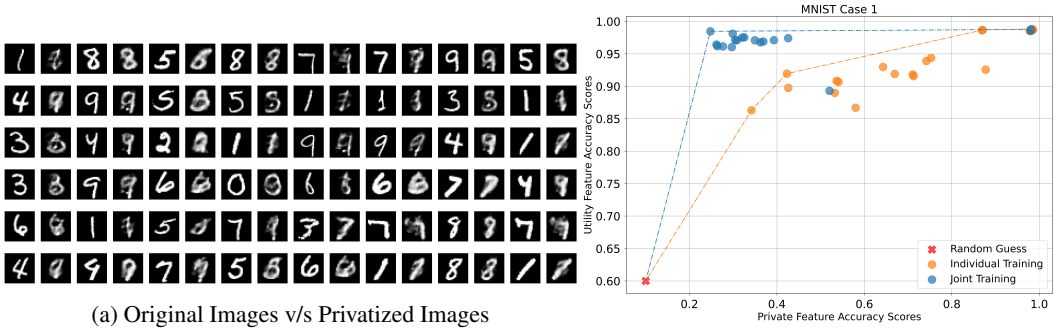
Utility Privacy Tradeoff (UPT) curve: The UPT curve is a concave curve on the graph plotting the operational points of the accuracy in inferring the utility feature vs. the accuracy in inferring the private feature (hence forth the *utility-privacy graph*), for different system parameter settings. The UPT curve is used to represent the upper bound on the performance of the privacy-utility mechanism. As such, it consists of the upper convex hull of all the achieved operational points, under various values of the system hyperparameters. The operational interpretation of the upper convex hull relies on an operational interpretation of any line connecting two operational points. Recall that each operational point is defined by two accuracy levels, achieved by averaging over an entire test dataset. If we split the test dataset in two parts of sizes α and $1-\alpha$ times the original size, respectively, and apply the first part to the mechanism achieving operational point P_1, and the second part to the mechanism achieving P_2, then the average over the entire dataset should achieve operational point $\alpha P_1 + (1-\alpha)P_2$. It is in this sense that the upper convex hull is achievable.

For our experiments, all the points on the utility-privacy graphs are achieved by selecting the highest accuracy obtained over the different adversaries and utility provider models, for some privatized data $\hat{X}$ which was generated by the privacy mechanism under some value of λ_P. For example, if we generate privatized data $\hat{X}$ using a certain λ_P value, and consider a total of six different adversary and six different utility provider models, we get six accuracy scores for the adversary and six accuracy scores for the utility provider. We select the highest of the six accuracy scores for the inference of the private feature, and also the highest of the six accuracy scores for the inference of the utility feature. This pair forms one point in the utility-privacy graph. repeating the process for a different λ_P yields a different point, and so forth. Points in the north-west region of the graph are preferable. In all our graphs, we take a *trivial-classifier* point as achievable by default – this represents the accuracy levels that could be achieved by always deciding in favor of the most highly represented class, and depends on the specification of the private and the utility features, as well as on the composition of the test dataset. For example, if the private feature lives on an alphabet of size 2, and the test dataset contains 60% points from class 1, and 40% from class 2, then always deciding in favor of class 1 produces an accuracy of 0.6. The trivial-classifier points will appear in the lower left portions of all our graphs.

MNIST Case 1: This case of MNIST considers similar setting as (Chen et al., 2019) where the identity of the digit $(0, 1, 2, 3 \cdots, 9)$ is considered private, while the utility encodes whether the digit contains a circle $(0, 6, 8, 9)$ or not $(1, 2, 3, 4, 5, 7)$. We test our privatization scheme based on different hyperparameter values of λ_P for both settings, i.e., dynamic and constant. Each point

Table 1: MNIST Case 1 accuracy and F1 scores

Models	**Private attr.**		**Utility attr.**	
	accuracy	F1 score	accuracy	F1 score
without distortion (raw)	0.98	0.98	0.98	0.98
UAE-PUPET	**0.29**	**0.26**	**0.9604**	**0.967**

