sensors

MDPI

Article

Respecting Partial Privacy of Unstructured Data via Spectrum-Based Encoder

Qingcai Luo [1,]* and Hui Li [2]

[1] School of Cyber Engineering, Xidian University, Xi'an 710126, China
[2] School of Computer Science and Technology, Xidian University, Xi'an 710071, China; hli@xidian.edu.cn
* Correspondence: luoqc@inspur.com

Abstract: Since the popularity of Machine Learning as a Service (MLaaS) has been increasing significantly, users are facing the risk of exposing sensitive information that is not task-related. The reason is that the data uploaded by users may include some information that is not useful for inference but can lead to privacy leakage. One straightforward approach to mitigate this issue is to filter out task-independent information to protect user privacy. However, this method is feasible for structured data with naturally independent entries, but it is challenging for unstructured data. Therefore, we propose a novel framework, which employs a spectrum-based encoder to transform unstructured data into the latent space and a task-specific model to identify the essential information for the target task. Our system has been comprehensively evaluated on three benchmark visual datasets and compared to previous works. The results demonstrate that our framework offers superior protection for task-independent information and maintains the usefulness of task-related information.

Keywords: spectrum-based encoder; latent code; machine learning; privacy preserving

check for updates

Citation: Luo, Q.; Li, H. Respecting Partial Privacy of Unstructured Data via Spectrum-Based Encoder. *Sensors* **2024**, *24*, 1015. https://doi.org/10.3390/s24031015

Academic Editors: Kechen Zheng, Xiaoying Liu, Zhao Li, Yang Xu and Jia Liu

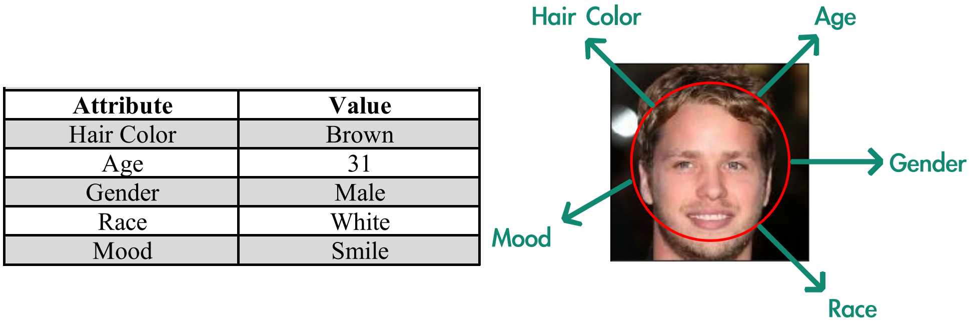
Received: 7 December 2023
Revised: 1 February 2024
Accepted: 2 February 2024
Published: 4 February 2024

1. Introduction

Machine learning has demonstrated impressive performance in several areas, such as natural language processing [1] and computer vision [2]. However, training an effective machine learning model requires proper model design, massive computing resources, and large datasets that may be beyond the reach of many individuals. In addition, deploying and running the model requires significant storage and computing resources that are also unfriendly to edge devices such as smartphones or sensors [3]. One promising approach is Machine Learning as a Service (MLaaS) [4], which supports the outsourcing of prediction. Well-trained models can be deployed by vendors in the cloud. This is attractive because it offloads the user's local computing and storage requirements and eliminates the cost of training new models. However, the outsourced data consist of not only task-related information, but also task-independent information [5], which does not significantly affect the inference results, but exposes users to unwanted risks of misuse or theft. Recently, China's Personal Information Protection Law has prompted information processors to prevent unauthorized access to personal information. Therefore, it is of paramount importance to protect unauthorized information while ensuring the usefulness of the data.

Previous works addressing privacy concerns have been devoted to balancing the trade-offs between privacy and utility. An obvious and widely adopted solution is to extract task-oriented features and upload them to servers instead of raw data, such as Google Now [6] and Google Cloud [7]. Although the mere transmission of features avoids direct disclosure of raw data, recent developments in model inversion attacks show that adversaries can use intermediate features to reconstruct the input and infer privacy attributes [8–10]. Ossia et al. [11] apply dimensionality reduction and noise injection to defend against adversaries before uploading features to the servers, but the cost is a non-negligible loss in utility. Inspired by Generative Adversarial Networks (GANs) [12], PAN [13], DeepObfuscator [3], and TIPRDC [14] propose to obtain an encoder through adversarial training to extract partial privacy-preserving features that keep a subset of the attributes available while specifying the attributes anonymously. However, these schemes artificially simulate proxy adversaries during the training phase, leading to dangers from potential attack models. This suspicion is also supported by the results of the potential adversary detection experiments in Section 4.2.

Therefore, we propose a partial privacy-preserving framework to preserve data utility while protecting task-independent attributes. An intuitive phenomenon is that not all data information is useful for inference. Some of the recent literature shows that the task model pays more attention to a part of regions [15–18], which becomes evidence that the data can be regarded as composed of task-related and task-independent information. Inspired by these works, our framework focuses on selecting the information relevant to the target task. This is feasible for structured data, but difficult for unstructured data. Taking Figure 1 as an example, users can flexibly select the attributes necessary for the task in Table (a) due to the naturally independent entries, while it is impractical for image (b) because different attributes are entangled and expressed in the same region. An intuitive approach is to express unstructured data in a structured form. However, naturally occurring data are often accompanied by redundant information, which hinders structured expression. Therefore, we introduce Fourier transform as a pre-processing method to reduce data redundancy, and propose the spectrum-based encoder to disentangle the unstructured data into a latent space [19]. We then propose a universally interpretable model, called an indicator, which marks the information necessary for the target task in the latent representation.



Attribute	Value
Hair Color	Brown
Age	31
Gender	Male
Race	White
Mood	Smile

(**a**) (**b**)

Figure 1. (**a**) Structured data and (**b**) unstructured data.

As shown in Figure 2, our framework consists of three parts: a spectrum encoder E, an indicator I, and a decoder D. The encoder E is intended to be used on the user side to extract the disentangled representation from unstructured data. Indicator I is also used on the user side, recommending task-related information by marking representation dimensions. The marked dimensions indicate the information required by the target task model, and the corresponding anonymized transform is designed. Specifically, the values of the marked dimensions are retained, while the values of the ignored dimensions are discarded and reassigned as default values. The decoder D runs on the server to reconstruct the data based on the transformed representation uploaded by the users. The classifier (green) is considered the target task model, and the reconstruction data will strive to maintain its usefulness. At the same time, the reconstruction data are expected to prevent adversaries (red) from inferring unauthorized attributes.

Discarding task-independent information according to Indicator's recommendations has four advantages. First, interpretable indicators provide interpretability for anonymized transformation. Second, target-task-driven attribute retention avoids unconscious utility loss and sensitive information leakage. Third, disentangled representation-based information selection provides an explicit and controllable balance for privacy–utility trade-offs. Finally, this allows our framework to withstand potential attack models. Furthermore, Indicator and encoder–decoder pairs of our framework are trained separately in two phases.

Compared to existing end-to-end adversarial training methods, our framework can adapt to the changes in the target tasks and adjust the retained attribute information more flexibly.

In summary, our key contributions are as follows:

- We introduce a novel interpretable model called Indicator, which can effectively indicate the critical information required for a specific target task within unstructured data.
- We present a partial privacy-preserving framework that utilizes the designed Indicator to restrict the access of undesired task-independent attacks while preserving the utility of target tasks.
- We fully implement our framework and demonstrate its wide applicability by performing experiments on several standard datasets. The evaluation results show that our framework can achieve sweet trade-offs between privacy and utility, and is resistant to potential attackers.

The rest of this paper is organized as follows. Section 2 introduces the preliminaries and reviews the related work. Section 3 describes the framework overview and the details of core modules. Section 4 reports the evaluation results. Section 5 concludes and discusses this paper.

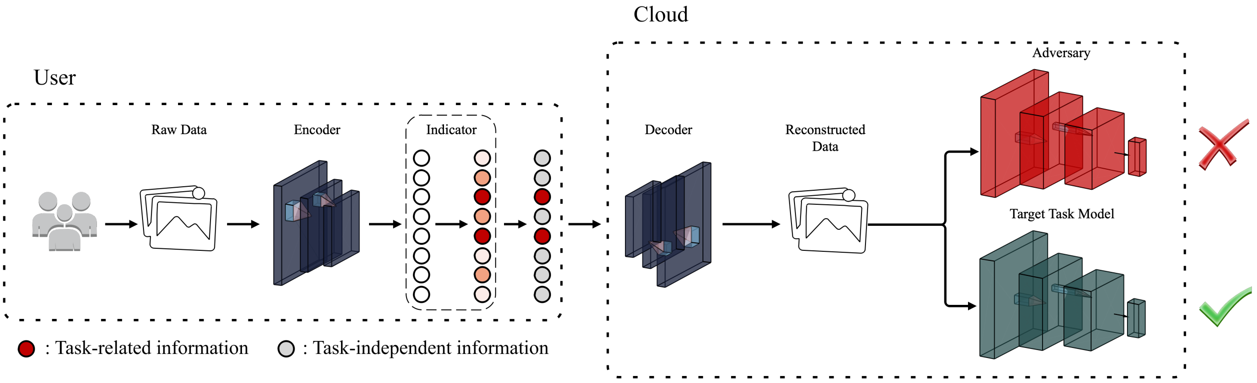


Figure 2. The VAE-based encoder maps the raw data to the latent space, and the proposed indicator points out the relevance of the latent code to the target task and removes irrelevant elements. The subsequent decoder reconstructs the data from the filtered code, with the target attributes being preserved while the remaining attributes are obfuscated.

2. Preliminaries and Related Work

In this section, we first introduce the work involved in this article. Then, we briefly review the most relevant work on privacy.

2.1. Disentangled Representation Learning

In general, disentangled representation learning aims to isolate different attributes into non-overlapping sub-dimensions in the latent space. As shown in Figure 3, different colours represent different attributes in the raw data, and the ball represents the factor containing attribute information. In the raw data, these factors are messy and entangled, and it is difficult to filter all the factors corresponding to a certain attribute in a common way. At the same time, the latent code obtained by the disentangled representation learning can express attributes regularly and independently. In other words, different attributes in the raw data can be determined by the different representation sub-dimensions in the latent space.

