# Compressive Privacy Generative Adversarial Networks

Presenter: Bo-Wei Tseng

Advisor: Prof. Pei-Yuan Wu

Place: EE-II Room 504

# Outline

- Introduction
- Related works
    - Attack schemes in Machine Learning Model
    - Privacy preserving mechanism
        - Differential privacy
        - Homomorphic encryption
        - Compressive privacy
        - Gan-inspired model: GAP and RAN
- Methodology
    - Architecture
    - Objective Function and Algorithm.
    - Theoretical analysis
- Empirical results
- Conclusion and Future Works (include one page summary)

# Introduction

- Machine Learning as a service (MLaaS) raises the serious privacy issue.

**In real world application:**

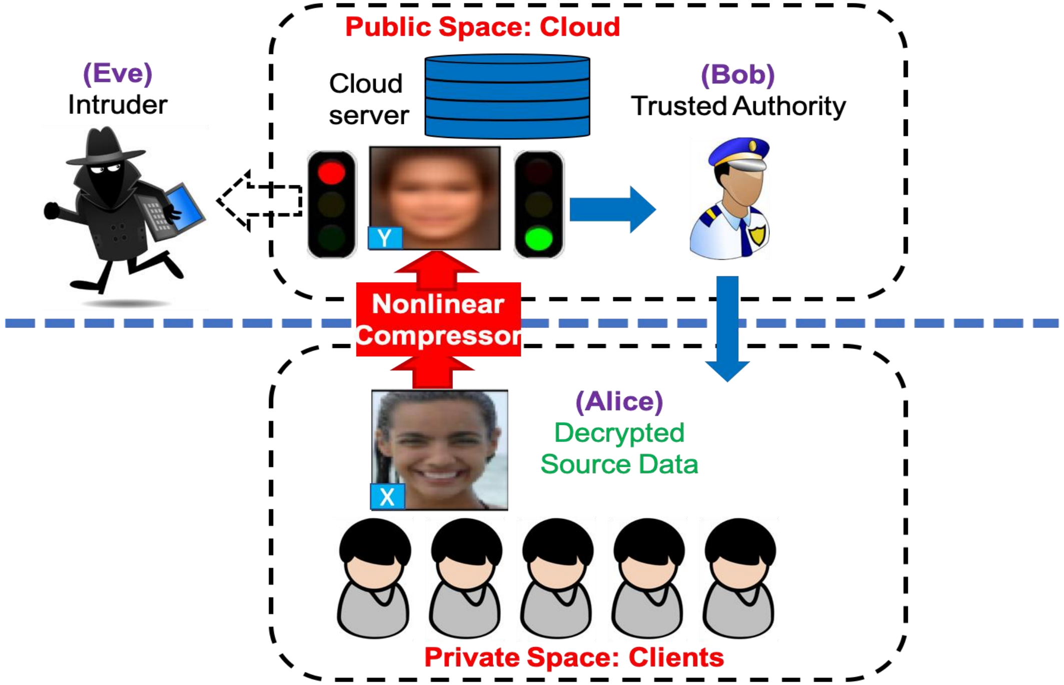

→ **Machine Learning Service** →

***Output:***

**Lipton's product**

**Yellow box in this picture leaks user's sensitive information.**

**Privatization mechanism must be applied in collaborative learning system**




Kung, S. Y. (2018). A Compressive Privacy approach to Generalized Information Bottleneck and Privacy Funnel problems. *Journal of the Franklin Institute*, *355*(4), 1846-1872.

# How Important ?

FORTUNE

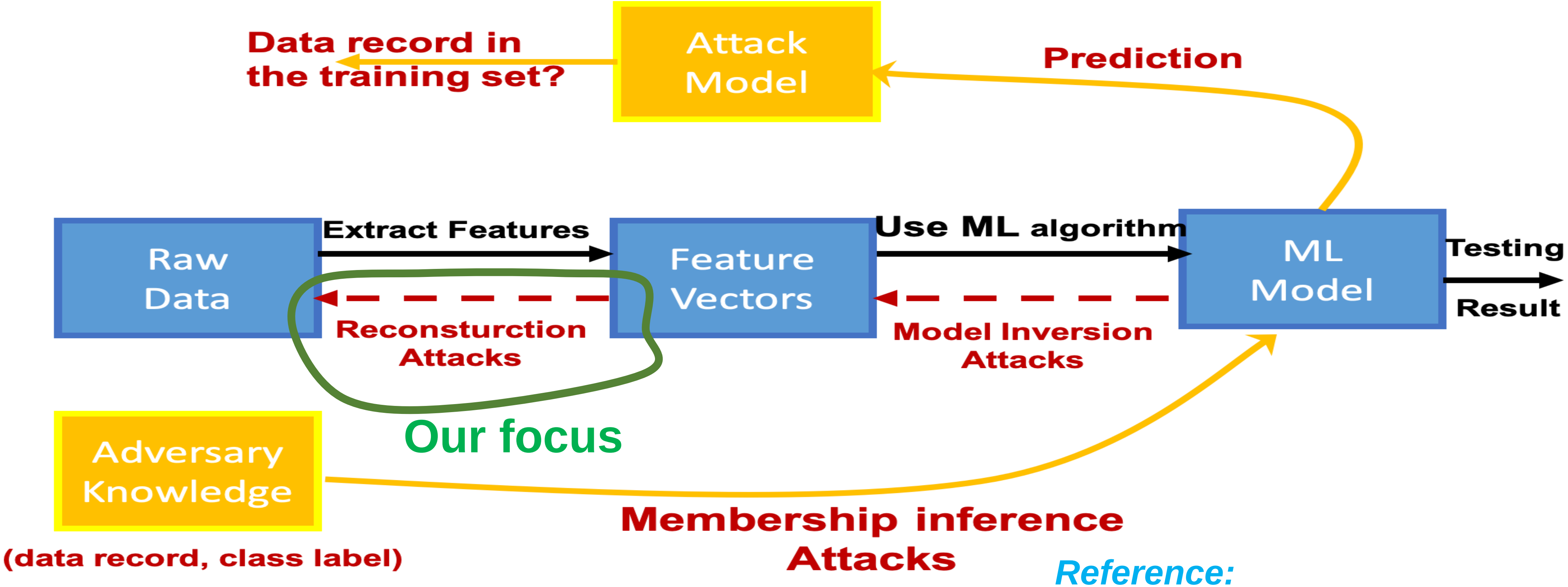
TECH • THE FUTURE OF WORK

**AI Has a Big Privacy Problem and Europe's New Data Protection Law Is About to Expose It**

-> Europe's new General Data Protection Regulation (GDPR)

Privacy is a **fundamental** human **right!**

# Attack schemes

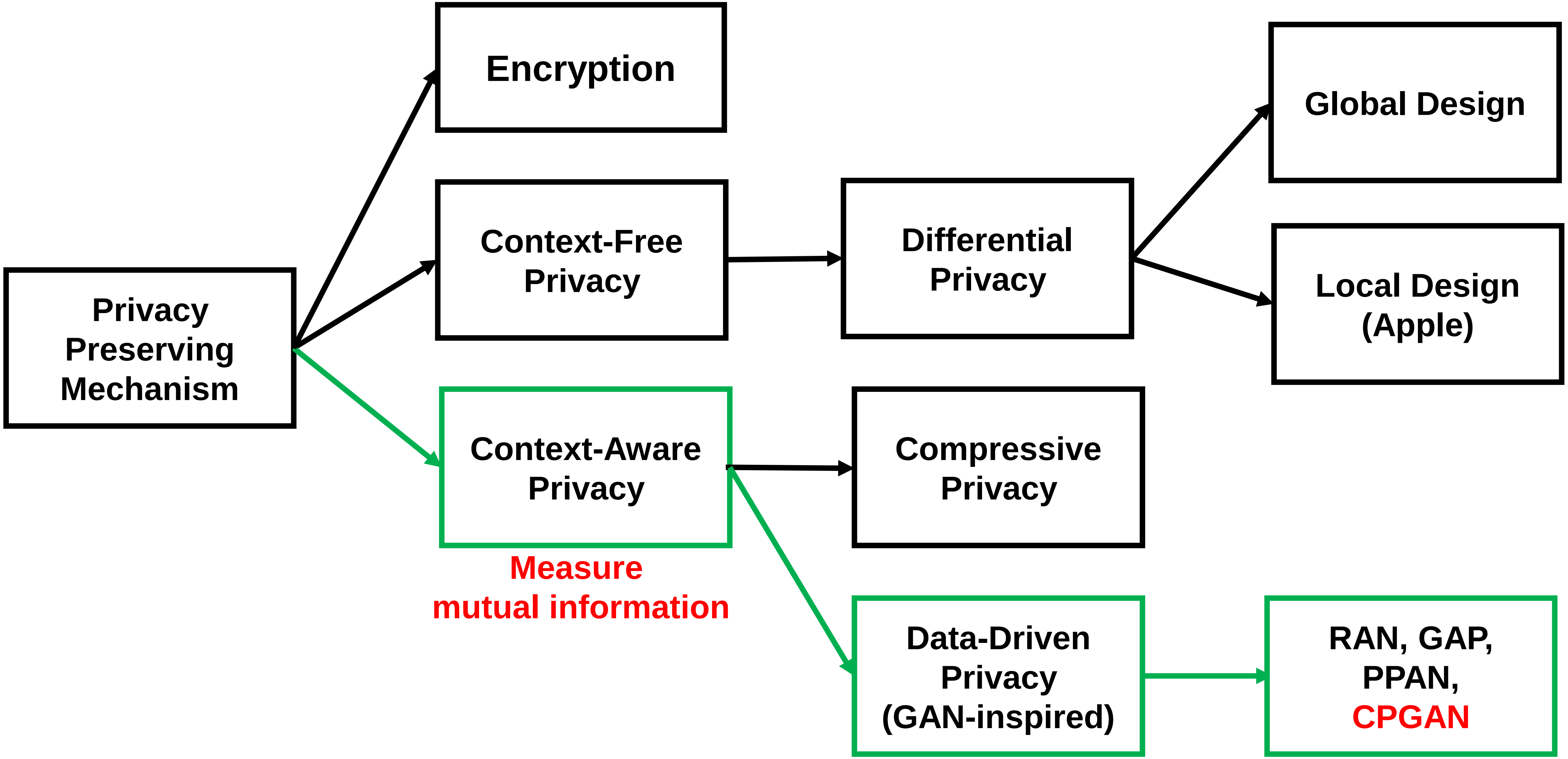



***Reference:***

R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, May 2017.

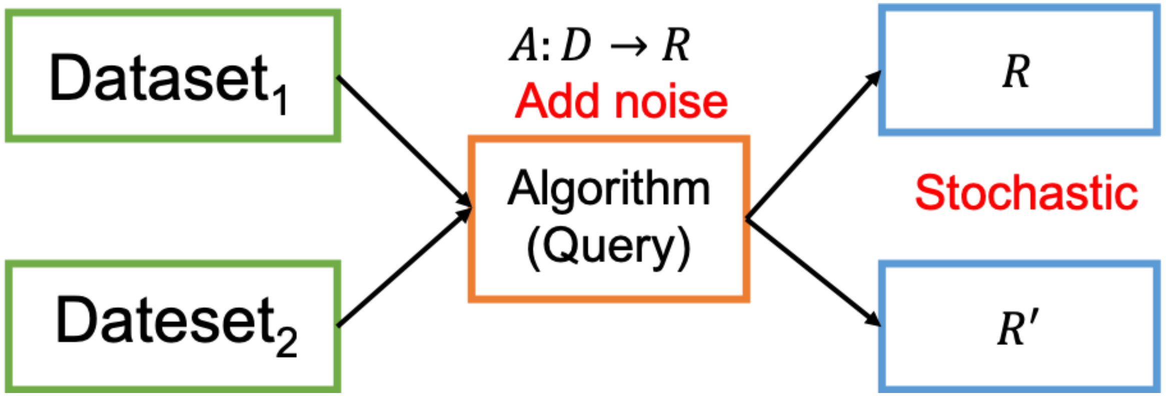
M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Computer and Communications Security*, pp. 1322–1333, ACM, 2015.