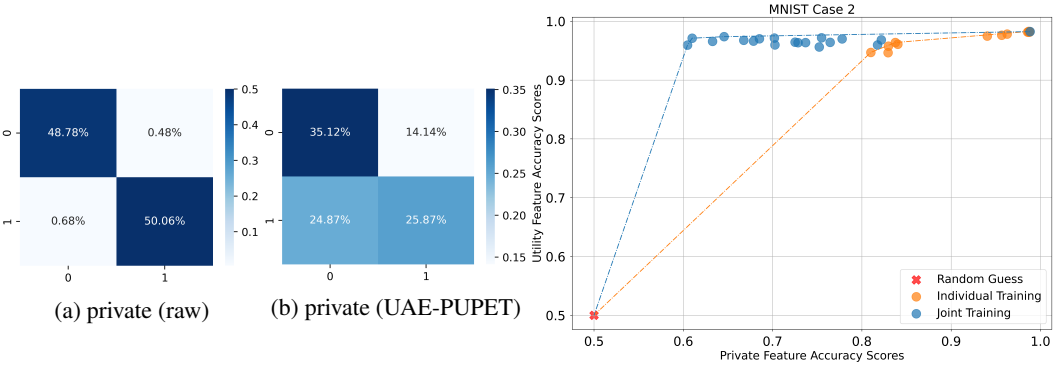


(a) Original Images v/s Privatized Images

(b) UPT curve (Accuracy Scores)

Figure 4: **MNIST Case 1**: (4a) Odd and even adjacent columns show original and privatized versions respectively (generated by joint training). Private feature refers to digit identity and utility attribute refers to whether a number contains a circle or not. It is interesting to see that some privatized data looks like a combination of more than one digits masked together, making it harder to infer the digit identity. (4b) shows the robustness of joint training against multiple adversaries and a performance comparison of the two settings.

on the utility-privacy graph in Figure 4b represents the best achieved accuracy for a particular λ_P value, over five different a_p and a_u models (in case of joint training, there are six a_u and a_p). With an increase in λ_P, the accuracy of the inference of the private feature decreases, with only a minimal change to the accuracy of the inference of the utility attribute. The belts of blue and orange points represent the best achievable points for joint training and individual training, respectively. It is clear that joint training outperforms individual training. Table 1 shows that the accuracy on private labels decreases from 98% to 29% on the best performing adversary, which in this case turns out to be the a_p from Model 4 above, while the best performing a_u is capable of capturing utility attributes with 96.7% accuracy after the privatization using the dynamic setting. Figure 3 shows the confusion matrix before and after privatization, which further supports our argument. Similarly, Figure 4a shows the private images that were generated using the dynamic setting with $\lambda_P = 175$. Individual training, despite not being as robust as Joint Training, still performs well, by reducing the accuracy of private attributes to 18.3% and keeping utility accuracy to 91.57%, both referring to Model 1 (a_u, a_p).



(a) private (raw)

(b) private (UAE-PUPET)

(c) UPT curve (Accuracy scores)

Figure 5: **MNIST Case 2**: (5a) shows the confusion matrix prior to distortion and (5b) shows confusion matrix post distortion. (5c) shows upper bound performance of two settings.

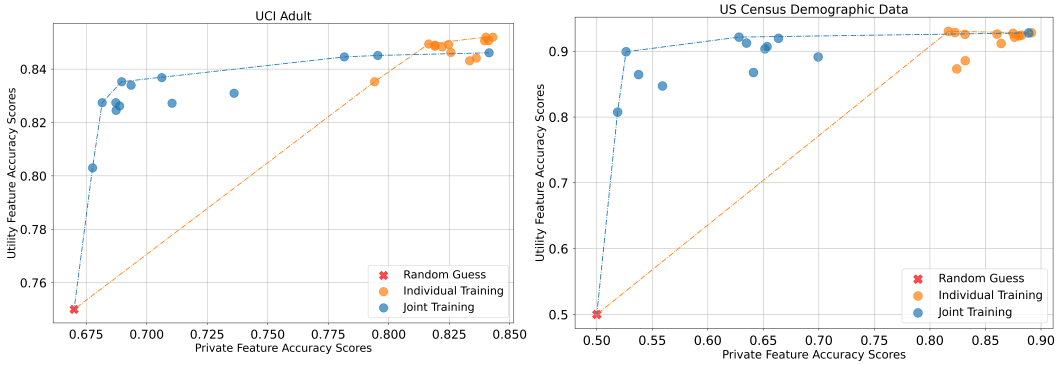
MNIST Case 2:

This case considers the private attribute as whether the number is odd or even, while the utility attribute encodes whether the number is ≥ 5 or not. Figure 1 shows the original and privatized images generated by the dynamic setting with $\lambda_P = 30$. Table 2 shows the accuracy result of joint training under hyperparameter $\lambda_P = 30$ and its comparison to emb-g-filter of (Chen et al., 2019).