Existing works about disentangled representation can be roughly divided into three categories: (1) based on Variational Autoencoders (VAE) [19–21], (2) based on GAN [22] and (3) based on the flow model [23]. Among them, the VAE-based model is attractive due to its lower cost and stability in the training phase.

VAE is an unsupervised generative network based on variational bayes inference, consisting of an encoder and a decoder. Given a sample x, VAE determines a distribution

z in the latent space as the encoding result. The optimization objective of VAE consists of two parts. The first part is to maximize the Evidence Lower Bound (ELBO) so that the variational distribution is close to the isotropic Gaussian prior $p(z)$, and the second part is to minimize the pixel-level metrics of the generated data and the original data:

$$L_{VAE} = -\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[log(p(\boldsymbol{x}|\boldsymbol{z})] + D_{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) \tag{1}$$

β-VAE [19] modified the objective function as:

$$L_{\beta VAE} = -\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[log(p(\boldsymbol{x}|\boldsymbol{z})] + \beta D_{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) \tag{2}$$

Compared to the original VAE, the hyperparameter $\beta > 1$ encourages the variational distribution to be closer to the Gaussian prior, thereby producing a disentangled latent code. Kim et al. [20] and Chen et al. [21] believed that the total correlation term obtained by decomposing the KL divergence plays a crucial role and proposed Factor-VAE and β-TCVAE, respectively.

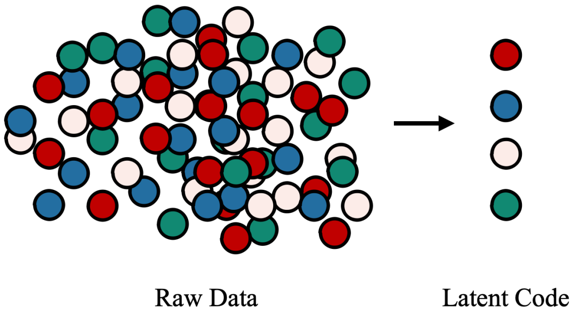


Figure 3. Different colours represent different attributes in the unstructured data, and the balls represent the factors that affect the attributes.

2.2. Data Privacy Protection

Several methods have been proposed to protect privacy. k-anonymity [24], l-diversity [25], and t-closeness [26] have been proposed as desensitization criteria. However, these methods are only designed for structured data and are difficult to scale to unstructured data. Differential privacy [27–29] and random noise injection [30,31] are common methods that are widely used to protect sensitive information in structured and unstructured data. Although security guarantees are provided, these methods often significantly reduce the usefulness of the data. Homomorphic encryption (HE) [32,33] and Secure Multi-Party Computation (MPC) [34–36] support the manipulation of encrypted data, but the computation of non-linear functions is always accompanied by unrealistic computational and communication complexity, leading to much lower efficiency than plaintext inference. iPrivacy [37] focuses on visual tasks by constructing a multi-task learning model to detect and blur objects that may leak sensitive information in the image. The types of these objects are preset. RAE [38] follows the same idea but is time-series-oriented. This scheme proposes to replace the features of each section corresponding to sensitive inferences with the values corresponding to non-sensitive inferences. Using GAN, RAE provides the security guarantee that it is almost impossible to detect the nature of sensitive inferences.

2.3. Representation Privacy–Utility Trade-Offs

Aloufi et al. [39] focused on the disentanglement of voice for the Voice User Interfaces (VUIs). VQ-VAE [40] was introduced to construct independent representations of emotion, identity, and semantics, while WaveRNN [41] was employed to reconstruct voice information. Gong et al. [42] are concerned about attributes preserving face de-identification and propose R^2VAEs to obfuscate identity-related information so as to achieve a balance between facial privacy protection and data utilities. Wu et al. [43] jointly proposed a securely recoverable visual information transformation and steganography PECAM based on deep learning. They used this technology to design a more general VSA privacy enhancement architecture and system implementation. PECAM can effectively transform the original data to other domains to hide sensitive information. At the same time, authorized users can inversely transform and restore the original data to complete detailed investigations. This secure reversible transformation relies on a security-enhanced generative adversarial network. Also, it introduces a key mechanism to ensure that attackers cannot restore the data protected by PECAM. The adversary and the defender are given the conflicting utility–privacy optimization goal, and the game between them is simulated. AttriGuard [44] proposed a two-phase practical framework to resist private attribute inference attacks. In phase I, existing evasion attacks in adversarial learning are adopted to find the minimum noise for each attribute value. In phase II, the attribute values are sampled with a certain probability distribution, and the minimum noise found in phase I is added to the dataset. Therefore, finding the probability distribution is formulated as a constrained convex optimization problem. Liu et al. proposed PAN to protect the privacy of a specific attribute while maintaining the data utility for a certain task. The representation obtained by PAN will remain anonymous, and the adversaries cannot launch reconstruction attacks or privacy attributes inference attacks. Wu et al. [45] designed an adversarial training framework to obtain the degradation transform of video inputs to resist privacy attribute attacks. Considering the diversity of attack models, and that it is impossible to enumerate all adversary models to enhance the features privacy, Budget Model Restarting and Budget Model Ensemble are enabled to enrich potential adversaries. TIPRDC is a task-independent privacy-respecting data crowdsourcing framework but following the same idea. Unlike the above works, the data utility maintained by TIPRDC does not limit to specific tasks but is effective for arbitrary tasks by maximizing mutual information. In a sense, our work is diametrically opposed to the idea of TIPRDC: TIPRDC struggles to **retain** all information in the data, except for privacy attributes, while our framework is expected to **remove** all information, except for the target task required.

3. Design of Framework

In this section, we introduce the VAE-based disentanglement method and propose the model called Indicator for filtering the factors related to the target task.

3.1. Overview

Because models do not need all the information in the uploaded data to make credible inferences, users tend to share only task-relevant details in a controlled manner. This is practical for structured data with naturally independent attribute records but is difficult for unstructured data. Thus, our framework is proposed to sift task-related information from the unstructured data while confusing task-independent information. Figure 2 shows that our framework addresses this problem in three stages. In the first stage, the encoder in the VAE family model is used to obtain the disentangled representation, from which different attribute information can be independently selected. Although the disentangled representation is similar in form to the structured data, users are still confused about which dimensions are necessary due to the lack of semantic interpretation. Therefore, in the second step, we propose a model called Indicator that provides suggestions for explicit user control over the information. In the representation, the dimensions marked by Indicator are frozen, while the values of the remaining dimensions are discarded and refilled. In the third stage, the transformed representation is fed to the decoder that corresponds to the encoder in the first stage for data reconstruction. The task-related information in the reconstructed data is preserved, while the task-independent information is unreliable.

3.2. Unstructured Data Disentanglement

The information of different attributes in unstructured data is often intertwined and almost impractical to select independently. By disentangling different attributes, it is possible to preserve the task-related factors of unstructured data while obfuscating the task-independent factors. As shown in Figure 4, we employ the VAE family models (β-VAE, Factor-VAE, and β-TCVAE) in the training phase to obtain an encoder–decoder pair. The encoder is used to extract the disentangled representation, and the decoder is used to reconstruct the data. However, the common problem is that the data generated by VAEs is always ambiguous. One view is that the pixel-wise reconstruction error metric causes the generated data to be too smooth [46]. In contrast, the main idea of GAN is to provide a game between the generator and the discriminator. During this game, the discriminator judges the original data as true and the generated data as false at each iteration. Meanwhile, the generator tries to fool the discriminator into judging the generated data as true in the same iteration. Therefore, the decoder can be considered as the generator and a discriminator is introduced to improve the quality of the generated data. To avoid affecting the disentanglement of the representation, in each iteration, the training of the GAN is carried out after the training of the VAEs, which means that the encoder and the discriminator are not end-to-end. Formally, the loss function can be defined as:

$$L_{VAEs}(\theta_{End}, \theta_{Dec}) = L_{VAEs} \tag{3}$$

$$L_{GAN}(\theta_{Dec}) = E_x[log(Dis(x)) + log(1 - Dis(x'))] \tag{4}$$

$$L_{GAN}(\theta_{Dis}) = -E_x[log(log(Dis(x')))] \tag{5}$$

where L_{VAEs} represents different loss functions in the VAE family and θ_* indicate the parameters to be updated.

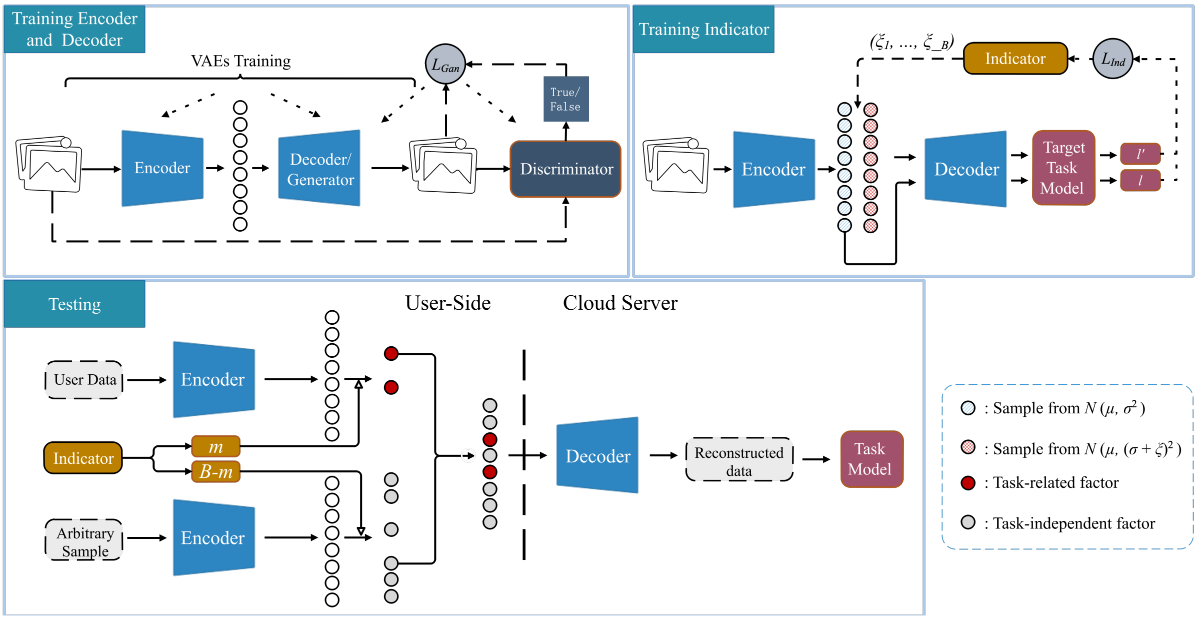


Figure 4. The workflow of our framework. The top line is the training stage, including the training of the encoder–decoder pair and Indicator. The bottom line is the test stage. An indicator is introduced to recommend the indexes of the representation dimensions that need to be retained. At the same time, an arbitrary sample is used as a carrier to supplement the remaining dimensions.