# Privacy Preserving mechanisms



# Differential Privacy (DP)

### *Global: if the aggregator is trustable.*




**Privacy guarantee formulation:**

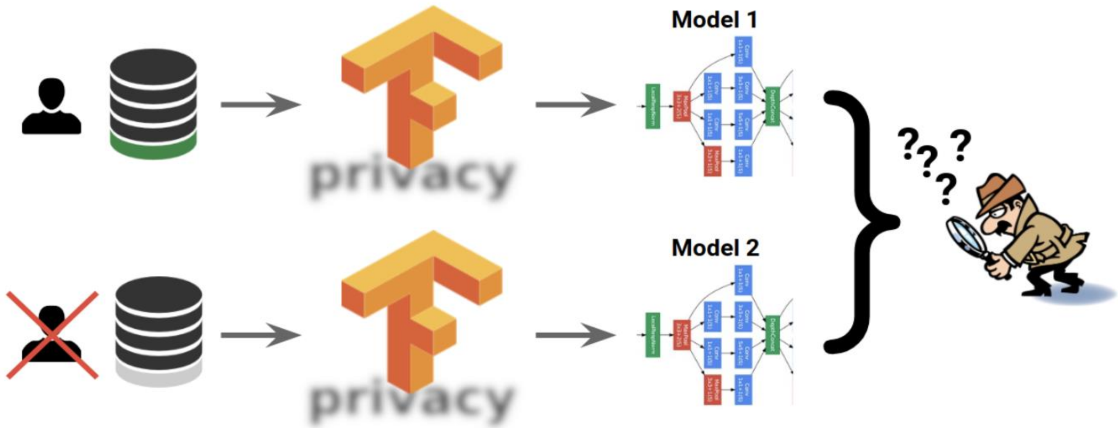
$$P(A(D_1) = o) \leq e^{\varepsilon} P(A(D_2) = o)$$

- $|D_1 - D_2| = 1$
- Popular used mechanism: Laplician($\frac{\Delta f}{\varepsilon}$)

**Composability (LDP):**

- Each $A_i$ satisfies $\varepsilon$-differential privacy, then for the n DP-mechanisms, it must become n$\varepsilon$-differential privacy.

### *DP in Deep Learning model*




- **May drop the utility of the model trained by DP optimization.**

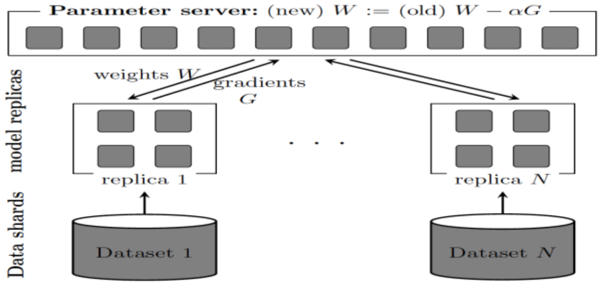
***Reference:***

- https://medium.com/tensorflow/introducing-tensorflow-privacy-learning-with-differential-privacy-for-training-data-b143c5e801b6

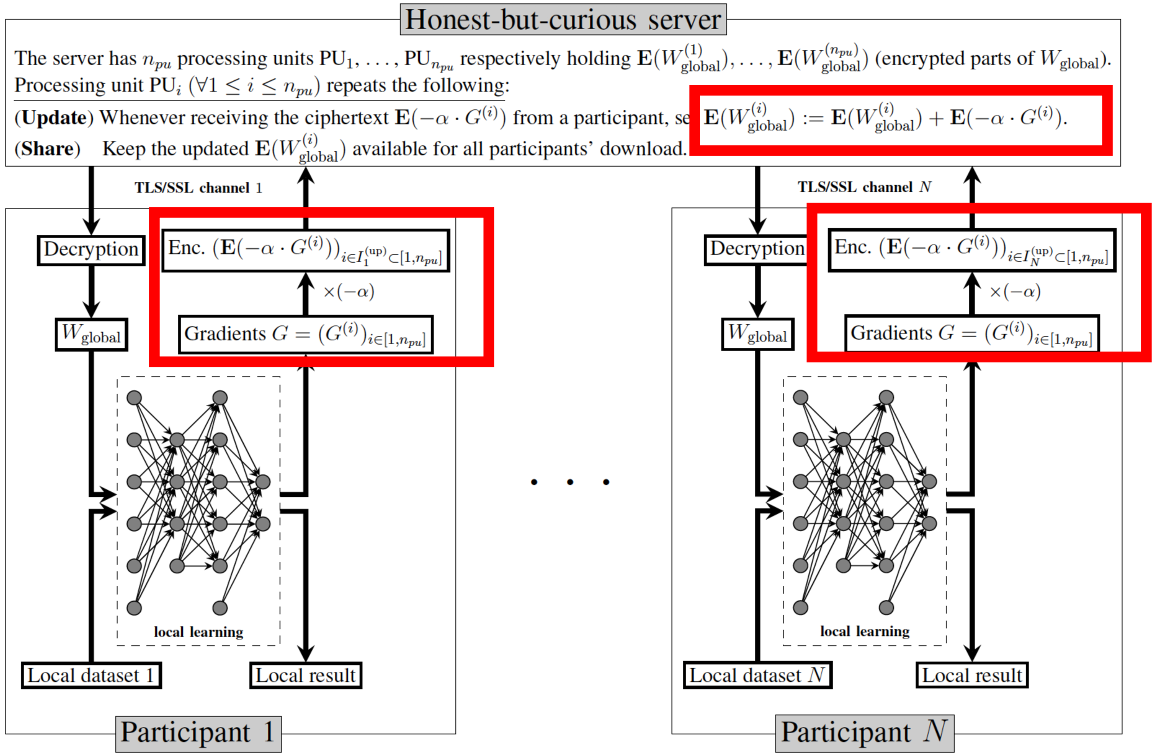
# Encryption in machine learning model

- In the training process, adversary reconstructs the user's private images using the gradients sent to the cloud.

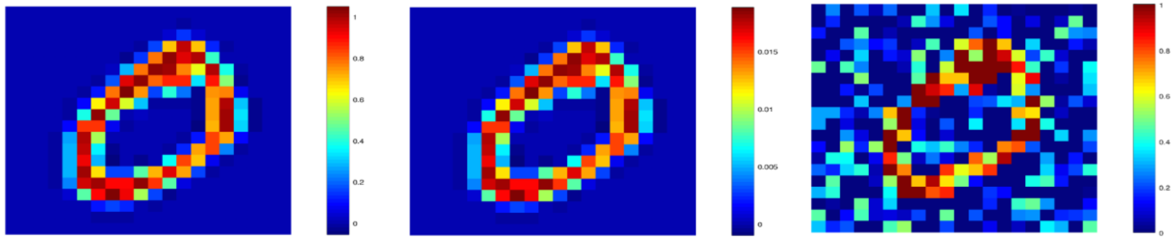
*Threat Model:*




**Adopt homomorphic encryption scheme.**




**Privacy leakage from gradients:**



(a) Original 20x20 image of handwritten number 0, seen as a vector over $\mathbb{R}^{400}$ fed to a neural network.

(b) Recovered image using 400/10285 (3.89%) gradients (see Sect.3, Example 2). The difference with the original (a) is only at the value bar.

(c) Recovered image using 400/10285 (3.89%) gradients (see Sect.3, Example 3). There are noises but the truth label 0 can still be seen.

*Reference:*

L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, May 2018.

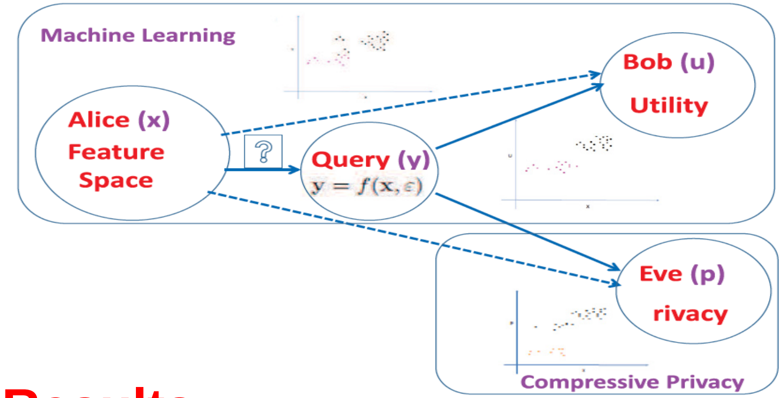
# Compressive Privacy (DCA、KDCA)

**Target:** Explore the low dimension representations ($y$) that retain high utility but low privacy information.

*Reference:*


S. Kung, T. Chanyaswad, J. Chang, and P.Y.Wu, "Collaborative pca/dca learning methods for compressive privacy," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, p. 76, 7 2017.

M. Al, T. Chanyaswad, and S. Y. Kung, "Multi-kernel, deep neural network and hybrid models for privacy preserving machine learning," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 2891–2895.


**DCA formulation:**

- $\max_{F:F^T[\bar{S}+\rho I]F=I} Tr(F^T S_{B_U} F)$

**KDCA formulation:**

- $\max_{F:F^T[\bar{K}^2+\rho \bar{K}]F=I} Tr(F^T K_{B_U} F)$
- Basic idea of Kernel Method: Map the original data to the RKHS space before applying DCA projection. And the inner product in RKHS is defined as $k(x,y) = \phi(x)^T\phi(y)$

**Shift invariant Kernel Function:**

- **RBF kernel**

$$k(x,y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$$

- **Laplician kernel:**

$$k(x,y) = e^{-\frac{\|x-y\|}{2\sigma}}$$

**Results:**

HAR: Experiment II

| | | Utility (%) | Privacy (%) |
|---|---|---|---|
| DCA | Random guess | 16.67 | 5.00 |
| | Compressive single linear kernel | 51.02 | 5.19 |
| KDCA | Compressive single RBF kernel | 86.20 | 6.48 |
| | Compressive single Laplacian kernel | 90.83 | 5.00 |
| | Compressive single sigmoid kernel | 82.59 | 7.04 |
| | Compressive uniform multi-kernel | 90.65 | 6.57 |
| | Compressive alignment-based multi-kernel | 91.30 | 6.57 |
| | Compressive SNR-based multi-kernel $\rho_{snr} = 0$ | 89.35 | 6.85 |
| | Compressive SNR-based multi-kernel $\rho_{snr} = 0.1$ | 91.39 | 5.00 |

TABLE II
HAR: UTILITY AND PRIVACY CLASSIFICATION ACCURACIES

# Generative Adversarial Privacy (GAP)

**GAN-inspired data-driven based model.**

***Reference:***

C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, "Generative adversarial privacy," *arxiv preprint arXiv:1807.05306*, 2018. [Online]. Available: http://arxiv.org/abs/1807.05306

**Utility**

- Distortion constraint

***Privacy:***

- Can not infer the specified sensitive attribute of this private images.