Table 2: MNIST Case 2 accuracy results

Model	Private attr. (acc)	Utility attr. (acc)
without distortion (raw)	0.98	0.98
emb-g-filter (Chen et al., 2019)	0.651	0.855
UAE-PUPET	**0.609**	**0.97**

Our proposed mechanism outperforms the existing method by achieving 4.2% smaller accuracy on the private attribute and 11.5% higher accuracy on the utility attribute. Similarly, Figure 5c shows that joint training performs much better than individual training. Also notice that considering only the Model 1 adversary, the private feature accuracy is 50%, whereas utility accuracy is maintained at 96%. However, when we test the same private data against multiple adversaries we find adversaries which can perform much better, hence collapsing the privacy guarantee made by (Chen et al., 2019) under this model.



(a) UPT curve (Accuracy scores)

(b) UPT curve (Accuracy scores)

Figure 6: (6a) Experiment results for **UCI adult**, (6b) Experiment results for **US Census Demographic Data**

UCI Adult: In this experiment, we set our private feature as gender, and utility feature as income. Data pre-processing steps include converting categorical variables to one-hot encoding and normalizing values based on their mean and standard deviation. Figure 6a shows that joint training is robust against multiple adversaries which is evident from the UPT curve reaching far in the north-west region. Similarly, we compare our joint training results to other existing works such as (Louizos et al. (2017); Song et al. (2018); Chen et al. (2019)) and it is evident through Table 3 that our proposed method has the least accuracy and AUROC scores for private features and comparable accuracy and AUROC scores for the utility feature.

Table 3: UCI Adult accuracy and AUROC result comparison with existing techniques

Models	Private attr.		Utility attr.	
	accuracy	AUROC	accuracy	AUROC
LFAE (Louizos et al., 2017)	0.802	0.703	**0.851**	0.761
LMFIR (Song et al., 2018)	0.728	0.659	0.829	0.741
emb-g-filter (Chen et al., 2019)	0.717	0.632	0.822	0.731
UAE-PUPET	**0.681**	0.52	0.8274	0.731

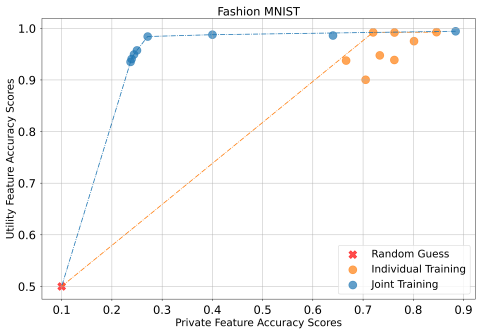
Fashion MNIST: We now consider a setup where the private feature is the identity of the fashion article (T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot) and the utility feature is encoded on two labels: Upper (meaning T-shirt/top, Pullover, Dress, Coat, Shirt) and Miscellaneous. Figure 7a shows the privatized images, along with their original versions. We

Table 4: Fashion MNIST accuracy and F1 scores

Models	Private attr.		Utility attr.	
	accuracy	F1 score	accuracy	F1 score
without distortion (raw)	0.98	0.98	0.99	0.99
UAE-PUPET	**0.24**	**0.20**	**0.95**	**0.95**

see that the privatized images appear to have been changed to different articles of clothing. We also notice blurriness of privatized images, in such a way that they appear sometimes to be comprised of two different images juxtaposed on one. Figure 7b shows the performance comparison of joint and individual training, while Table 4 shows a drop of inference accuracy on private feature from 98% to 24% whereas the inference accuracy on utility feature decrease slightly from 99% to 95%.

(a) Original images v/s Privatized images



(b) UPT curve (Accuracy scores)

Figure 7: **Fashion MNIST:** Figure 7a obtained from dynamic setting with hyperparameter $\lambda_P = 60$, where clothing identity is the private feature, while "upper body clothing" and "miscellaneous" are the two classes of the utility feature. Figure 7b shows the best achievable points for individual and joint training under different adversaries.