In the testing phase, the encoder is deployed on the user side while the corresponding decoder runs on the cloud server.

3.3. Representation Oriented Indicator

After the encoder and decoder training, the encoder can standard express different attributes in the latent space. Such a disentangled representation allows us to obfuscate the task-independent factors without changing task-related factors. However, data contain many factors, and it is impractical to enumerate all task-independent attributes. In addition, whether a factor is related to the task depends on the specific task model. Different classifiers may focus on different associated attributes for the same classification task. For example, one classifier will concentrate on hair when judging the gender of a face image, while another classifier may focus on beards. The tendency of the classifier depends on the training set and model structure, which is uncontrollable for the user. If the factors to be obfuscated are rashly determined based on human perception, it will inevitably affect the effectiveness of the primary task. For this consideration, the task-adaptive Indicator is proposed to mark the attributes that the specific task model focuses on.

Different dimensions in the disentangled representation are considered disjoint, and a set of sub-dimensions can only express a particular data attribute. Meanwhile, the task model does not view all the information to make credible inferences but pays more attention to specific attributes. This is equivalent to that only one set of sub-dimensions in the disentangled representation contributes to the task model inference while discarding the values of the remaining dimensions has almost no effect. Following the idea, the proposed Indicator is designed to search this set of sub-dimensions. Indicator is expected to have both fidelity and interpretability. Fidelity means that Indicator can accurately mark the representation dimensions necessary for the task model. The interpretability signifies that the decision-making process is consistent with the human perspective.

Figure 5 reviews the paradigm of the VAE family. Each original datapoint $\boldsymbol{x}^{(i)}$ is encoded into a multivariate gaussian distribution $\mathcal{N}(\boldsymbol{\mu}^{(i)}, (\boldsymbol{\sigma}^{(i)})^2)$, and the decoding results $\boldsymbol{x}'^{(i)}$ of all samples in $\mathcal{N}(\boldsymbol{\mu}^{(i)}, (\boldsymbol{\sigma}^{(i)})^2)$ are similar to the original data $\boldsymbol{x}^{(i)}$. Given an original datapoint $\boldsymbol{x}^{(i)}$, its disentangled representation $\boldsymbol{z}^{(i)}$ can be represented by $\boldsymbol{z}^{(i)} \sim \mathcal{N}(\boldsymbol{\mu}^{(i)}, (\boldsymbol{\sigma}^{(i)})^2)$, and $\boldsymbol{z}^{(i)} \in \mathbb{R}^B$. For the inference of a certain task model, there are m necessary dimensions in $\boldsymbol{z}^{(i)}$, whose value fluctuation will significantly affect the result of the inference, while the change in the remaining $B - m$ dimensions can hardly have impacts. This demonstrates that under the premise of not affecting the inference confidence, the larger variance is not tolerated by the m dimensions, but is acceptable for the $B - m$ dimensions. Therefore, the ultimate goal of Indicator can be expressed as finding a variance bias $\boldsymbol{\xi}$ as large as possible and encoding the data $\boldsymbol{x}^{(i)}$ into the new distribution $\mathcal{N}(\boldsymbol{\mu}^{(i)}, (\boldsymbol{\sigma}^{(i)} + \boldsymbol{\xi})^2)$, as shown in Figure 5. Among them, the decoding result $\widetilde{\boldsymbol{x}}'^{(i)}$ of the sample on $\mathcal{N}(\boldsymbol{\mu}^{(i)}, (\boldsymbol{\sigma}^{(i)} + \boldsymbol{\xi})^2)$ and the decoding result $\boldsymbol{x}'^{(i)}$ of the sample on $\mathcal{N}(\boldsymbol{\mu}^{(i)}, (\boldsymbol{\sigma}^{(i)})^2)$ show the same confidence in the task model. In general, the optimization goal of $\boldsymbol{\xi}$ can be formulated as:

$$
\begin{aligned}
&minL(\boldsymbol{\xi}) = minE_{p(\boldsymbol{x})}[E_{q(\boldsymbol{z}|\boldsymbol{x}),\, \widetilde{q}(\widetilde{\boldsymbol{z}}|\boldsymbol{x})}[L2(TM(Dec(\boldsymbol{z})), TM(Dec(\widetilde{\boldsymbol{z}})))]] - \lambda \sum_{i=1}^{B} \boldsymbol{\xi}_i, \\
&q(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2),\ \widetilde{q}(\widetilde{\boldsymbol{z}}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}, (\boldsymbol{\sigma} + \boldsymbol{\xi})^2)
\end{aligned}
\tag{6}
$$

where TM represents the target task model.

After training, the representation dimensions corresponding to the smaller $\boldsymbol{\xi}_i$ cannot support the larger sampling ranges while maintaining effectiveness for the task model, which means that the task model will pay more attention to these dimensions. Conversely, the dimensions corresponding to a larger $\boldsymbol{\xi}_i$ contribute less to the task model inference. To determine $\boldsymbol{\xi}$, an intuitive method is to set $\boldsymbol{\xi}$ as trainable parameters. However, there are two problems with this method. First, since Indicator aims to explore the tolerance of different dimensions to the larger variance, $\boldsymbol{\xi}_i$ is expected to be non-negative. Also, too large a variance $\boldsymbol{\sigma} + \boldsymbol{\xi}$, which leads to meaningless sampling, can cause training to collapse. Therefore, $\boldsymbol{\xi}_i$ should be restricted to the interval $[0, \delta)$. Second, $\boldsymbol{\xi}$ is the variable in the distribution $\mathcal{N}(\boldsymbol{\mu}, (\boldsymbol{\sigma} + \boldsymbol{\xi})^2)$ and the sampling process is not differentiable.

For the first problem, we design a function $\boldsymbol{\xi}_i = f(\boldsymbol{\beta}_i)$, $\boldsymbol{\beta}_i \in R$, $\boldsymbol{\xi}_i \in [0, \delta)$ to eliminate the constraint on $\boldsymbol{\xi}$, where $\boldsymbol{\beta}$ is Indicator parameters. Considering the $\lambda \sum_{i=1}^{B} \boldsymbol{\xi}_i$ term in Equation (6), $f()$ should also satisfy monotonicity. If $\boldsymbol{\xi}_i$ can take the minimum value of 0 when $\boldsymbol{\beta}_i = 0$, the training of Indicator will benefit from the sparse parameters. Formally, $f()$ can be defined as:

$$\xi_i = f(\boldsymbol{\beta}_i) = \delta \cdot (\frac{1 - e^{-\boldsymbol{\beta}_i^2}}{e^{-\boldsymbol{\beta}_i^2} + 1}) \tag{7}$$

Among them, $f()$ is monotonically increasing in $[0, +\infty)$, monotonically decreasing in $(-\infty, 0]$, and the minimum value is 0 at $\boldsymbol{\beta}_i = 0$. In order to solve the second problem, we borrow the reparameterization trick to convert $\widetilde{z}^{(i)} \sim \mathcal{N}(\boldsymbol{\mu}^{(i)}, (\boldsymbol{\sigma} + \boldsymbol{\xi})^{2(i)})$ to $\widetilde{z}^{(i)} = \boldsymbol{\mu}^{(i)} + (\boldsymbol{\sigma} + \boldsymbol{\xi})^{(i)}\epsilon$, $\epsilon \sim \mathcal{N}(0, I)$ to make $\boldsymbol{\xi}$ differentiable. In summary, the formal loss function $L_{Indicator}$ is expressed as follows:

$$\begin{aligned} L_{indicator}(\boldsymbol{\beta}) =& E_{p(\boldsymbol{x})}[E_{q(\boldsymbol{z}|\boldsymbol{x}),\, \widetilde{q}(\widetilde{z}|\boldsymbol{x})}[L2(TM(Dec(\boldsymbol{z})), TM(Dec(\widetilde{\boldsymbol{z}})))]] - \lambda \sum_{i=1}^{B} f(\boldsymbol{\beta}_i) + \|\boldsymbol{\beta}\|_2, \\ q(\boldsymbol{z}|\boldsymbol{x}) =& \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2),\ \widetilde{q}(\widetilde{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}, (\boldsymbol{\sigma} + f(\boldsymbol{\beta}))^2) \end{aligned} \tag{8}$$

Finally, the representation dimensions corresponding to the parameters satisfying $|\boldsymbol{\beta}_i| < \psi$ are considered more relevant by the task model and their indices are recorded, where ψ is the threshold. The entire training process of the proposed Indicator is shown in Algorithm 1.