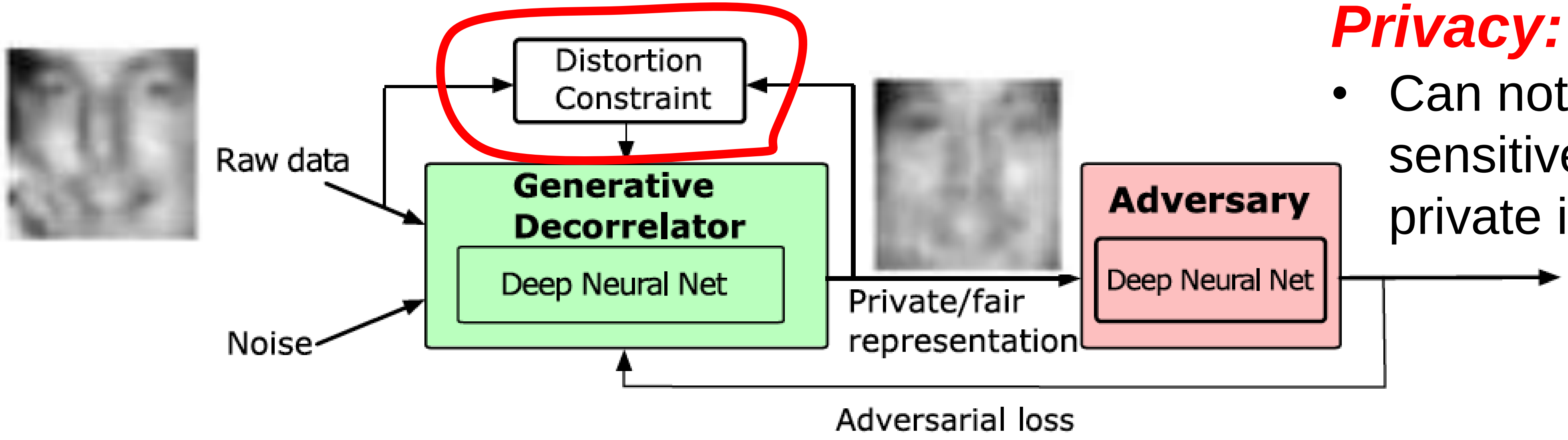



Figure 1: Generative adversarial model for privacy and fairness

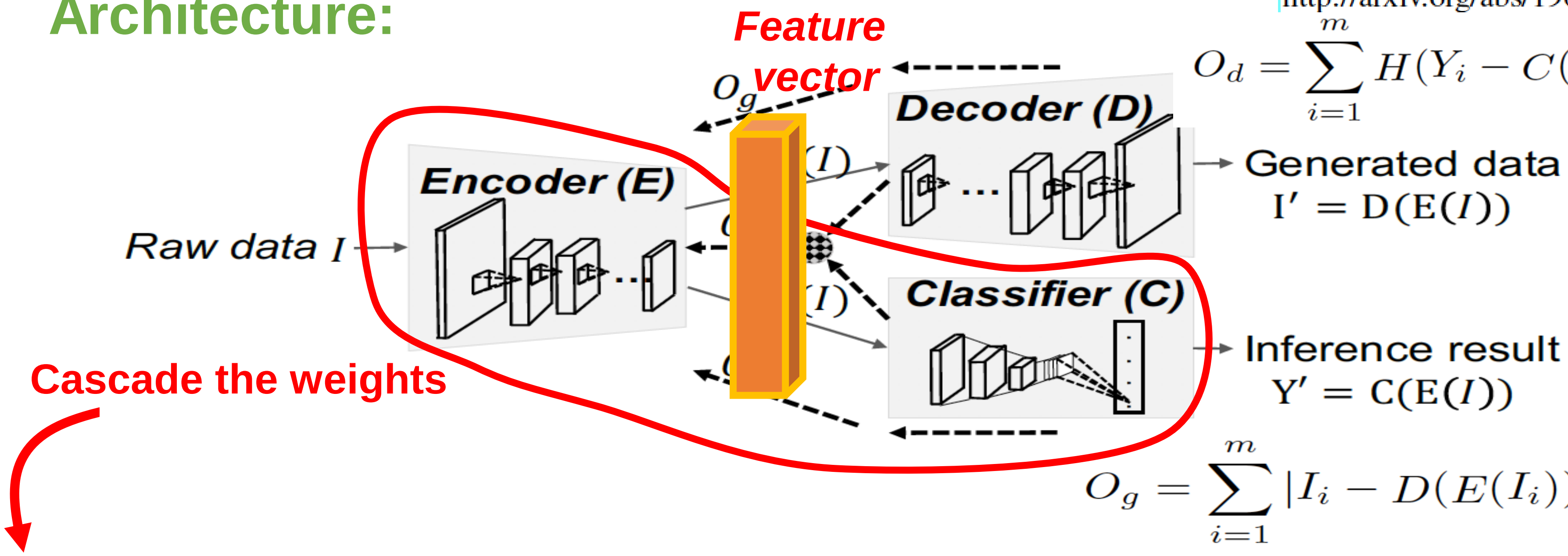
# Reconstructive adversarial network (RAN)

## GAN-inspired data-driven based model.

**Reference:**
S. Liu, A. Shrivastava, J. Du, and L. Zhong, "Better accuracy with quantified privacy: representations learned via reconstructive adversarial network," *arXiv preprint arXiv:1901.08730*, 2019. [Online]. Available: http://arxiv.org/abs/1901.08730

**Architecture:**

$$O_d = \sum_{i=1}^{m} H(Y_i - C(E(I_i))).$$

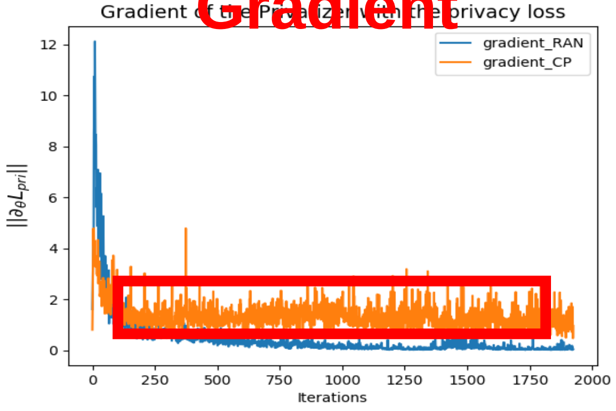



$$O_g = \sum_{i=1}^{m} |I_i - D(E(I_i))|^2$$

**Optimization:**

1. $\min_{E,C} \boldsymbol{O_d}$
2. $\min_{D} \boldsymbol{O_g}$
3. $\min_{E,C} \boldsymbol{\lambda O_g - (1-\lambda) O_d}$

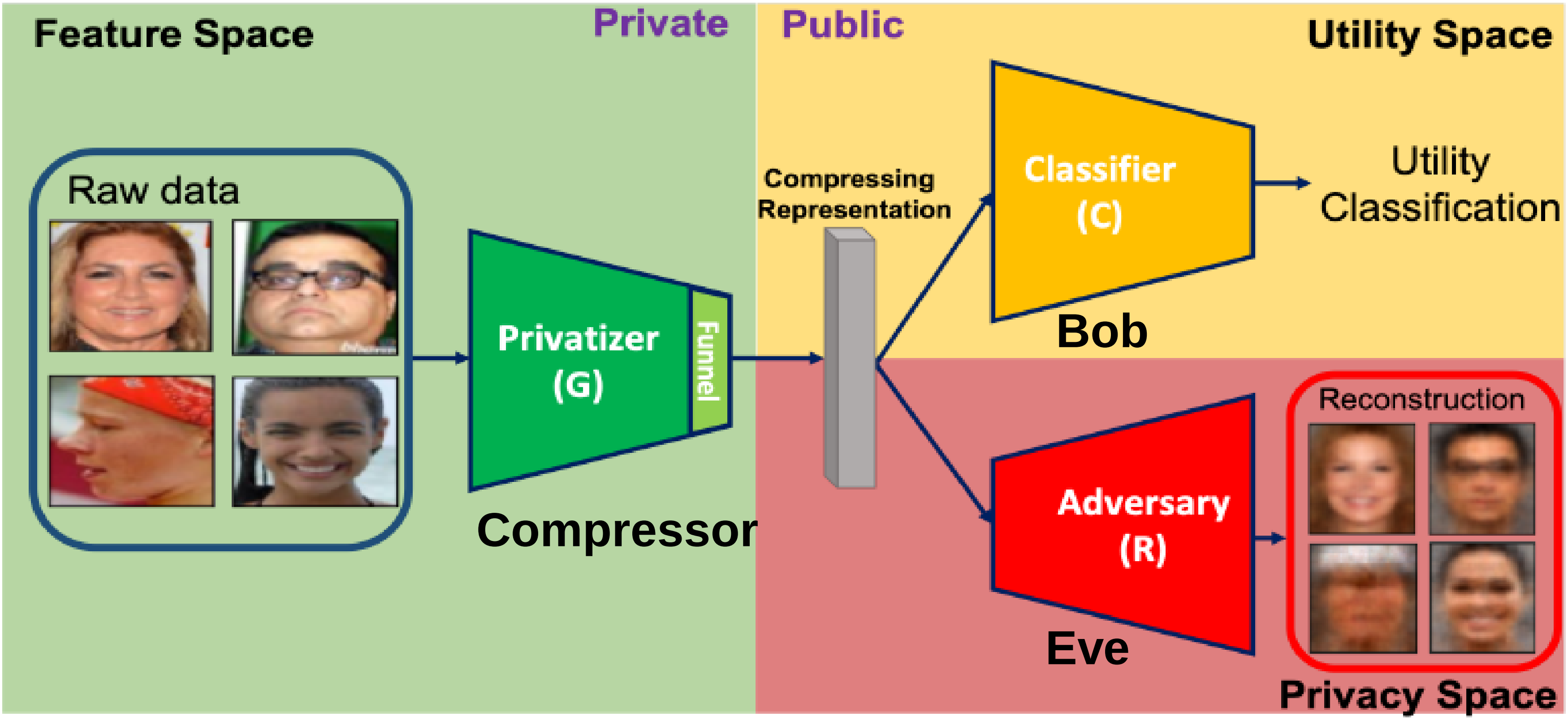
**Gradient**



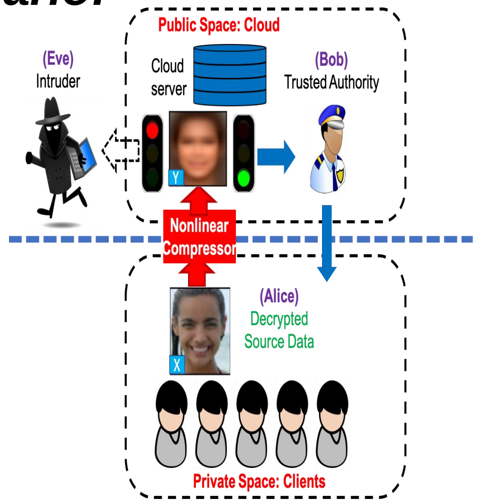

## Problems in RAN:

- **Gradient flowing to the encoder (E) is too weak.**
- **Without considering the dimension of the encoded vectors.**
- **Only apply neural network to the decoder (adversary).**

# CPGAN architecture




***Our scenario:***




Kung, S. Y. (2018). A Compressive Privacy approach to Generalized Information Bottleneck and Privacy Funnel problems. *Journal of the Franklin Institute*, *355*(4), 1846-1872.