US Census Demographic Data: The American Community Survey data for 2017 consists of 74,001 records for different counties. It has a total of 37 features, out of which we select the sixteen features which are highly correlated to each other similar to the setting in (Sharma et al., 2021). Examples of some of the selected features include the men population, women population, population of citizens eligible to vote, per capita income, percentage of population unemployed etc. Among the sixteen features, we select *Employed* as the utility and *Income* as the private feature. We further categorize the utility feature into two labels i.e. ≤ 2000 or > 2000 and private feature into two labels i.e. ≤ 55000 or > 55000 to make the dataset balanced. All fourteen features are numerical, and thus we normalize them based on the mean and standard deviation. Similarly, some data points have missing values. In such case the entire data point was ignored. We use a total of 43,657 data points for training, and 29,105 data points for testing purposes. The UPT curve is given in Figure 6b. Table 5 shows that the accuracy for private features drops down from 88.7% to 52% and the utility accuracy drops ever so slightly from 92% to 90%.

Table 5: US Census Demographic Data Accuracy results

Models	**Private attr.**		**Utility attr.**	
	accuracy	AUROC	accuracy	AUROC
without distortion (raw)	0.887	0.885	0.92	0.925
UAE-PUPET	**0.52**	**0.52**	**0.90**	**0.899**

4 Conclusion

In this paper, we introduced a novel UAE-based privacy mechanism (UAE-PUPET), and showed that it can attain better privacy-utility tradeoffs than the existing works. This implies that forcing a Gaussian distribution on the latent variable of autoencoders (such as in VAE-based privacy mechanisms) appears to hinder, rather than help, the privacy mechanism. We emphasized the importance of testing the privacy mechanism against multiple adversaries to provide better privacy guarantees. To compare different privacy mechanisms under a fair framework, we propose to use a new metric called the Utility-Privacy Tradeoff (UPT) curve, which is the upper convex hull of the set of best achievable accuracies for private and utility inference, under various hyperparameters.
For more details about the architecture of generators, discriminators, different λ_P values used for multiple experiments please refer to our source code in the following repository : `https://anonymous.4open.science/r/abc-6BAC/README.md`

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Mário S. Alvim, Miguel E. Andrés, Konstantinos Chatzikokolakis, Pierpaolo Degano, and Catuscia Palamidessi. Differential privacy: On the trade-off between utility and information leakage. *Formal Aspects of Security and Trust*, pp. 39–54, 2012. ISSN 1611-3349. doi: 10.1007/978-3-642-29420-4_3. URL `http://dx.doi.org/10.1007/978-3-642-29420-4_3`.

Alex M. Andrew. Another efficient algorithm for convex hulls in two dimensions. *Inf. Process. Lett.*, 9:216–219, 1979.

Shahab Asoodeh, Fady Alajaji, and Tamás Linder. On maximal correlation, mutual information and data privacy. In *2015 IEEE 14th Canadian Workshop on Information Theory (CWIT)*, pp. 27–31. IEEE, 2015.

Yuksel Ozan Basciftci, Ye Wang, and Prakash Ishwar. On privacy-utility tradeoffs for constrained data release mechanisms. In *2016 Information Theory and Applications Workshop (ITA)*, pp. 1–6. IEEE, 2016.

Xiao Chen, Thomas Navidi, Stefano Ermon, and Ram Rajagopal. Distributed generation of privacy preserving data with user customization, 2019.

François Chollet et al. Keras. `https://keras.io`, 2015.

Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *ANNALS OF OPERATIONS RESEARCH*, 134, 2004.

Mario Diaz, Hao Wang, Flavio P Calmon, and Lalitha Sankar. On the robustness of information-theoretic privacy measures and mechanisms. *IEEE Transactions on Information Theory*, 66(4): 1949–1978, 2019.

Josep Domingo-Ferrer and Vicenç Torra. A critique of k-anonymity and some of its enhancements. In *2008 Third International Conference on Availability, Reliability and Security*, pp. 990–993, 2008. doi: 10.1109/ARES.2008.97.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener (eds.), *Automata, Languages and Programming*, pp. 1–12, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-35908-1.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014. ISSN 1551-305X. doi: 10.1561/0400000042. URL `https://doi.org/10.1561/0400000042`.

Harrison Edwards and Amos Storkey. Censoring representations with an adversary, 2016.

Ecenaz Erdemir, Pier Luigi Dragotti, and Deniz Gunduz. Active privacy-utility trade-off against a hypothesis testing adversary, 2021.

Murat A Erdogdu and Nadia Fawaz. Privacy-utility trade-off under continual observation. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 1801–1805. IEEE, 2015.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

Aditya Grover and Stefano Ermon. Uncertainty autoencoders: Learning compressed representations via variational information maximization, 2019.

Hsiang Hsu, Shahab Asoodeh, and Flavio Calmon. Obfuscation via information density estimation. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 906–917. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/hsu20a.html.