Algorithm 1 Indicator Training

```
Input: Encoder $Enc()$, Conversion function $f()$, Decoder $Dec()$, task model $Tm()$, disentangled representation size $B$, threshold $\psi$.
Output: Dimension index to be retained $Index$.
$\boldsymbol{\beta} \leftarrow$ random initialize Indicator parameters
for epochs do
    Random mini-batch $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)}\}$,
    $\boldsymbol{\mu}, \boldsymbol{\sigma} = Enc(\boldsymbol{X}), \boldsymbol{\mu}, \boldsymbol{\sigma} \in R^{n \times B}$
    $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \times \epsilon,\ \epsilon \sim \mathcal{N}(0, I)$
    $\widetilde{\boldsymbol{z}} = \boldsymbol{\mu} + (\boldsymbol{\sigma} + f(\boldsymbol{\beta})) \times \epsilon,\ \epsilon \sim \mathcal{N}(0, I)$
    $\boldsymbol{l} = Tm(Dec(\boldsymbol{z})),\ \widetilde{\boldsymbol{l}} = Tm(Dec(\widetilde{\boldsymbol{z}}))$
    $\boldsymbol{\beta} \overset{update}{\longleftarrow} min(L2(\boldsymbol{l}, \widetilde{\boldsymbol{l}}) - \sum_{i=1}^{B} f(\boldsymbol{\beta}_i)) + \|\boldsymbol{\beta}\|_2$
end for
$Index = \{\}$
for $\boldsymbol{\beta}_i \in \boldsymbol{\beta}$ do
    if $|\boldsymbol{\beta}_i| < \psi$ then
        $Index.append(i)$
    end if
end for
```

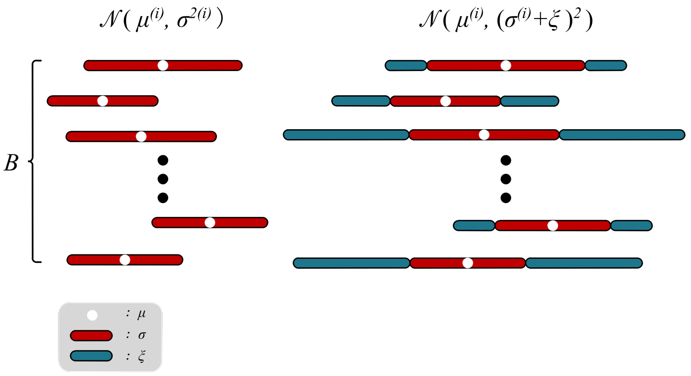

$\mathcal{N}(\mu^{(i)}, \sigma^{2(i)})$
$\mathcal{N}(\mu^{(i)}, (\sigma^{(i)}+\xi)^2)$
B
: μ
: σ
: ξ

Figure 5. Illustration of how the indicator works. Indicator searches for the maximum allowable oscillation range that remains utility for the task model in the B representation dimensions.

3.4. Data Reconstruction

The disentangled representation encoded by the encoder allows obfuscation of the task-independent features without changing the task-related features. It is necessary to preserve the m dimensions marked by Indicator because the task model pays more attention to them. At the same time, the remaining $B - m$ dimensions, which contribute less to task inference but contain excessive task-independent information, should be discarded. Theoretically, it is possible to replace the original values of the $B - m$ dimensions with arbitrary values. In practice, however, completely random values will make it easier for the transformed representation to decode ambiguous data, resulting in task-relevant information not being correctly expressed. Even though the $B - m$ dimensions have nothing to do with the task model, it is still necessary to be careful when choosing their replacement values. As shown in the test phase of Figure 4, our method uses an arbitrary sample as a carrier. It concatenates the $B - m$ dimensions in the carrier representation with the m dimensions in the original data representation. By reconstructing the data from such a representation, only the factors that the task model focuses on are credible, while others are confusing.

4. Experimental Study

In this section, we first qualitatively evaluate the proposed Indicator and report the experimental results. Then, we quantify the privacy–utility trade-offs of our framework and present a comparison with other popular methods. The following experiments involve three datasets: dSprites [47], MNIST [48], and CelebA [49]. dSprites contains $737,280$ 2D synthesis samples with 6 attributes. We randomly divide $589,824$ samples for training VAE family models and Indicators and $147,456$ samples for testing. MNIST contains grayscale images of 10 classes of handwritten digits, including $60,000$ training samples and $10,000$ testing samples. CelebA includes $202,599$ face images labeled with 40 binary attributes, of which $162,770$ images are divided for training and $39,829$ images for testing. The experiments are conducted on Nvidia GTX 3080Ti GPU in Pytorch.

4.1. Indicator Evaluation

To qualitatively demonstrate the effectiveness of the proposed Indicator, we conduct experiments on dSprites and MNIST from three perspectives. (a) Versatility: whether the proposed Indicator can be effectively combined with the various VAEs models. (b) Reliability: whether the task model considers the dimensions marked by Indicator. (c) Stability: whether the Indicator can make the same decision under different initial conditions and training subsets.

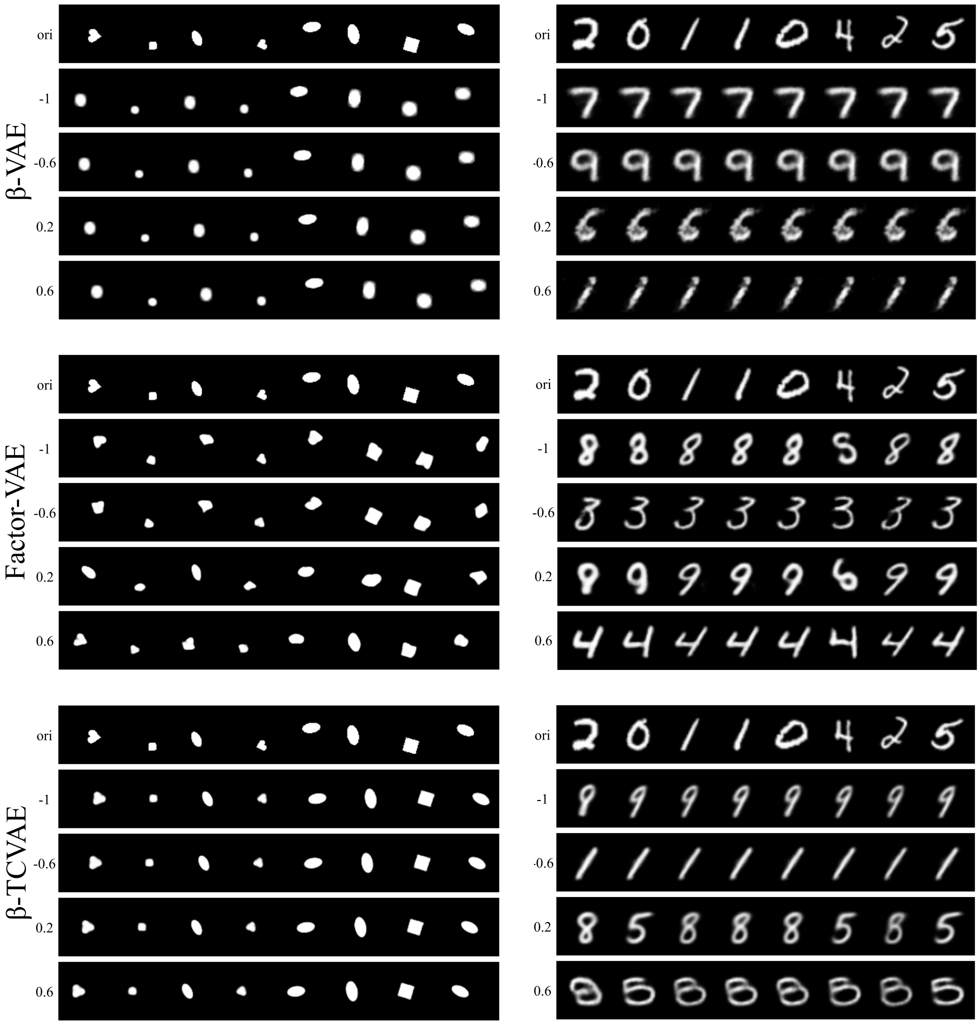
4.1.1. Versatility

To illustrate the versatility, the following experiments are performed on β-VAE, Factor-VAE, and β-TCVAE, respectively.

4.1.2. Reliability

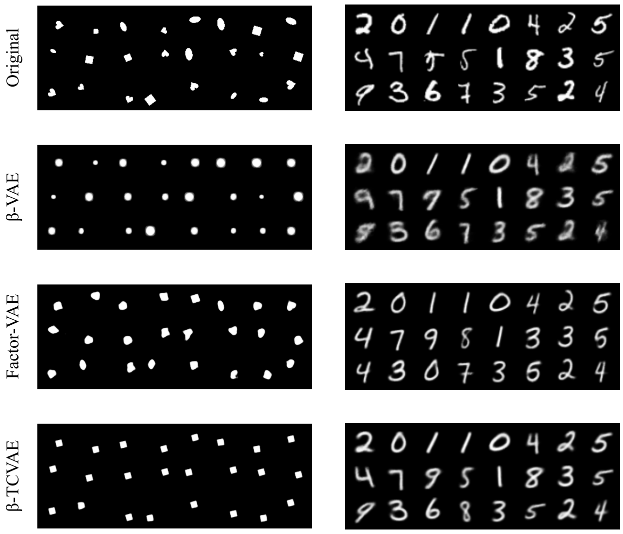
The verification of the reliability is studied by two experiments. For dSprites and MNIST, the dimension B of the disentangled representation is set to 10, and the threshold δ is set to 0.5. For dSprites, a classifier focusing on the X-position is used as the target task model. For β-VAE, Factor-VAE, and β-TCVAE, Indicator finds 3, 4, and 2 dimensions on which the target task model focuses. For MNIST, a classifier that distinguishes digits is the target task model. In the above three VAE models, Indicator finds 3, 4, and 3 task-related dimensions in the representation.

The first experiment is to interpolate the dimensions marked by Indicator while freezing the remaining dimensions. Figure 6 visualizes the reconstructed image traversing the marked dimensions. The first line is the original data, and the second to fifth lines are the reconstruction of the interpolated representation. On the one hand, the "X-position" that the task model focuses on changes with the change in the marked representation dimension. On the other hand, the experimental results show the difference in disentangling performance of different VAE schemes. In the second experiment, we fix the dimensions marked by Indicator and replace the values of the remaining dimensions with 0. Figure 7 shows the reconstruction of the processed representation. The attribute focused by the target task model is preserved, while the others become irrelevant to the original data. The above two experiments show that the task-related dimensions determined by the proposed Indicator are consistent with the human view, which confirms the reliability of Indicator to a certain extent. The quantitative measure of reliability can be decomposed into target task accuracy and the availability of task-independent attributes. Target task accuracy reflects whether the task-related dimensions are fully selected. The availability of task-independent attributes is directly proportional to the redundancy of the selected dimensions. Therefore, reliability is equivalent to the privacy–utility trade-offs of our framework, which will be discussed on the CelebA dataset in Section 4.2.



(**a**) dSprites (**b**) MNIST

Figure 6. Reconstructed image visualization of traversing the representation dimensions marked by the indicator.



(**a**) dSprites (**b**) MNIST

Figure 7. The representation dimensions marked by Indicator are fixed, while the values of the remaining dimensions are replaced with 0. The above illustrations are reconstructed images based on these processed representations.