CPGAN's scenario thus can be formulated as GAN's min-max function

$$\min_G \max_D [\log P(D(x) + \log P(1 - D(G(z))] \text{, where } z \sim \text{Gaussian and } x \sim P_x$$

# Formulate Proposed CPGAN

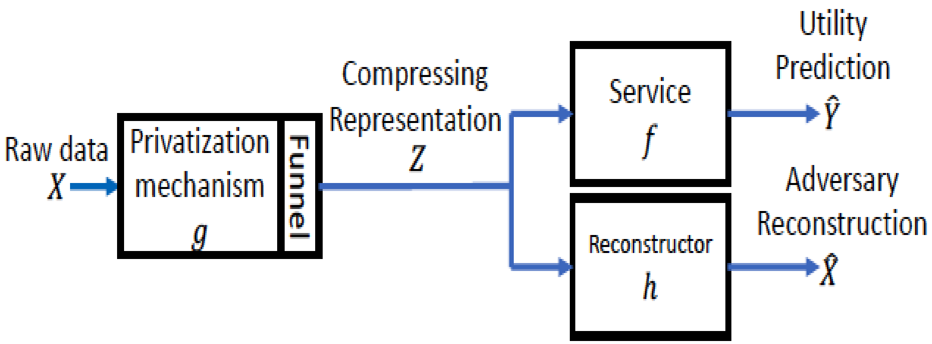
- We follow GAN's objective function to formulate CPGAN's architecture.
- Let $(X,Y)\sim P_{\mathcal{X},\mathcal{Y}}$, $Z|X\sim P_{g_\theta}(\cdot\,|X)$, $\hat{Y}|Z\sim P_{f_\tau}(\cdot\,|Z)$

$$L_{\text{util}}(\mathrm{P_f}(\cdot\,|Z),Y)=\mathbb{E}[-\log\mathrm{P_f}(\mathrm{Y}|\mathrm{Z})]$$

**It's equal to cross entropy**

$$L_{adv}=\mathbb{E}_{\hat{X}\sim P_h(\cdot|Z)}[\|X-\hat{X}\|_2^2]$$

**Mean square error**



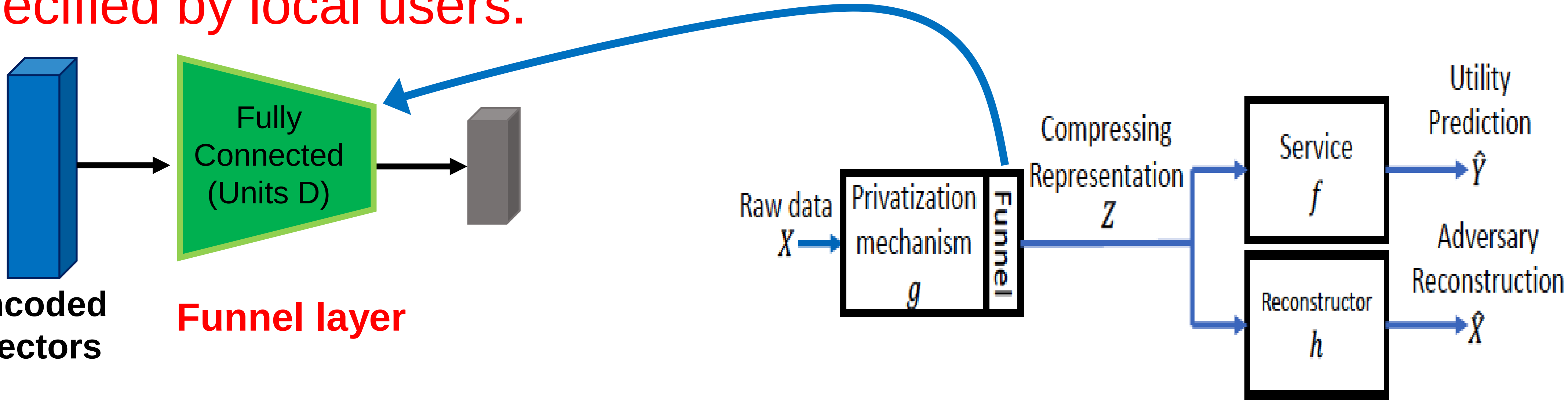

$$\max_g(\min_h L_{adv}(g,h)-\lambda\min_f L_{util}(g,f))$$

**Explore the best service ($f$) and reconstructor($h$)**

**The privatizer($g$) targets at attaining better accuracy ($f$) while fooling the reconstructor($h$)**

# Design of the Privatizer

- Funnel layer:
  - Compress the data into the dimension specified by local users.

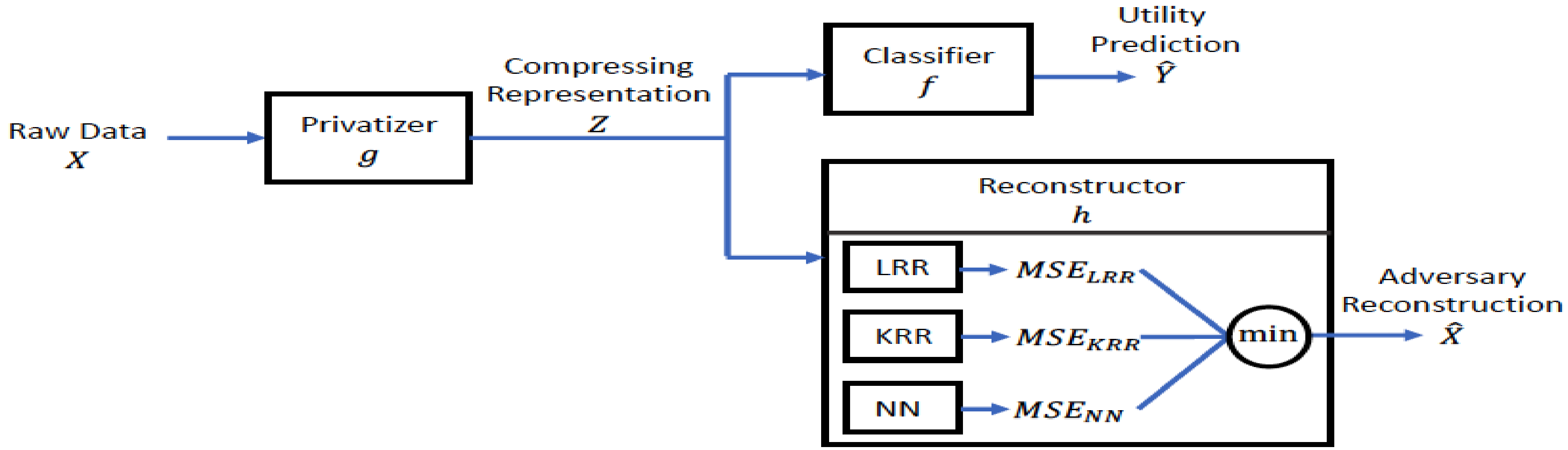



- Light-weight design:
  - It's thus applicable for the limiting computation resource, such as mobile device.

# Multiple adversaries scheme

- Why?
  - It is well known that the optimization of the nonlinear neural network is intractable, furthermore, it is questionable whether NN achieves the global optimum or saddle point.

***Architecture:***

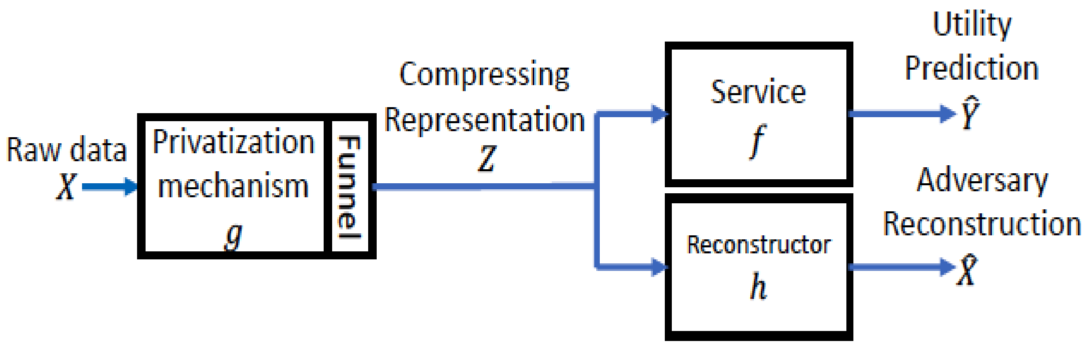



**LRR: Linear Ridge Regression, KRR: Kernel Ridge Regression**

# The close-form solution of LRR and KRR

**Assuming Z and X is center-adjusted, where $\rho$ is the regularization term.**

- For Linear Ridge Regression:




$$\underline{W_{LRR} = (ZZ^T + \rho I)^{-1} ZX}$$

- For Kernel Ridge Regression:

$$W_{LRR} = (\phi(Z)\phi(Z)^T + \rho I)^{-1}\phi(Z)X$$
$$= \phi\,(Z)(\phi(Z)^T\phi(Z) + \rho I)^{-1}X$$

$\boldsymbol{\phi: R^m \rightarrow R^n, where\ m < n}$

**Intrinsic space**

**Learning subspace property:**
$\boldsymbol{W = \phi(Z)A}$

$$\underline{A = W_{KRR} = (K + \rho I)^{-1}X}$$

**Empirical space**

***Reference:***

S. Y. Kung, *Kernel Methods and Machine Learning*. Cambridge University Press, 2014.

# Random Fourier Feature (RFF)

- Caused by the high computation cost on kernel matrix O($N^2$).
- RFF is inspired from Bochner's theorem:
  - The expectation of the inner product of two mapping points is the unbiased approximation of the shift-invariant kernel. (i.e. $k(x,y) = k(x-y)$)

$$k(\mathbf{x}-\mathbf{y}) = \int_{\mathcal{R}^d} p(\omega) e^{j\omega'(\mathbf{x}-\mathbf{y})}\, d\omega = E_\omega[\zeta_\omega(\mathbf{x})\zeta_\omega(\mathbf{y})^*],$$

- Some parameters used in LRR and KRR adversary:

Table I. Parameters of KRR on different dataset

| | Synthetic dataset | MNIST | HAR | GENKI-4K | SVHN | CIFAR-10 | CelebA |
|---|---|---|---|---|---|---|---|
| Ridge | 1 | 0.001 | 1 | 1 | 0.001 | 0.001 | 0.001 |
| Mapping dimension | 10000 | 500 | 5000 | 2048 | 5000 | 5000 | 2000 |
| Gamma | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |

***Reference:***

A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, (USA), pp. 1177–1184, Curran Associates Inc., 2007.