Chong Huang, Peter Kairouz, Xiao Chen, Lalitha Sankar, and Ram Rajagopal. Context-aware generative adversarial privacy. *Entropy*, 19(12), 2017. ISSN 1099-4300. doi: 10.3390/e19120656. URL https://www.mdpi.com/1099-4300/19/12/656.

Chong Huang, Peter Kairouz, Xiao Chen, Lalitha Sankar, and Ram Rajagopal. Generative adversarial privacy, 2019.

Indu Ilanchezian, Praneeth Vepakomma, Abhishek Singh, Otkrist Gupta, G. N. Srinivasa Prasanna, and Ramesh Raskar. Maximal adversarial perturbations for obfuscation: Hiding certain attributes while preserving rest, 2019.

Kousha Kalantari, Lalitha Sankar, and Oliver Kosut. On information-theoretic privacy with general distortion cost functions. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 2865–2869. IEEE, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The variational fair autoencoder, 2017.

David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations, 2018.

A. Makhdoumi and N. Fawaz. Privacy-utility tradeoff under statistical uncertainty. *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1627–1634, 2013.

MuonNeutrino. US Census Demographic Data, 2017. URL https://www.kaggle.com/muonneutrino/us-census-demographic-data?select=acs2017_census_tract_data.csv, Last accessed on 2020-4-24.

S. Raj Rajagopalan, Lalitha Sankar, Soheil Mohajer, and H. Vincent Poor. Smart meter privacy: A utility-privacy framework. *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Oct 2011. doi: 10.1109/smartgridcomm.2011.6102315. URL http://dx.doi.org/10.1109/SmartGridComm.2011.6102315.

Borzoo Rassouli and Deniz Gündüz. Optimal utility-privacy trade-off with total variation distance as a privacy measure. *IEEE Transactions on Information Forensics and Security*, 15:594–603, 2019.

Lalitha Sankar, S. Raj Rajagopalan, and H. Vincent Poor. Utility-privacy tradeoffs in databases: An information-theoretic approach. *IEEE Transactions on Information Forensics and Security*, 8(6): 838–852, 2013. ISSN 1556-6013. doi: 10.1109/TIFS.2013.2253320. Copyright: Copyright 2013 Elsevier B.V., All rights reserved.

Chandra Sharma, Bishwas Mandal, and George Amariucai. A practical approach to navigating the tradeoff between privacy and precise utility. In *ICC 2021 - IEEE International Conference on Communications*, pp. 1–6, 2021. doi: 10.1109/ICC42927.2021.9500410.

Jiaming Song, Pratyusha Kalluri, Aditya Grover, Shengjia Zhao, and Stefano Ermon. Learning controllable fair representations. *CoRR*, abs/1812.04218, 2018. URL http://arxiv.org/abs/1812.04218.

Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002. URL http://dblp.uni-trier.de/db/journals/ijufks/ijufks10.html#Sweene02.

Hao Wang and Flavio P Calmon. An estimation-theoretic view of privacy. In *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 886–893. IEEE, 2017.

Hao Wang, Mario Diaz, Flavio P Calmon, and Lalitha Sankar. The utility cost of robust privacy guarantees. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 706–710. IEEE, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

A APPENDIX

In this section we provide the implementation details for different adversary and utility models:

A.1 MNIST EXPERIMENTS

Optimizer: Stochastic Gradient Descent (SGD)
Learning rate: 0.01
Loss: Categorical Cross-entropy
Batch Size: 32
Epochs: 35

For MNIST Case 1 experiment where discriminators are trained either on distorted data or trained together with the joint training:
$\lambda_P = 50$ for (Model 4 and Model 5), $\lambda_P = 0.8$ for (Model 3).

For MNIST Case 2 experiment where discriminators are trained either on distorted data or trained together with the joint training:
$\lambda_P = 60$ for (Model 4 and Model 5), $\lambda_P = 0.7$ for (Model 3).