4.1.3. Stability

To illustrate the stability of the proposed Indicator, we perform experiments on dSprites and MNIST with the same settings as in Section 4.1.2. The train sets of dSprites and MNIST are divided into 3 subsets, and then Indicator searches for task-related dimensions on each subset with random initial parameters. Figure 8 provides a visualization of Indicator parameters changing with epochs. In Figure 8, the dimensions that fall into the yellow area are considered more concerned by the task model. In rows 1 and 2, the indicators mark the disentangled representations generated from β-VAE. The Indicators in lines 3 and 4 mark the disentangled representations generated using Factor-VAE. Lines 5 and 6 are Indicator marking the disentangled representations generated using β-TCVAE. Taking the three subfigures in the first row as an example, the parameters corresponding to dimensions 4, 8, and 10 in the Indicator eventually converge to the yellow region, while the rest diverge. This represents that Indicator considers dimensions 4, 8 and 10 as being attended to by the target task. Under different dataset slices and random initial parameters, the tendency of Indicators to represent dimensions shows the same trend. This demonstrates the stability of the Indicator, where the marking process is not affected by the initial parameters and the division of the dataset. Moreover, the experimental results also support the conclusion that the same dimensions of the latent code of different samples correspond to the same information.

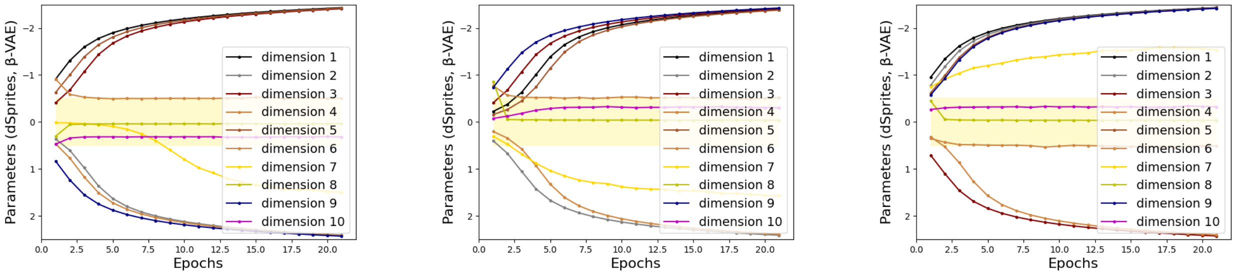


Figure 8. *Cont.*

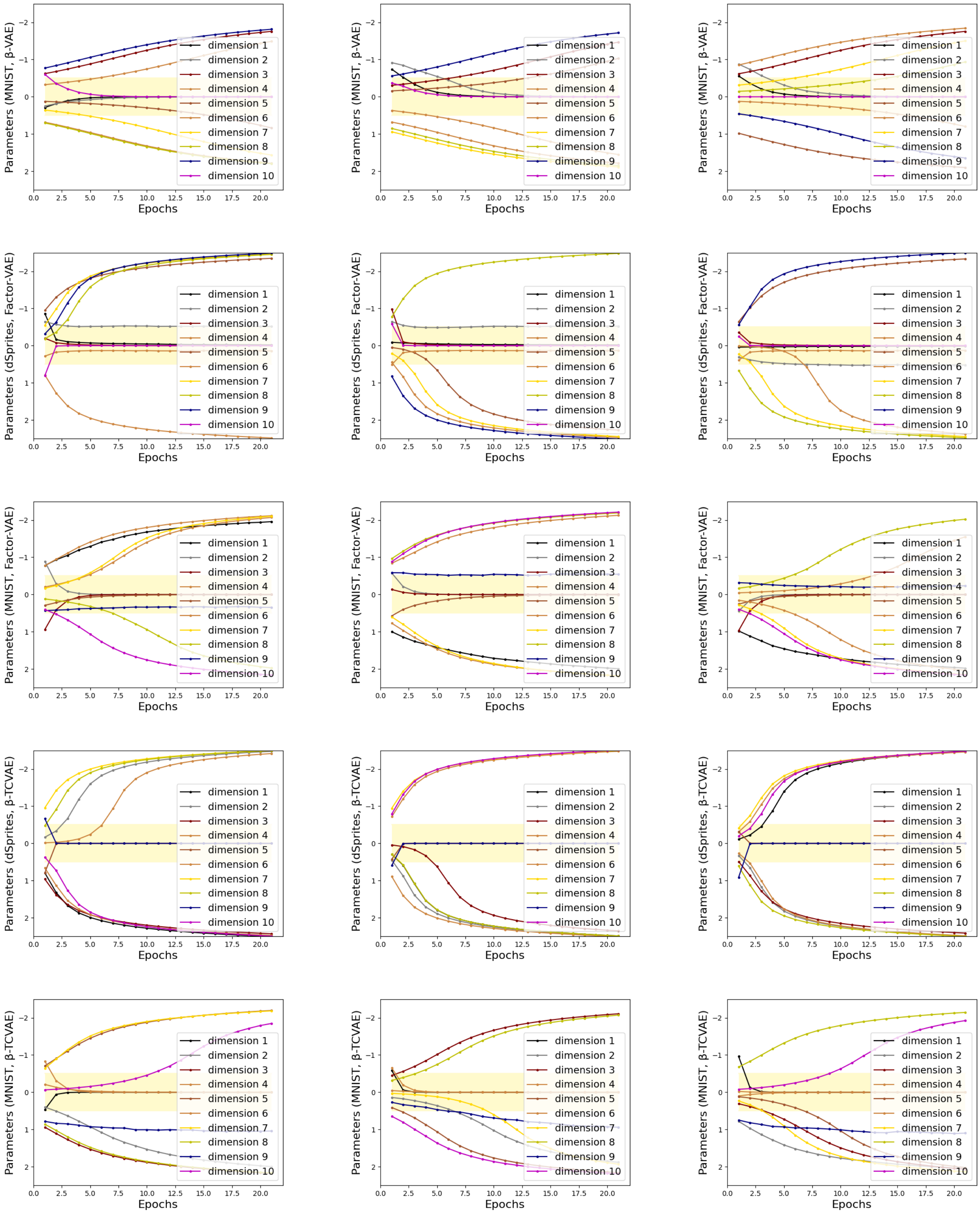


Figure 8. The training sets of dSprites and MNIST are divided into 3 subsets, respectively. The above illustration is the parameter curve obtained by Indicator training on these subsets. In the illustration, the dimensions that fall into the yellow area are considered more concerned by the task model. In rows 1 and 2, the indicators mark the disentangled representations generated from β-VAE. Indicators in lines 3 and 4 mark the disentangled representations generated using Factor-VAE. Lines 5 and 6 are Indicator marking the disentangled representations generated using β-TCVAE.

4.2. Privacy–Utility Trade-Offs Evaluation and Comparison

4.2.1. Setup

We design experiments to verify the effectiveness of our framework's utility–privacy trade-offs on the real-world dataset CelebA. The images are normalized and resized to $3 \times 64 \times 64$ for preprocessing. Due to the better disentanglement of β-TCVAE, the β-TCVAE optimized by GAN is chosen to construct our partial privacy-preserving framework. The encoder and decoder are optimized using RmSprop, with $alpha$ and *eps* set to 0.9 and 1×10^{-8}, respectively. The discriminator is trained using an SGD optimizer, with *momentum* and *weight_decay* set to 0.9 and 1×10^{-4}, respectively. We train these three components for 40 epochs with a fixed learning rate of 3×10^{-4}, and the batch_size is set to 128. The dimension size of the disentangled representation is set to 128, which represents the output of the encoder, including 128 means and 128 variances. Indicator is trained using the SGD optimizer with 0.9 *momentum* and 1×10^{-4} *weight_decay* for 20 epochs with batch_size 256. The learning rate is set to 1×10^{-4}. Empirically, we set the hyperparameter λ to 2 and δ to 1. The classifier trained on the original data with the standard ResNet18 architecture is considered the task model.

In our experiments, the accuracy of the task model is used to quantify utility. Several attack models designed to infer privacy attributes are introduced, and we propose two new metrics as privacy measures. The data processed by the task model and the attack models come from the reconstruction of the decoder. Despite the introduction of GAN, the reconstructed data still inevitably loses details. In order to avoid exaggerating the protective effect of our privacy attribute framework due to the fuzziness of the reconstruction, we set the easily recognizable attributes as the platform for the privacy utility measurement. Specifically, we set "Eyeglasses" and "Gender" as the target attributes of the task model, while enumerating "Wearing_Hat" and "Bald" as the target for the attack models.

4.2.2. Baselines

We choose three classical privacy-preserving schemes that are widely used in the literature as a baseline against which to compare our framework. A brief description of these schemes is given below. Gaussian noise obfuscates the raw data by adding Gaussian noise $\mathcal{N}(0, \sigma^2)$, where σ is set to 0.5 and 1, respectively. Because Gaussian noise can provide rigorous differential privacy guarantees with less local noise, it is widely used in federated learning scenarios [30,50]. Laplacian noise is also a classic differential privacy method that injects Laplacian noise into the raw data according to the privacy budget $\{0.3, 0.9\}$. PAN is a representative framework for adversarial training methods [13]. In the training phase, PAN simulates adversaries interested in private information to obtain an encoder that can extract the representation with good utility–privacy trade-offs. In the comparison phase, the objective function adopts two sets of coefficients $\{0.1, 0.7, 0.2\}$, $\{0.5, 0.3, 0.2\}$ to show its performance under different privacy budgets.