# Neural Network (NN)

## Our implementation detail of the adversary(NN), privatizer and classifier.

Table IV. Implementation detail of proposed CPGAN on SVHN and CIFAR-10 dataset

| | SVHN | | | | CIFAR-10 | | | |
|---|---|---|---|---|---|---|---|---|
| | Layers | Units | Optimizer | Learning rate | Layers | Units | Optimizer | Learning rate |
| Privatizer | 13-layer<br>Residual Network [59], [63] | | Adam | 0.001 | 13-layer<br>Residual Network | | Adam | 0.001 |
| Reconstructor | Conv-T[1] stride=1<br>Conv-T, stride=1<br>Conv-T, stride=1<br>Conv-T, stride=1 | 128<br>64<br>32<br>3 | Adam | 0.001 | Conv-T, stride=1<br>Conv-T, stride=1<br>Conv-T, stride=1<br>Conv-T, stride=1 | 128<br>64<br>32<br>3 | Adam | 0.001 |
| Classifier | 16-8 Wide<br>Residual Networks [57] | | Adam | 0.01 [2] | 26-2x32d<br>Shake-shake Regularization [54] | | Momentum [64] | 0.01 [2] |
| Epochs | 160 | | | | 1800 | | | |

[1] The notation "Conv-t" means the deconvolution layers (upsampling).
[2] We apply cosine learning rate decay [54].

Table V. Parameters and computation cost on SVHN and CIFAR-10 dataset.

| | SVHN | | | CIFAR-10 | | |
|---|---|---|---|---|---|---|
| | Parameters | Addition | Multiplication | Parameters | Addition | Multiplication |
| Privatizer | 1647 | 1575963 | 1575954 | 1647 | 1575963 | 1575954 |
| Classifier | 2923162 | 438281548 | 439272780 | 10961834 | 1547703315 | 1547703309 |

Table VI. Implementation detail of proposed CPGAN on CelebA dataset

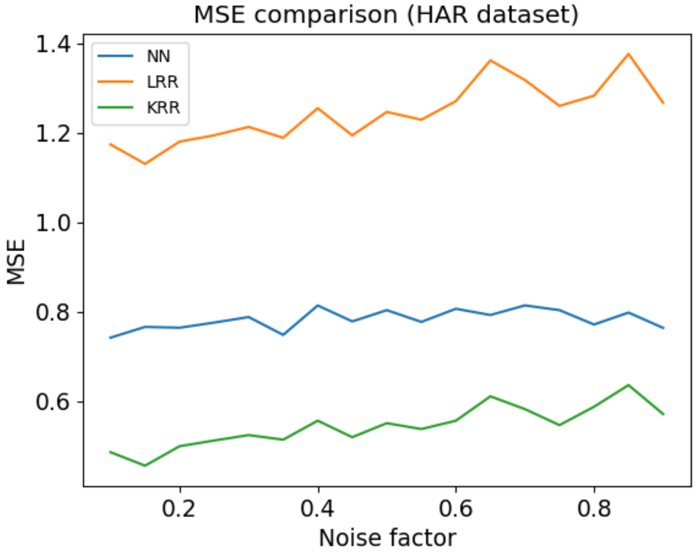
| | Single task CelebA | | | | | Multiple task CelebA | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Layers | Units | Optimizer | Learning rate | Parameter | Layers | Units | Optimizer | Learning rate | Parameter |
| Privatizer | Conv, stride=2<br>Conv, stride=2<br>Conv, stride=2<br>Conv, stride=2<br>Fully Conneted | 64<br>128<br>256<br>512<br>compressive-d[1] | Adam [65] | 0.001 | 414466 | From "conv_1" to "concat" in ATNET_GT [60]<br>Fully Connected with compressive-d units | | Adam | 0.001 | 673600 |
| Reconstructor | Fully Connected<br>Reshape<br>Conv-T[2] stride=2<br>Conv-T, stride=2<br>Conv-T, stride=2<br>Conv-T, stride=2 | 192<br><br>128<br>64<br>32<br>3 | Adam | 0.001 | | Fully Connected<br>Batch Norm [66]<br>Reshape<br>Conv-T, stride=2<br>Conv-T, stride=2<br>Conv-T, stride=2<br>Conv-T, stride=2<br>Conv-T, stride=2 | 5*5*128<br><br><br>128<br>128<br>64<br>32<br>3 | Adam | 0.001 | |
| Classifier | Fully Connected<br>Batch Norm<br>Fully Connected<br>Fully Connected | 256<br><br>256<br>1 | Adam | 0.001 | 68098 | Fully Connected<br>Batch Norm<br>Fully Connected<br>Fully Connected<br>(40 branches) | 64<br><br>64<br>1 | Adam | 0.001 | 30160 |
| Epochs | 30 | | | | | 30 | | | | |

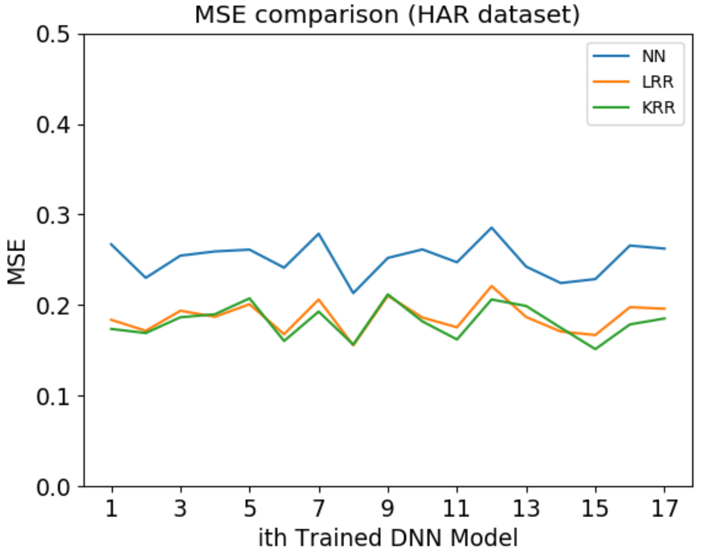
[1] The notation "compressive-d" means that the dimension of the compressing representations.
[2] The notation "Conv-t" means the deconvolution layers (upsampling).

Table VI. Implementation detail of proposed CPGAN on CelebA dataset

# Comparison Between the Adversaries

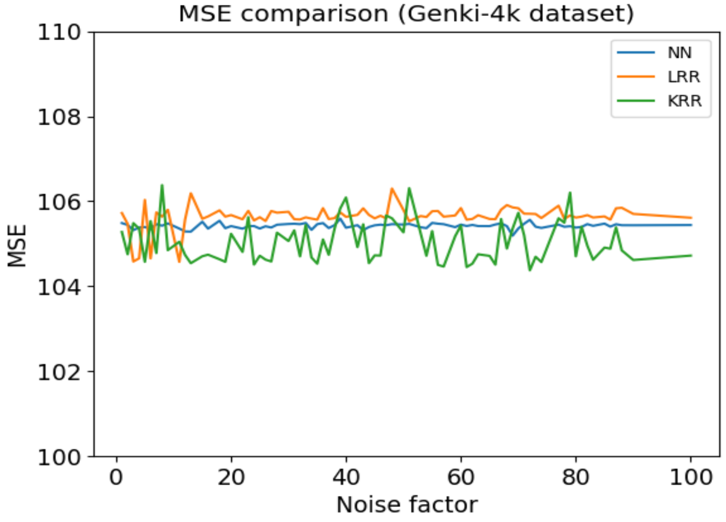
## HAR dataset:




**DNN (Resize) and DNN architecture**

## Genki-4K dataset:



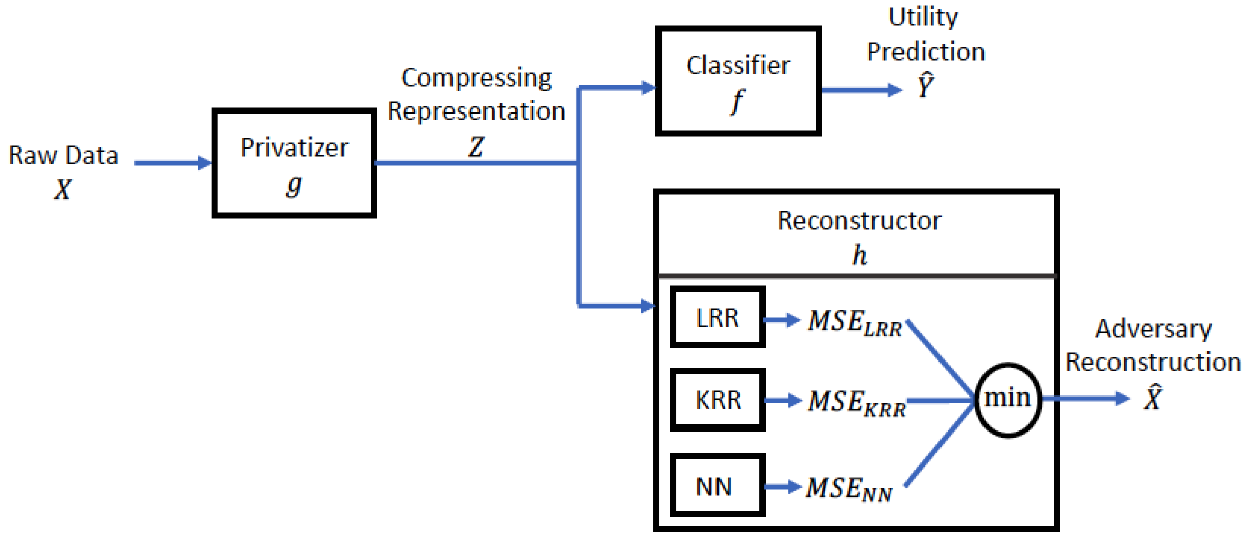

**CPGAN architecture**

**These figures indicate the neural network can not guarantee to achieve the best reconstruction error in the evaluation phase.**