Table 6: This neural network architecture reflect the architecture of both adversary and utility models for **MNIST case 1 and case 2** for discriminator mentioned in Model 1, Model 3, Model 4, Model 5 and Model 6(in case of joint setting)

Name	**Configuration**	**Repetition**
Input Layer	Input shape = (28 * 28)	1
Reshape	Reshape(28,28,1)	1
Convolution	conv2D (filter = 32 kernel size = 4), activation = relu	1
MaxPooling	MaxPooling2D(pool_size=(2, 2))	1
Convolution	conv2D (filter = 16 kernel size = 4), activation = relu	1
MaxPooling	MaxPooling2D(pool_size=(2, 2))	1
Flatten	Flatten()	1
Output	Fully Connected FC (output shape), activation = softmax	1

Table 7: This neural network architecture reflect the architecture of both adversary and utility models for **MNIST case 1 and case 2** for discriminator mentioned in Model 2

Name	**Configuration**	**Repetition**
Input Layer	Input shape = (28 * 28)	1
Reshape	Reshape(28,28,1)	1
Convolution	conv2D (filter = 64, kernel size = 4), activation = relu	1
MaxPooling	MaxPooling2D(pool_size=(2, 2))	1
Convolution	conv2D (filter = 16 kernel size = 4), activation = relu	1
MaxPooling	MaxPooling2D(pool_size=(2, 2))	1
Flatten	Flatten()	1
Output	Fully Connected FC (output shape), activation = softmax	1

A.2 FASHION MNIST EXPERIMENTS

Optimizer: Stochastic Gradient Descent (SGD)
Learning rate: 0.01
Loss: Categorical Cross-entropy
Batch Size: 32

Epochs: 30

For FashionMNIST experiment where discriminators are trained either on distorted data or trained together with the joint training:
$\lambda_P = 60$ for (Model 4 and Model 5), $\lambda_P = 0.8$ for (Model 3).

Table 8: This neural network architecture reflect the architecture of both adversary and utility models for all discriminators of **FashionMNIST** experiments

Name	Configuration	Repetition
Input Layer	Input shape = (28 * 28)	1
Reshape	Reshape(28,28,1)	1
Convolution	conv2D (filter = 32, kernel size = 3), activation = relu, kernel initializer = he_uniform, padding = same, BatchNormalization()	2
MaxPool	Maxpooling2D(pool size = (2,2)), Dropout(0.3)	1
Convolution	conv2D (filter = 64, kernel size = 3), activation = relu, kernel initializer = he_uniform, padding = same, BatchNormalization()	2
MaxPool	Maxpooling2D(pool size = (2,2)), Dropout(0.4)	1
Convolution	conv2D (filter = 128, kernel size = 3), activation = relu, kernel initializer = he_uniform, padding = same, BatchNormalization()	2
MaxPool	Maxpooling2D(pool size = (2,2)), Dropout(0.5)	1
Flatten	Flatten()	1
Dense	FC(128), activation = relu, kernel initializer = he_uniform	1
BatchNorm	BatchNormalization()	1
Dropout	Dropout(0.6)	1
Output	FC(Output shape), activation = softmax	1

A.3 UCI ADULT EXPERIMENTS

Optimizer: Adam
Loss: Categorical Cross-entropy
Batch Size: 512
Epochs: 20

For UCI Adult experiment where discriminators are trained either on distorted data or trained together with the joint training:
$\lambda_P = 50$ for (Model 4 and Model 5), $\lambda_P = 0.8$ for (Model 3).

Table 9: This neural network architecture reflect the architecture of both adversary and utility models for all discriminators of **UCI adult** experiments

Name	Configuration	Repetition
Input Layer	Input shape = (input shape = 102)	1
Dense	FC(256), activation = relu	1
Dropout	Dropout(0.2)	1
Dense	FC(256), activation = relu	1
Dropout	Dropout(0.3)	1
Dense	FC(128), activation = relu	1
Dropout	Dropout(0.4)	1
Output	FC(output shape), activation = softmax	1

A.4 US DEMOGRAPHY CENSUS DATA EXPERIMENTS

Optimizer: Adam
Loss: Categorical Cross-entropy
Batch Size: 512
Epochs: 50

For US Demography Census Data experiment where discriminators are trained either on distorted data or trained together with the joint training:
$\lambda_P = 60$ for (Model 4 and Model 5), $\lambda_P = 0.6$ for (Model 3).

Table 10: This neural network architecture reflect the architecture of both adversary and utility models for all discriminators of **US Census Demographic Data** experiments

Name	**Configuration**	**Repetition**
Input Layer	Input shape = (input shape = 14)	1
Dense	FC(64), activation = relu	3
Output	FC(output shape) , activation = softmax	1

For more details on the architecture of the encoder and decoder, different system parameters based on which the experiments were conducted please refer to this url: `https://anonymous.4open.science/r/abc-6BAC/README.md`. It also provides the weights of all the pre-trained adversary and utility models for testing of the privacy mechanism.