4.2.3. Evaluation and Comparison

To quantify the data utility maintained by different schemes, the classification accuracy of the target attribute is measured. Specifically, the two noise injection methods and our framework use classifiers trained on the raw dataset, while PAN uses the utility discriminator generated in the adversarial training. In terms of privacy measurement, the accuracy of the adversarial model's inference of privacy attributes is not convincing. This is because the model's decision is biased and the test set samples may be uneven, which means that lower accuracy does not necessarily mean better privacy protection. Taking the "bald" attribute as an example, the uniform random noise with a value in the range $[0, 1]$ will be 100% judged as not bald by the adversary model with an average confidence of 0.99. Although these noises are unrelated to the raw data, they will still achieve 97.88% inference accuracy when considered as a processed private image. In addition, the confidence difference in the attack models in inferring private images will reveal additional information compared to inferring random noise. Therefore, we propose the average confidence difference *Con-Diff* and the distribution shift *Dis-Shift* as the privacy quantification. The two formulas are defined as follows:

$$Con\_Diff = \frac{1}{N}\sum_{i=1}^{N}(AM^{l}(x'^{(i)})^{(0)} - AM^{l}(noise^{(i)})^{(0)}) \tag{9}$$

$$Dis\_Shift = \frac{1}{N}\sum_{i=1}^{N}|AM(x'^{(i)}) - AM(noise^{(i)})| \tag{10}$$

Among them, AM is the attack model with l layers, x' represents private images generated by different methods, and N is the total number of samples in the test set. $AM^{l}(\cdot)^{(0)}$ indicates the first element output by the AM, and $AM(\cdot)$ represents class 0 or 1. The lower *Con-Diff* and *Dis-Shift* represent that the attack model's inference on the processed data privacy attributes is closer to a non-priority guess, which demonstrates better privacy considerations.

In Figure 9, we use t-SNE [51] to visualize the features learned by the attack model at layer l-1 to analyze the effectiveness of our framework. The first column is the t-SNE plot of the original image facing the attack model. The second and third columns show the t-SNE plot for anonymously transformed reconstruction with "Eyeglasses" and "Gender" as the task-related attributes. The original data features show significant clustering for the two task-independent attributes, while the features of anonymously transformed reconstruction are indistinguishable.

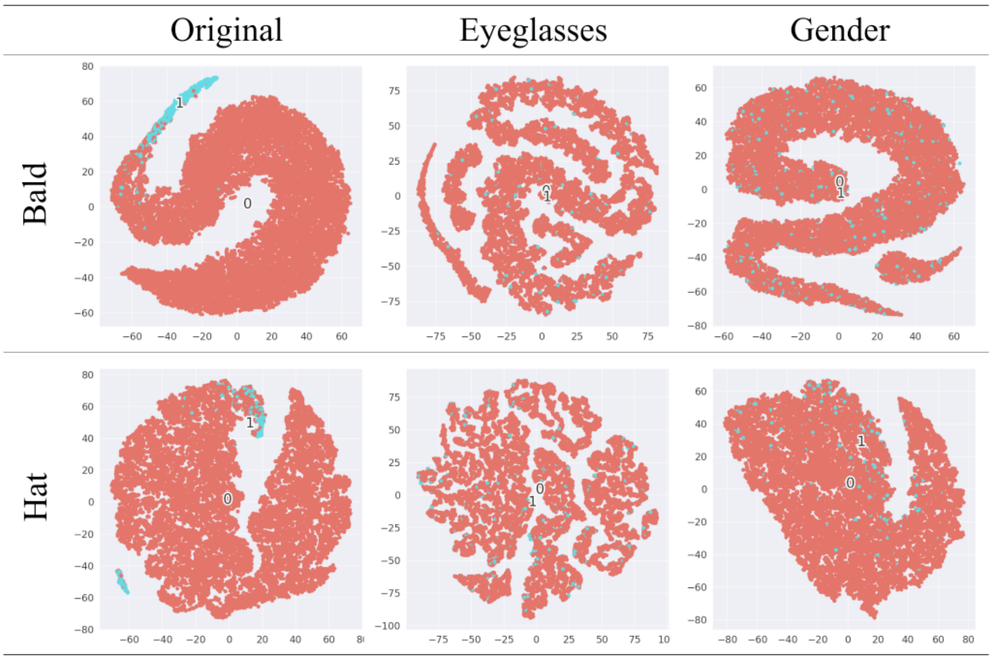
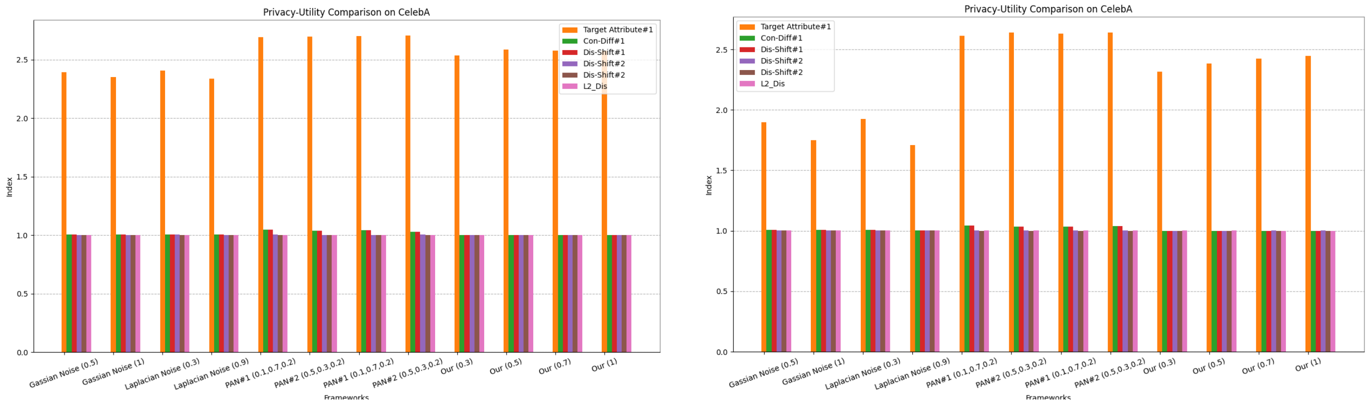


Figure 9. The t-SNE visualization of the AM^{l-1} output. The first column represents the performance of the original data in the face of the attack model. The second and third columns are the anonymously transformed reconstruction performance facing the attack model, with "Eyeglasses" and "Gender" as the task-related attributes, respectively.

Table 1 shows the evaluation and comparison of different methods on the utility–privacy trade-offs. It also includes L2 distance to measure the similarity between the processed image and the original data. "Target Attribute #1" and "Target Attribute #2" represent "Eyeglasses" and "Gender", while "Privacy Attribute #1" and "Privacy Attribute #2" represent "Wearing_Hat" and "Bald". Injecting Gaussian noise and Laplacian noise are general methods that will affect all attributes indiscriminately. Therefore, the privacy protection of these methods will significantly sacrifice its utility. In addition, L2-DIS indicate that the processed image still has a high similarity to the original data, which will also lead to the risk of privacy leakage. For the evaluation of PAN, we follow its recommendation on the encoder structure and design 4 convolutional layers, 4 normalization layers, 2 Maxpooling, and 2 upsampling layers. After 15 epochs of training, PAN achieves the ideal utility, but more discussion about privacy is necessary. In our experiment, the attack model for privacy attribute #1 will classify uniform random noise as class "1" with 100% probability, and 3.3% of the samples in the test set are class "1". At the same time, the attack model for privacy attribute #2 will infer uniform random noise as class "0" with 100% probability, and 97.9% of the test set are "0" samples. Under different experimental settings, some samples fool the attack model's judgment on privacy attribute #1, but have little effect on privacy attribute #2. Similarly, for these two privacy attributes, the attack model's judgments on the encrypted data are closer to its judgments on random noise. The mechanism of classifiers based on neural networks can be simply described as being oriented to data distribution. The nonlinear transformation of the original data by the encoder in PAN essentially causes a distribution shift. From our point of view, this is the privacy guarantee of PAN. However, there are still a large number of samples that reveal the privacy attributes. The evaluation of our framework uses the inference results of the attack model on the carrier as a benchmark. While the reconstructed image retains utility, there is almost no difference in confidence and distribution shift compared to the carrier. This shows that the reconstructed image produces a low-level information gain for the attack model, demonstrating the privacy of our framework. In order to compare different frameworks more intuitively, we further describe the evaluation results in Figure 10. The visualization of the reconstructed images shown in Figure 11 supports the evaluation results. The upper part of each sub-region in Figure 9 are the original images, and the lower part are the reconstructed images, which retain the target attribute while others still belong to the carrier. Since the reconstructed images are also facial images, the structure is similar to the original image. Another advantage of our framework is its flexibility. When the target or privacy attributes change, our framework needs to retrain 128 parameters, but PAN needs to retrain 22.44 M.



(**a**) Comparison on Target Attribute #1 (**b**) Comparison on Target Attribute #1

Figure 10. Privacy–utility comparison on CelebA. Among them, the y-axis takes the $exp(\cdot)$ of the evaluation result.

Using the privacy attribute #1 as a platform, we further explore potential attackers against the baseline methods and our framework. More powerful attack models are trained using the privacy-edited data as input and combining the ground truth. It should be noted that these attacks may not be feasible in real-world scenarios, and we aim to explore whether the above methods can effectively confuse the original data. The experimental results are reported in Table 2. The new attack models do not perform more effective attacks against the two noise injection methods and our framework, which shows that the topological space of the original data is broken. On the contrary, PAN is vulnerable to new attacks, supporting the suspicion mentioned in Section 1.

Table 1. Privacy–utility comparison on CelebA.