# Algorithm

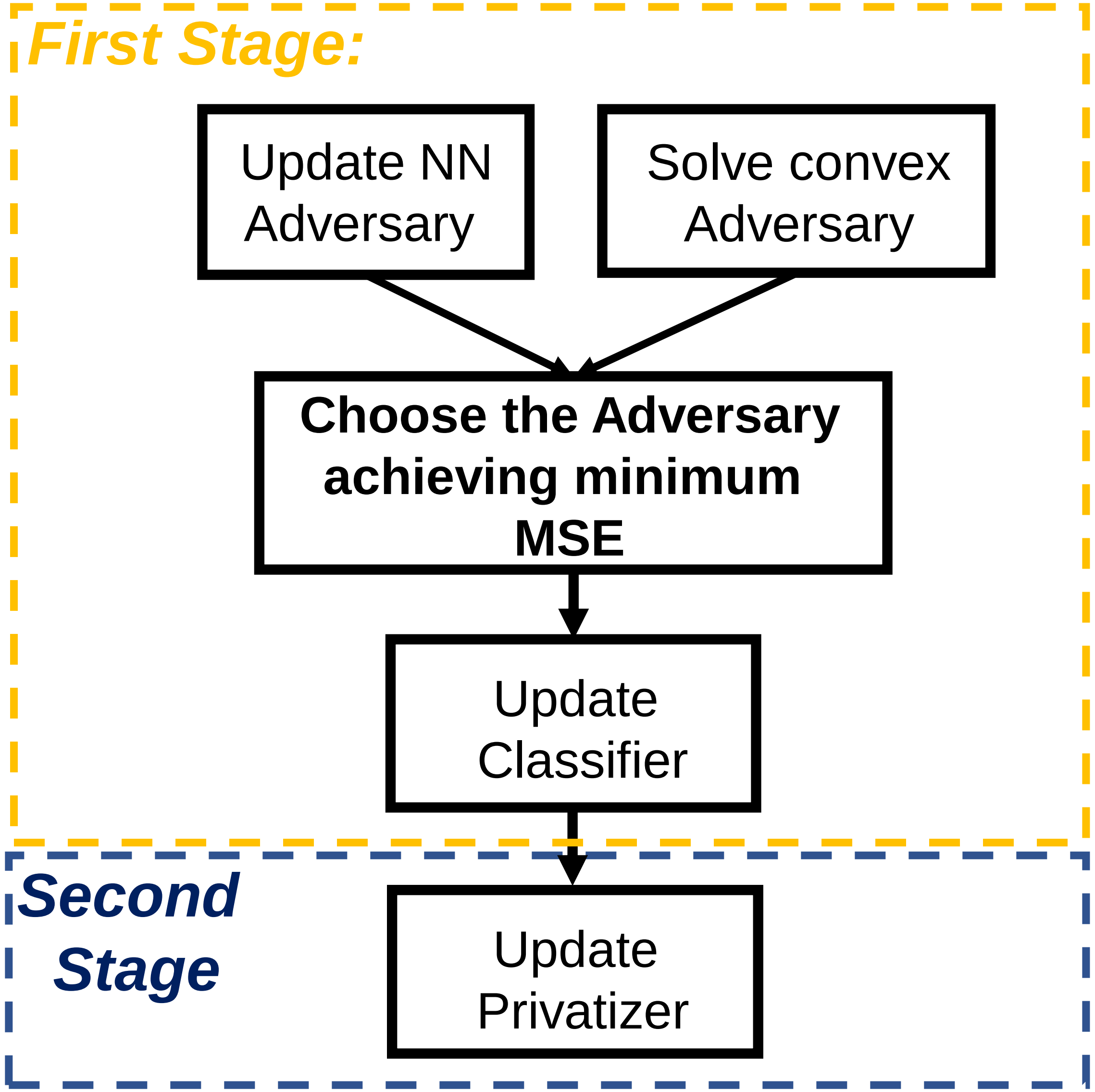
**We follow GAN's training strategy !!!**

## Architecture




## Objective function:

$$\max_g(\min_h L_{adv}(g,h) - \lambda \min_f L_{util}(g,f))$$

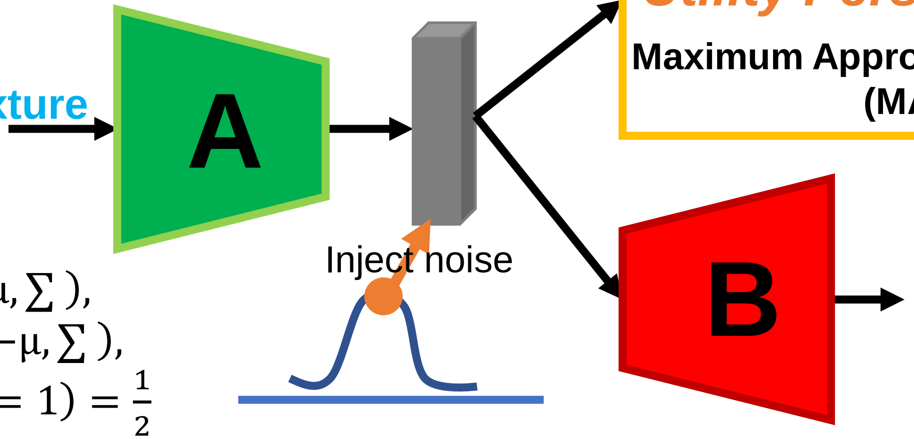


# Theoretical Analysis For CPGAN

***Reference:***

C. Huang, P. Kairouz, L. S. Xiao Chen, and R. Rajagopal, "Generative adversarial privacy," in *arXiv preprint arXiv:1807.05306*, 2018.

$X$~Gaussian mixture model




Assuming:

$P(X|Y=1) \sim \mathcal{N}(\mu, \Sigma)$,

$P(X|Y=0) \sim \mathcal{N}(-\mu, \Sigma)$,

$P(Y=0) = P(Y=1) = \frac{1}{2}$

***Utility Perspective:***

**Maximum Approximate Posterior (MAP)**

**Loss:**

$$P_G = qQ\left(\frac{-\alpha}{2} + \frac{1}{\alpha} ln\left(\frac{1-q}{q}\right)\right) + (1-q)Q\left(\frac{-\alpha}{2} - \frac{1}{\alpha} ln\left(\frac{1-q}{q}\right)\right) \quad (3)$$

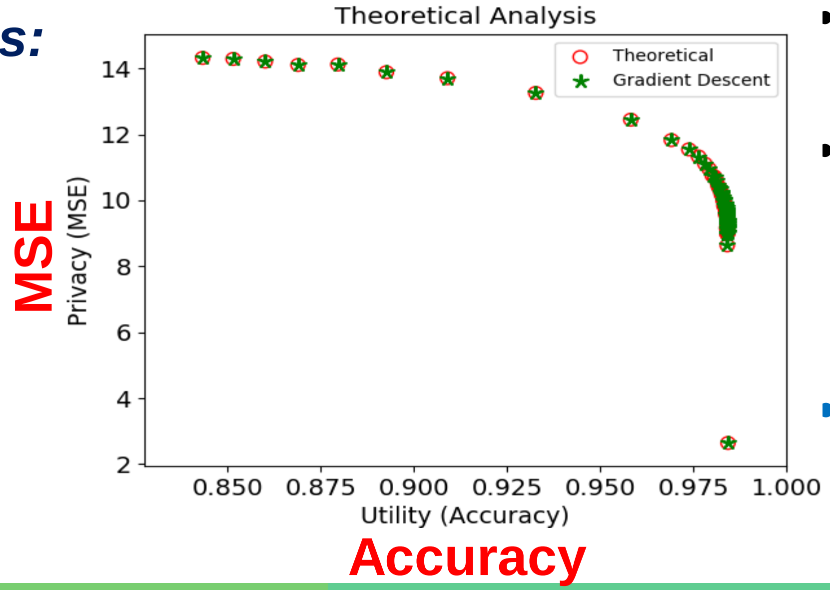
$$\text{where } \alpha = \sqrt{(2\mathbf{A}\vec{\mu})^T(\mathbf{A}\mathbf{R}_\xi\mathbf{A}^T + \mathbf{R}_\epsilon)^{-1}(2\mathbf{A}\vec{\mu})}$$

***Privacy Perspective (BLUE):***

**Pirvacy Loss conditioned on achieving the optimal $B$:**

$$Tr(R_x) - Tr(R_x A^T (A R_x A^T + \mathrm{R}_\epsilon)^{-1} A R_x)$$

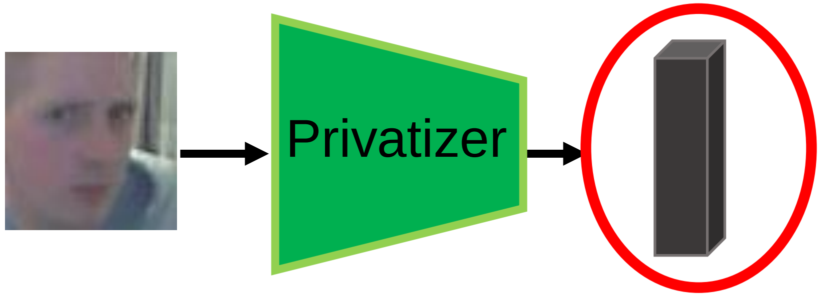
***Results:***




- It's intractable to optimize with the linear combination of privacy and utility loss (i.e. $\lambda L_{uti} - L_{pri}$)
- **Alternative way**:
  - Use gradient descent to determine the solution (**A**).
  - Substitute **CPGAN's privatizer with A.**
  - Train the Classifier and Reconstructor, respectively.
- **Conclusion:**
  - CPGAN achieves the trade-off approximate to the theoretical solution.

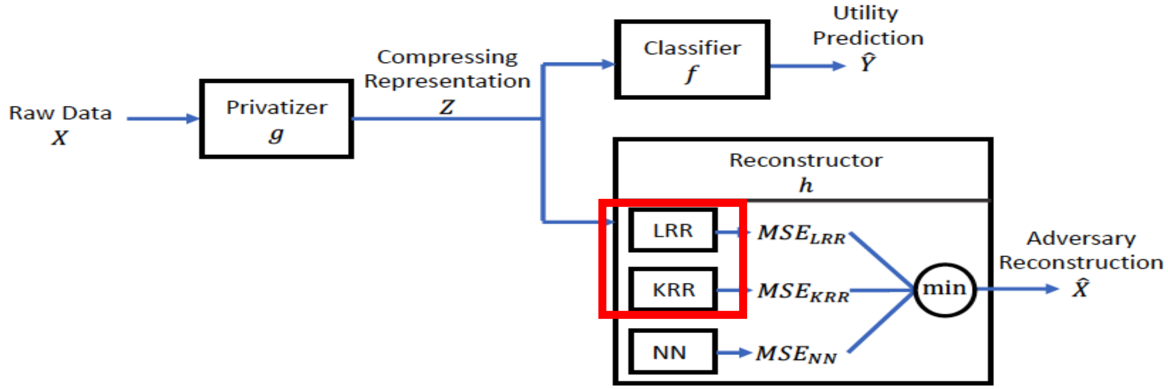
# Distinction between RAN and CPGAN

- Two tuning factors for the trade-off between privacy and utility.
  - $\max_{g}(\min_{h} L_{adv}(g,h) - \lambda \min_{f} L_{util}(g,f))$
  - Dimension of the compressing representation

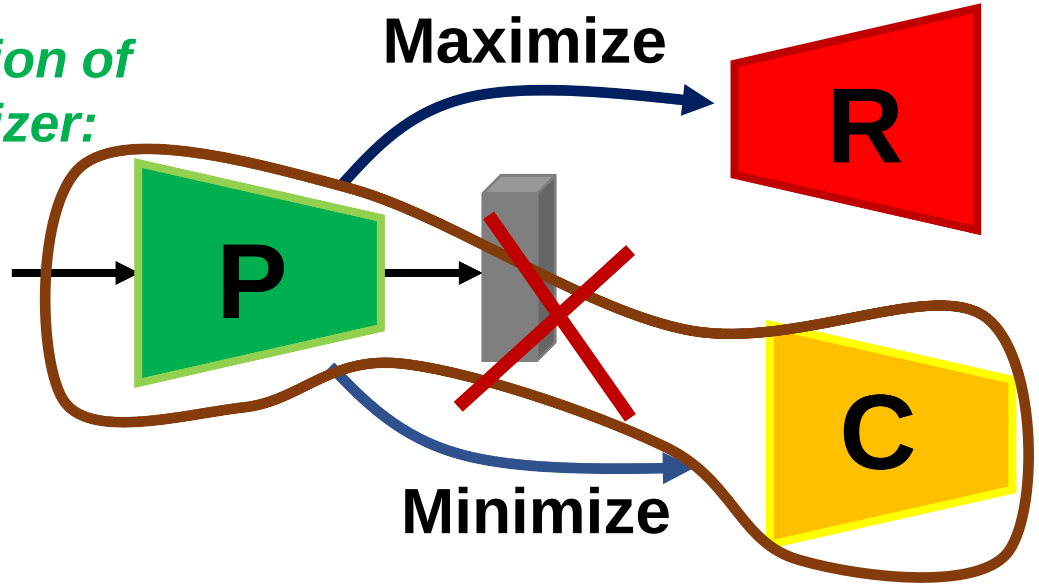



- Architecture and training strategy

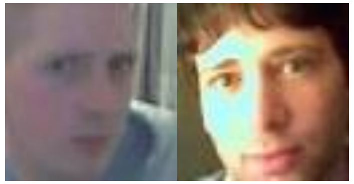
***Multiple adversaries Strategy in training/evaluation:***