Methods	Target Attribute #1	Privacy Attribute #1		Privacy Attribute #2		L2-DIS
		Dis-Shift	Con-Diff	Dis-Shift	Con-Diff	
Gaussian noise ($\sigma^2 = 0.5$)	87.3%	2.45%	6.11×10^{-2}	0.01%	1.90×10^{-2}	6.19×10^{-4}
Gaussian noise ($\sigma^2 = 1$)	85.5%	0.31%	5.05×10^{-2}	0.02%	1.85×10^{-2}	8.04×10^{-4}
Laplacian noise ($\lambda = 0.3$)	87.8%	12.8%	5.51×10^{-2}	0.03%	1.91×10^{-2}	5.82×10^{-4}
Laplacian noise ($\lambda = 0.9$)	85.0%	3.53%	3.49×10^{-2}	0.01%	1.80×10^{-2}	8.06×10^{-4}
PAN (#1, $(0.1, 0.7, 0.2)$)	99.1%	50.7%	4.72×10^{-1}	0.05%	1.33×10^{-4}	1.52×10^{-3}
PAN (#1, $(0.5, 0.3, 0.2)$)	99.3%	37.3%	3.53×10^{-1}	0.02%	1.89×10^{-4}	1.95×10^{-3}
PAN (#2, $(0.1, 0.7, 0.2)$)	99.4%	41.4%	4.12×10^{-1}	0.02%	3.49×10^{-4}	1.26×10^{-3}
PAN (#2, $(0.5, 0.3, 0.2)$)	99.5%	16.1%	2.73×10^{-1}	0.05%	3.18×10^{-4}	1.69×10^{-3}
Our framework ($\psi = 0.3$)	93.1%	**0%**	$\mathbf{1.32 \times 10^{-6}}$	**0%**	**0**	$\mathbf{9.86 \times 10^{-4}}$
Our framework ($\psi = 0.5$)	95.0%	**0%**	$\mathbf{1.81 \times 10^{-6}}$	**0%**	**0**	$\mathbf{8.96 \times 10^{-4}}$
Our framework ($\psi = 0.7$)	94.6%	**0%**	$\mathbf{1.44 \times 10^{-6}}$	**0%**	**0**	$\mathbf{8.44 \times 10^{-4}}$
Our framework ($\psi = 1$)	94.6%	**0%**	$\mathbf{3.15 \times 10^{-6}}$	**0%**	$\mathbf{2.60 \times 10^{-10}}$	$\mathbf{7.52 \times 10^{-4}}$
Methods	**Target Attribute #2**	**Privacy Attribute #1**		**Privacy Attribute #2**		**L2-DIS**
		Dis-Shift	**Con-Diff**	**Dis-Shift**	**Con-Diff**	
Gaussian noise ($\sigma^2 = 0.5$)	64.0%	2.45%	6.11×10^{-2}	0.01%	1.90×10^{-2}	6.19×10^{-2}
Gaussian noise ($\sigma^2 = 1$)	55.8%	0.31%	5.05×10^{-2}	0.02%	1.85×10^{-2}	8.04×10^{-4}
Laplacian noise ($\lambda = 0.3$)	65.4%	12.8%	5.51×10^{-2}	0.03%	1.91×10^{-2}	5.82×10^{-4}
Laplacian noise ($\lambda = 0.9$)	53.7%	3.53%	3.49×10^{-2}	0.01%	1.80×10^{-2}	8.06×10^{-4}
PAN (#1, $(0.1, 0.7, 0.2)$)	96.1%	46.9%	4.47×10^{-1}	0.15%	2.20×10^{-4}	1.05×10^{-3}
PAN (#1, $(0.5, 0.3, 0.2)$)	97.0%	43.3%	3.34×10^{-1}	0.22%	3.69×10^{-4}	1.05×10^{-3}
PAN (#2, $(0.1, 0.7, 0.2)$)	96.8%	28.5%	3.45×10^{-1}	0.27%	2.28×10^{-4}	1.06×10^{-3}
PAN (#2, $(0.5, 0.3, 0.2)$)	97.0%	35.6%	3.93×10^{-1}	0.15%	2.45×10^{-4}	1.05×10^{-3}
Our framework ($\psi = 0.3$)	84.0%	**0.02%**	$\mathbf{1.65 \times 10^{-6}}$	**0%**	**0**	$\mathbf{7.41 \times 10^{-4}}$
Our framework ($\psi = 0.5$)	86.9%	**0.03%**	$\mathbf{1.60 \times 10^{-6}}$	**0%**	**0**	$\mathbf{5.56 \times 10^{-4}}$
Our framework ($\psi = 0.7$)	88.6%	**0.05%**	$\mathbf{4.09 \times 10^{-6}}$	**0.12%**	$\mathbf{1.03 \times 10^{-7}}$	$\mathbf{3.29 \times 10^{-4}}$
Our framework ($\psi = 1$)	89.5%	**0.78%**	$\mathbf{1.61 \times 10^{-4}}$	**0.40%**	$\mathbf{2.92 \times 10^{-5}}$	$\mathbf{2.21 \times 10^{-4}}$

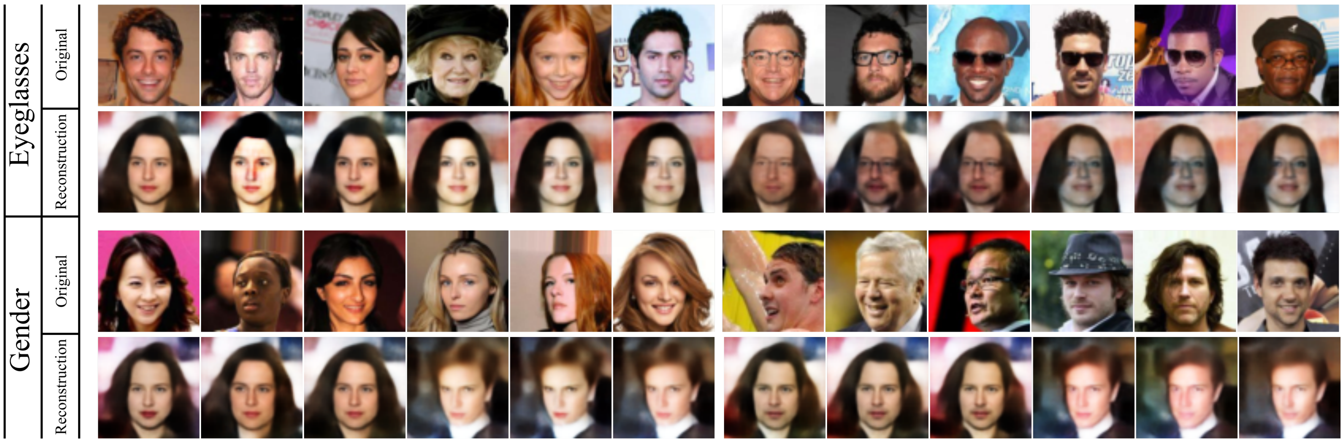


Figure 11. The above illustrations are facial images whose task-independent attributes are confused. The upper part takes "Eyeglasses" as the task-related attribute, and the bottom part, "Gender" is regarded as the task-related attribute.

To further show the effectiveness of our framework, we also use SVM for experiments. We choose the RBF kernel function, use libsvm to set the hyperparameters and let the latent code be the input. Evaluation results are shown in Table 3.

Table 2. Search for potential attacker.

Methods	**Dis-Shift (Original Attacker)**	**Dis-Shift (Potential Attacker)**
Gaussian noise ($\sigma^2 = 0.5$)	2.45%	3.76%
Gaussian noise ($\sigma^2 = 1$)	0.31%	3.51%
Laplacian noise ($\lambda = 0.3$)	12.8%	4.44%
Laplacian noise ($\lambda = 0.9$)	3.53%	3.14%
PAN $(0.5, 0.3, 0.2)$	43.3%	95.52%
PAN $(0.1, 0.7, 0.2)$	46.9%	96.74%
Our framework ($\psi = 0.5$)	**0.03**%	**0.05**%
Our framework ($\psi = 1$)	**0.78**%	**0.91**%

Table 3. Evaluating privacy–utility on CelebA via SVM.

Method	**Target Attribute #1**	**Privacy Attribute #1**	**Privacy Attribute #2**
Our framework ($\psi = 0.3$)	84.3%	43.1%	54.2%
Our framework ($\psi = 0.5$)	85.9%	46.3%	55.6%
Our framework ($\psi = 0.7$)	87.2%	48.4%	56.9%
Our framework ($\psi = 1$)	87.5%	50.3%	58.4%
Method	**Target Attribute #1**	**Privacy Attribute #1**	**Privacy Attribute #2**
Our framework ($\psi = 0.3$)	72.5%	44.5%	54.6%
Our framework ($\psi = 0.5$)	73.2%	47.8%	55.8%
Our framework ($\psi = 0.7$)	73.7%	47.7%	57.2%
Our framework ($\psi = 1$)	74.1%	48.4%	59.1%

Based on the above experimental results, on the one hand, it can be observed that our framework can be more effective against potential attackers compared to adversarial training based PAN. On the other hand, our framework maintains better data availability as well as privacy of task-independent attributes compared to the noise adding approach.

5. Discussion and Conclusions

In this work, we design an Indicator to indicate the region of interest of the target task model on the disentangled representation. By retaining the information necessary for the target task through Indicator, we further construct a privacy-preserving prediction framework that respects the task-independent attributes. Evaluations on multiple standard datasets show that our framework achieves competitive utility–privacy trade-offs.

However, our framework has not yet reached the ideal situation of preserving all utility and protecting all privacy. On the one hand, our framework partially loses accuracy in the target task. On the other hand, the attacker's accuracy in inferring privacy attributes is higher than the guess without prior knowledge. We speculate that there are two reasons: (a) the quality of the reconstructed image limits the utility; (b) there is information overlap between the different representation dimensions, leading to sensitive information leakage. These are also problems that we hope to solve in the future.