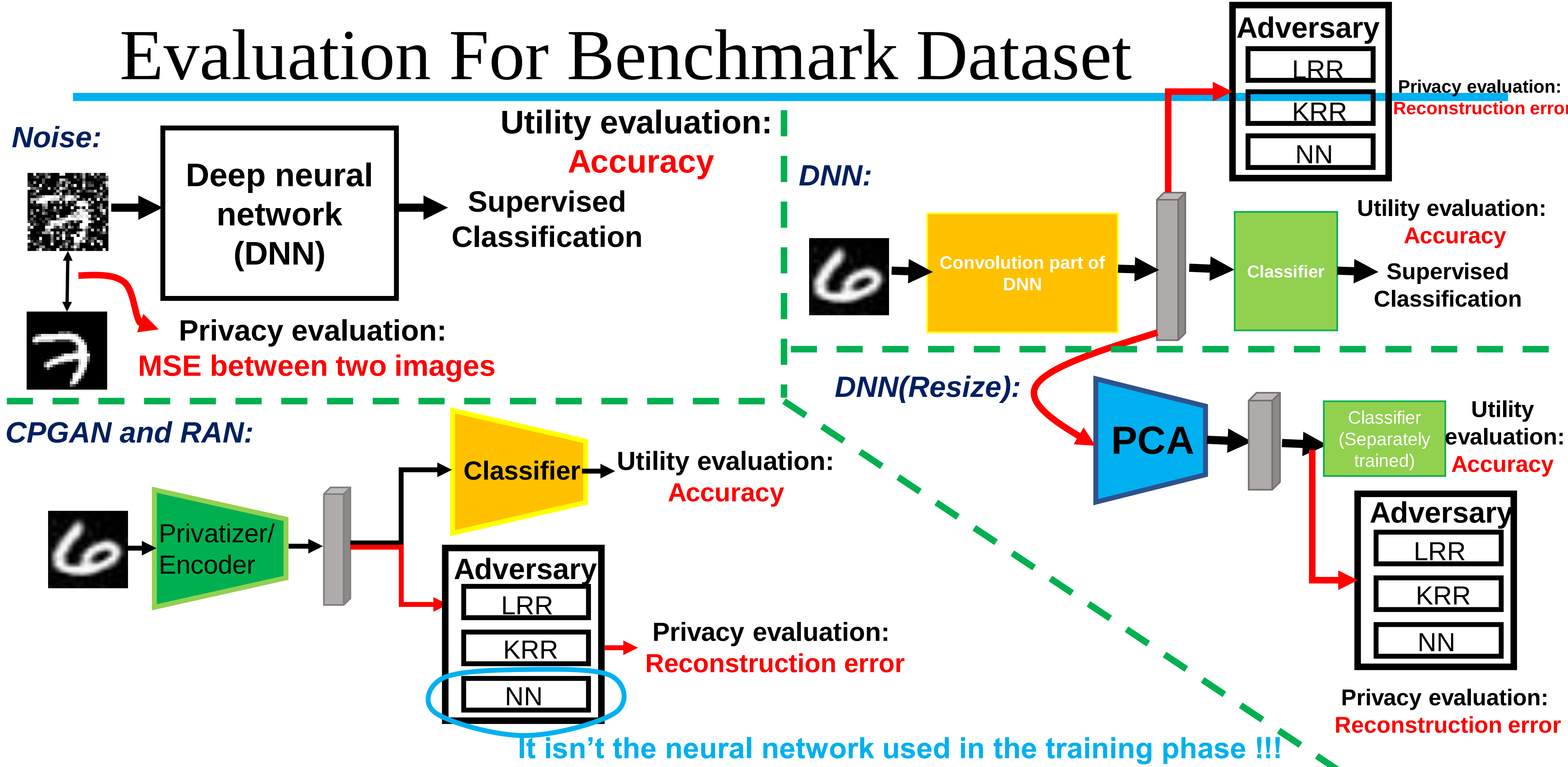

***Optimization of the Privatizer:***



# CPGAN for Benchmark Dataset

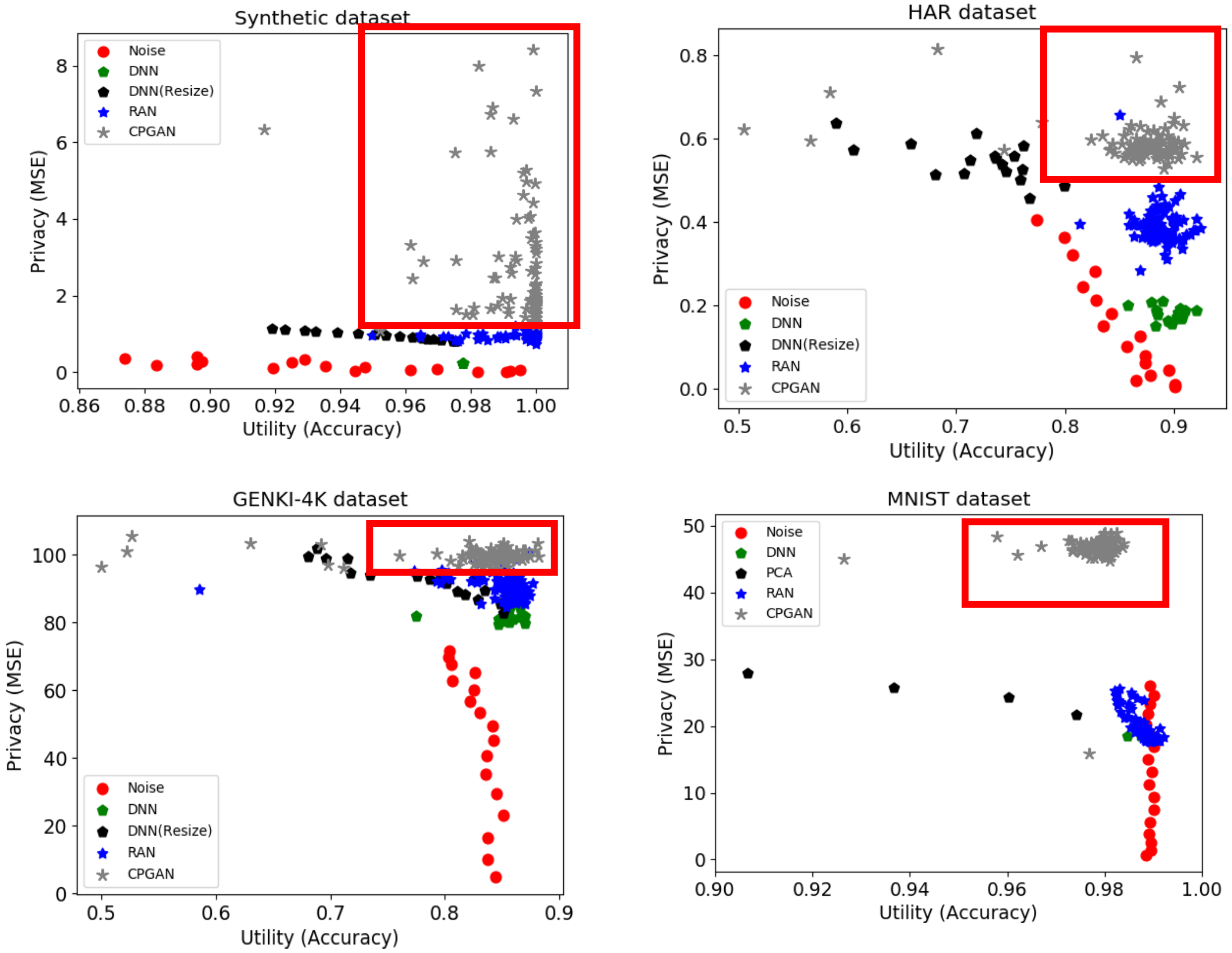
- Synthetic dataset:
  - Sampled from Gaussian mixture data model with binary class.
  - Training/testing samples: 20K/2K
- MNIST:
  - Training/testing samples: 55000/10000
  - Examples 
- UCI Human activity recognition (HAR) dataset
  - Given the time-series sensor record from ten identities.
  - Six activities: walking, sitting, standing etc.
- Genki-4K dataset:
  - Face images with 4000 sample. Detect the expression of this image.
  - Example: 

# Evaluation For Benchmark Dataset




*Under white-box attack: attacker has full access to the training data and representations.*

# Quantitative Analysis




**X-axis -> Utility Accuracy**
**Y-axis -> Privacy MSE**

***Privacy Perspective:***

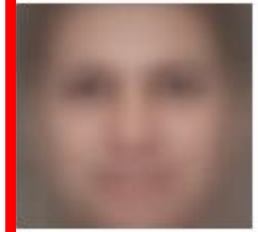
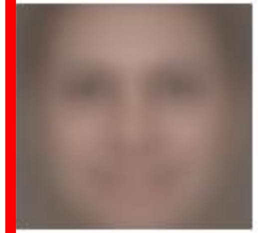
- Outperform than other methods

***Utility Perspective:***

- Drop by 1% on MNIST dataset.
- Get comparable accuracy on Synthetic, HAR and GenkI-4K dataset.