Author Contributions: Conceptualization, H.L. and Q.L.; methodology, H.L.; software, Q.L.; validation, H.L.; formal analysis, Q.L.; investigation, Q.L.; resources, H.L.; data curation, Q.L.; writing—original draft preparation, Q.L.; writing—review and editing, Q.L.; visualization, Q.L.; supervision, H.L.; project administration, H.L.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Our experiments use the public datasets MNIST, dSprites and CelebA.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Burstein, J., Doran, C., Solorio, T., Eds.; Long and Short Papers; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; Volume 1, pp. 4171–4186. [CrossRef]
2. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929. Available online: http://arxiv.org/abs/2010.11929 (accessed on 6 December 2023).
3. Li, A.; Guo, J.; Yang, H.; Salim, F.D.; Chen, Y. DeepObfuscator: Obfuscating Intermediate Representations with Privacy-Preserving Adversarial Learning on Smartphones. In Proceedings of the IoTDI'21: International Conference on Internet-of-Things Design and Implementation, Charlottesville, VA, USA, 18–21 May 2021; ACM: New York, NY, USA, 2021; pp. 28–39. [CrossRef]
4. Ribeiro, M.; Grolinger, K.; Capretz, M.A. MLaaS: Machine Learning as a Service. In Proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 9–11 December 2015; pp. 896–902. [CrossRef]
5. Achille, A.; Soatto, S. Emergence of Invariance and Disentanglement in Deep Representations. *J. Mach. Learn. Res.* **2018**, *19*, 1947–1980.
6. Google. Google Now Launcher. 2018. Available online: https://en.wikipedia.org/wiki/Google_Now (accessed on 6 December 2023).
7. Google. Data Preparation. 2018. Available online: https://cloud.google.com/ml-engine/docs/tensorflow/data-prep (accessed on 6 December 2023).
8. Fredrikson, M.; Jha, S.; Ristenpart, T. Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures. In Proceedings of the CCS'15, 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; pp. 1322–1333. [CrossRef]
9. Mahendran, A.; Vedaldi, A. Understanding deep image representations by inverting them. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5188–5196. [CrossRef]
10. Hidano, S.; Murakami, T.; Katsumata, S.; Kiyomoto, S.; Hanaoka, G. Model Inversion Attacks for Prediction Systems: Without Knowledge of Non-Sensitive Attributes. In Proceedings of the 2017 15th Annual Conference on Privacy, Security and Trust (PST), Calgary, AB, Canada, 28–30 August 2017; pp. 115–11509. [CrossRef]
11. Osia, S.A.; Shahin Shamsabadi, A.; Sajadmanesh, S.; Taheri, A.; Katevas, K.; Rabiee, H.R.; Lane, N.D.; Haddadi, H. A Hybrid Deep Learning Architecture for Privacy-Preserving Mobile Analytics. *IEEE Internet Things J.* **2020**, *7*, 4505–4518. [CrossRef]
12. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2014; Volume 27.
13. Liu, S.; Du, J.; Shrivastava, A.; Zhong, L. Privacy Adversarial Network: Representation Learning for Mobile Data Privacy. *Proc. Acm Interact. Mobile, Wearable Ubiquitous Technol.* **2019**, *3*, 144. [CrossRef]
14. Li, A.; Duan, Y.; Yang, H.; Chen, Y.; Yang, J. TIPRDC: Task-Independent Privacy-Respecting Data Crowdsourcing Framework for Deep Learning with Anonymized Intermediate Representations. In Proceedings of the KDD'20, 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual, 6–10 July 2020; pp. 824–832. [CrossRef]
15. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning Deep Features for Discriminative Localization. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2921–2929. [CrossRef]
16. Chattopadhay, A.; Sarkar, A.; Howlader, P.; Balasubramanian, V.N. Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 839–847. [CrossRef]
17. Zhang, Q.; Rao, L.; Yang, Y. Group-CAM: Group Score-Weighted Visual Explanations for Deep Convolutional Networks. *arXiv* **2021**, arXiv:2103.13859. Available online: http://arxiv.org/abs/2103.13859 (accessed on 6 December 2023).
18. Zhang, Q.; Wang, X.; Wu, Y.N.; Zhou, H.; Zhu, S.C. Interpretable CNNs for Object Classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 3416–3431. [CrossRef] [PubMed]
19. Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; Lerchner, A. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017; Conference Track Proceedings. Available online: http://OpenReview.net (accessed on 6 December 2023).
20. Kim, H.; Mnih, A. Disentangling by Factorising. In Proceedings of the Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018; Dy, J.G., Krause, A., Eds.; Proceedings of Machine Learning Research: Breckenridge, CO, USA, 2018; Volume 80, pp. 2654–2663.
21. Chen, T.Q.; Li, X.; Grosse, R.B.; Duvenaud, D. Isolating Sources of Disentanglement in Variational Autoencoders. In *Advances in Neural Information Processing Systems 31, Proceedings of theAnnual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, Montréal, QC, Canada, 3–8 December 2018*; Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Neural Information Processing Systems: La Jolla, CA, USA, 2018; pp. 2615–2625.

22. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 29, Proceedings of theAnnual Conference on Neural Information Processing Systems 2016, Barcelona, Spain, 5–10 December 2016*; Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R., Eds.; Neural Information Processing Systems: La Jolla, CA, USA, 2016; pp. 2172–2180.
23. Kingma, D.P.; Dhariwal, P. Glow: Generative Flow with Invertible 1x1 Convolutions. In *Advances in Neural Information Processing Systems 31, Proceedings of the Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, Montréal, QC, Canada 3–8 December 2018*; Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Neural Information Processing Systems: La Jolla, CA, USA, 2018; pp. 10236–10245.
24. Sweeney, L. k-Anonymity: A Model for Protecting Privacy. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* **2002**, *10*, 557–570. [CrossRef]
25. Machanavajjhala, A.; Gehrke, J.; Kifer, D.; Venkitasubramaniam, M. L-diversity: Privacy beyond k-anonymity. In Proceedings of the 22nd International Conference on Data Engineering (ICDE'06), Atlanta, GA, USA, 3–7 April 2006; pp. 24–24. [CrossRef]
26. Li, N.; Li, T.; Venkatasubramanian, S. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering, Atlanta, GA, USA, 3–7 April 2006; pp. 106–115. [CrossRef]
27. Dwork, C.; Roth, A. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* **2014**, *9*, 211–407. [CrossRef]
28. Mironov, I. Rényi Differential Privacy. In Proceedings of the 2017 IEEE 30th Computer Security Foundations Symposium (CSF), Santa Barbara, CA, USA, 21–25 August 2017; pp. 263–275. [CrossRef]
29. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep Learning with Differential Privacy. In Proceedings of the CCS'16, 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 308–318. [CrossRef]
30. Papernot, N.; Song, S.; Mironov, I.; Raghunathan, A.; Talwar, K.; Erlingsson, Ú. Scalable Private Learning with PATE. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, 30 April–3 May 2018; Conference Track Proceedings. Available online: http://OpenReview.net (accessed on 6 December 2023).
31. Oh, S.J.; Benenson, R.; Fritz, M.; Schiele, B. Faceless Person Recognition: Privacy Implications in Social Media. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Cham, Switzerland, 2016; pp. 19–35.
32. Dowlin, N.; Gilad-Bachrach, R.; Laine, K.; Lauter, K.; Naehrig, M.; Wernsing, J. CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy. In Proceedings of the ICML'16 33rd International Conference on International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; Volume 48, pp. 201–210. Available online: http://JMLR.org (accessed on 6 December 2023).
33. Li, J.; Kuang, X.; Lin, S.; Ma, X.; Tang, Y. Privacy preservation for machine learning training and classification based on homomorphic encryption schemes. *Inf. Sci.* **2020**, *526*, 166–179. [CrossRef]
34. Riazi, M.S.; Weinert, C.; Tkachenko, O.; Songhori, E.M.; Schneider, T.; Koushanfar, F. Chameleon: A Hybrid Secure Computation Framework for Machine Learning Applications. In Proceedings of the ASIACCS'18, 2018 on Asia Conference on Computer and Communications Security, Incheon, Republic of Korea, 4 June 2018; pp. 707–721. [CrossRef]
35. Liu, J.; Juuti, M.; Lu, Y.; Asokan, N. Oblivious Neural Network Predictions via MiniONN Transformations. In Proceedings of the CCS'17, Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX USA, 30 October–3 November 2017; pp. 619–631. [CrossRef]
36. Mohassel, P.; Zhang, Y. SecureML: A System for Scalable Privacy-Preserving Machine Learning. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 19–38. [CrossRef]
37. Yu, J.; Zhang, B.; Kuang, Z.; Lin, D.; Fan, J. iPrivacy: Image Privacy Protection by Identifying Sensitive Objects via Deep Multi-Task Learning. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 1005–1016. [CrossRef]
38. Malekzadeh, M.; Clegg, R.G.; Haddadi, H. Replacement AutoEncoder: A Privacy-Preserving Algorithm for Sensory Data Analysis. In Proceedings of the 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI), Orlando, FL, USA, 17–20 April 2018; pp. 165–176. [CrossRef]
39. Aloufi, R.; Haddadi, H.; Boyle, D. Privacy-Preserving Voice Analysis via Disentangled Representations. In Proceedings of the CCSW'20, 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop, Virtual, 9 November 2020; pp. 1–14. [CrossRef]
40. van den Oord, A.; Vinyals, O.; kavukcuoglu, K. Neural Discrete Representation Learning. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
41. Kalchbrenner, N.; Elsen, E.; Simonyan, K.; Noury, S.; Casagrande, N.; Lockhart, E.; Stimberg, F.; van den Oord, A.; Dieleman, S.; Kavukcuoglu, K. Efficient Neural Audio Synthesis. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018; Dy, J.G., Krause, A., Eds.; Proceedings of Machine Learning Research: Breckenridge, CO, USA, 2018; Volume 80, pp. 2415–2424.
42. Gong, M.; Liu, J.; Li, H.; Xie, Y.; Tang, Z. Disentangled Representation Learning for Multiple Attributes Preserving Face Deidentification. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *33*, 244–256. [CrossRef]

43. Wu, H.; Tian, X.; Li, M.; Liu, Y.; Ananthanarayanan, G.; Xu, F.; Zhong, S. PECAM: Privacy-Enhanced Video Streaming and Analytics via Securely Reversible Transformation. In Proceedings of the MobiCom'21, 27th Annual International Conference on Mobile Computing and Networking, New Orleans, Louisiana, 25–29 October 2021; pp. 229–241. [CrossRef]
44. Jia, J.; Gong, N.Z. Attriguard: A Practical Defense against Attribute Inference Attacks via Adversarial Machine Learning. In Proceedings of the SEC'18, 27th USENIX Conference on Security Symposium, Baltimore, MD, USA, 15–17 August 2018; pp. 513–529.
45. Wu, Z.; Wang, Z.; Wang, Z.; Jin, H. Towards Privacy-Preserving Visual Recognition via Adversarial Training: A Pilot Study. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer: Cham, Switzerland, 2018; pp. 627–645.
46. Larsen, A.B.L.; Sønderby, S.K.; Larochelle, H.; Winther, O. Autoencoding beyond Pixels Using a Learned Similarity Metric. In Proceedings of the ICML'16, 33rd International Conference on International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; Volume 48, pp. 1558–1566.
47. Matthey, L.; Higgins, I.; Hassabis, D.; Lerchner, A. dSprites: Disentanglement Testing Sprites Dataset. 2017. https://github.com/deepmind/dsprites-dataset/ (accessed on 6 December 2023).
48. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
49. Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep Learning Face Attributes in the Wild. In Proceedings of the International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
50. Truex, S.; Baracaldo, N.; Anwar, A.; Steinke, T.; Ludwig, H.; Zhang, R. A Hybrid Approach to Privacy-Preserving Federated Learning. In Proceedings of the AISec'19, 12th ACM Workshop on Artificial Intelligence and Security, London, UK, 15 November 2019; pp. 1–2.
51. van der Maaten, L.; Hinton, G. Viualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.