# Qualitative Analysis

Table II. Reconstructed images from five privacy preserving mechanisms on GENKI-4K dataset.

| | Original | Noise | DNN | DNN (Resize) | RAN | CPGAN |
|---|---|---|---|---|---|---|
| Average Accuracy (%) | | 69.83 | 85.19 | 77.70 | 84.89 | 84.93 |
| Image1 |  |  |  |  |  |  |
| Image2 |  |  |  |  |  |  |

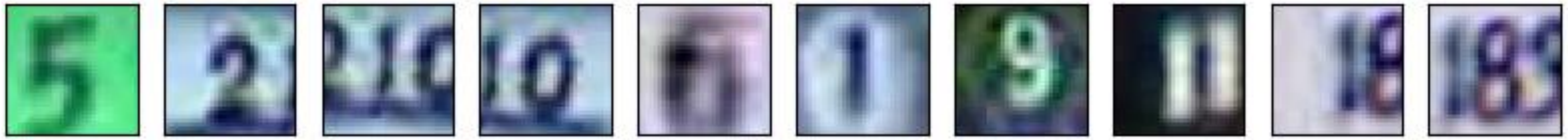
**Reconstructed images from our CPGAN is the most unrecognizable and the average accuracy only drops by 0.26%**

# CPGAN for Real Dataset

- CelebA
  - 202599 images (218*178*3), 10122 identities, each image has forty attributes.
  - Image is cropped to 175*175*3/112*112*3 for multi/single attribute classification
  - Example:

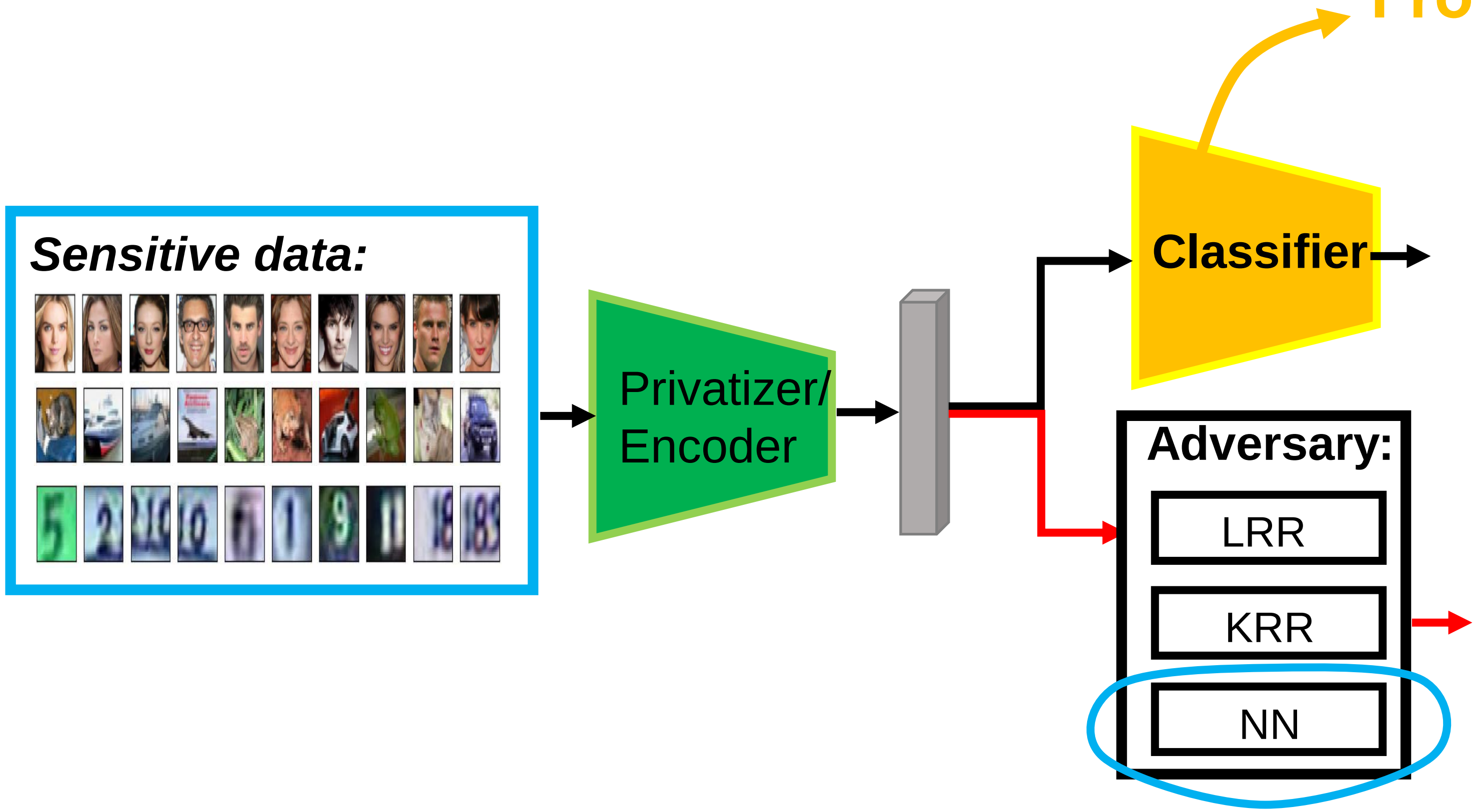
- CIFAR-10
  - There are 50000/10000 images for training/test, each image size is 32*32*3.
  - Ten classes (such as cat, airplane, .. etc.)
  - Example:

- SVHN
  - 604388/26032 images for training/testing, each image size is 32*32*3.
  - Ten classes (from 0 to 9)
  - Example:



# Evaluation For Real Dataset

From CPGAN’s Classifier !!!




**Utility evaluation:**
**Inference accuracy compared with the state-of-the-art model**

**Qualitative Privacy evaluation:**
**Reconstruction images**

**It isn’t the neural network used in the training phase !!!**

***Under white-box attack: attacker has full access to the training data and compressing representations.***

# Results on CIFAR-10 and SVHN

**Adversary reconstruction**

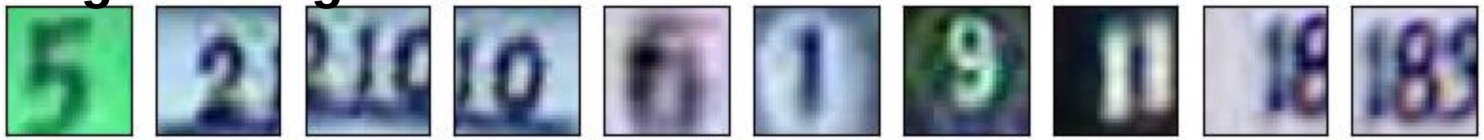
***CIFAR-10:***

**Original images**

**Reconstructed image by adversary**

***SVHN:***

**Original images**



**Reconstructed image by adversary**

***Utility accuracy:***

| | CIFAR-10 | SVHN |
|---|---|---|
| CPGAN | 93.87% | 97.68% |
| ResNet-20 [2] | 92.28% | 97.70% |
| Xavier [3] | 96.45% | 98.6% |
| Zagoruyko [4] | 95.83% | 98.3% |

**Classify by compressed data**

**Classify by original image**

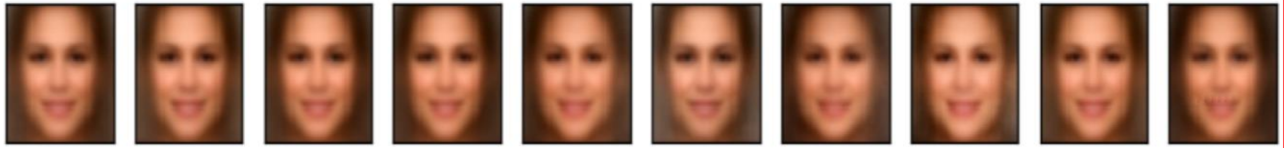
***CPGAN defends the reconstruction attack under white-box attack while achieving satisfactory utility performance***

# Results on CelebA

**CELEBA:** ***Single attribute classification***

**Original images**

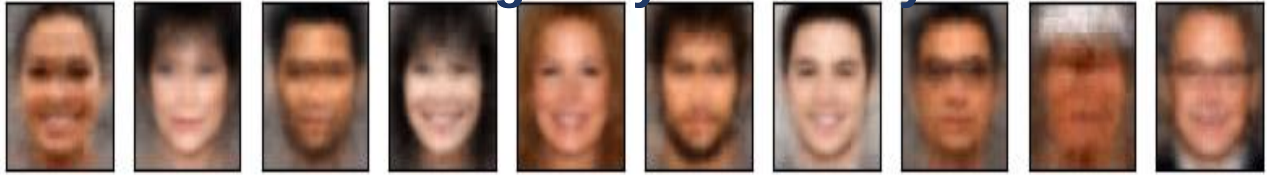
**Reconstructed images by adversary**



**CELEBA:** ***Multiple attribute classification***

**Original images**

**Reconstructed images by adversary**



***Utility accuracy:***

Table V. Average accuracy of Single attribute CPGAN

| | LNets+ANets [58] | Zhong [66] | CPGAN |
|---|---|---|---|
| Accuracy | 87.30% | 89.97% | 89.92% |

**Classified by original image** **Classified by Compressed data**

Table VI. Average accuracy of multiple attribute CPGAN

| | Han [64] | ATNET_GT [63] | CPGAN |
|---|---|---|---|
| Accuracy | 92.52% | 90.18% | 90.30% |

**Classified by original image** **Classified by Compressed data**

***Accuracy of 40 attributes:***

| | 5 o Clock Shadow | Arched Eyebrows | Attractive | Bags Under Eyes | Bald | Bangs | Big Lips | Big Nose | Black Hair | Blond Hair | Blurry | Brown Hair | Bushy Eyebrows | Chubby | Double Chin | Eyeglasses | Goatee | Gray Hair | Heavy Makeup | High Cheekbones |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LNets+ANets [39] | 91 | 79 | 81 | 79 | 98 | 95 | 68 | 78 | 88 | 95 | 84 | 80 | 90 | 91 | 92 | 99 | 95 | 97 | 90 | 87 |
| Zhong [40] | 93 | 83 | 81 | 82 | 98 | 96 | 70 | 83 | 86 | 95 | 96 | 84 | 92 | 95 | 96 | 100 | 97 | 98 | 90 | 86 |
| Hu [42] | 95 | 86 | 83 | 85 | 99 | 99 | 96 | 85 | 91 | 96 | 96 | 88 | 92 | 96 | 97 | 99 | 99 | 98 | 92 | 88 |
| ATNET_GT [41] | 92 | 81 | 81 | 84 | 99 | 96 | 71 | 83 | 89 | 95 | 96 | 87 | 92 | 94 | 96 | 99 | 97 | 98 | 90 | 86 |
| Single CPGAN | 92 | 82 | 80 | 83 | 98 | 95 | 71 | 83 | 89 | 95 | 95 | 85 | 90 | 95 | 96 | 99 | 96 | 98 | 90 | 85 |
| Multi CPGAN | 93 | 82 | 82 | 84 | 98 | 95 | 71 | 83 | 88 | 96 | 96 | 88 | 92 | 95 | 96 | 99 | 97 | 98 | 91 | 86 |

| | Male | Mouth S. Open | Mustache | Narrow Eyes | No Beard | Oval Face | Pale Skin | Pointy Nose | Receding Hairline | Rosy Cheeks | Sideburns | Smiling | Straight Hair | Wavy Hair | Wearing Earrings | Wearing Hat | Wearing Lipstick | Wearing Necklace | Wearing Necktie | Young |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LNets+ANets [39] | 98 | 92 | 95 | 81 | 95 | 66 | 91 | 72 | 89 | 90 | 96 | 92 | 73 | 80 | 82 | 99 | 93 | 71 | 93 | 87 |
| Zhong [40] | 98 | 93 | 97 | 87 | 95 | 71 | 97 | 76 | 92 | 94 | 97 | 92 | 80 | 77 | 87 | 99 | 92 | 86 | 94 | 88 |
| Hu [42] | 98 | 94 | 97 | 90 | 96 | 78 | 97 | 78 | 94 | 96 | 98 | 94 | 85 | 87 | 91 | 99 | 93 | 89 | 97 | 90 |
| ATNET_GT [41] | 97 | 93 | 97 | 86 | 94 | 76 | 97 | 75 | 93 | 95 | 97 | 92 | 80 | 82 | 89 | 99 | 93 | 86 | 96 | 88 |
| Single CPGAN | 100 | 93 | 97 | 89 | 91 | 72 | 96 | 75 | 94 | 95 | 96 | 92 | 79 | 78 | 88 | 99 | 92 | 83 | 94 | 87 |
| Multi CPGAN | 96 | 93 | 96 | 87 | 96 | 74 | 97 | 76 | 93 | 95 | 97 | 91 | 82 | 81 | 88 | 99 | 93 | 86 | 96 | 87 |

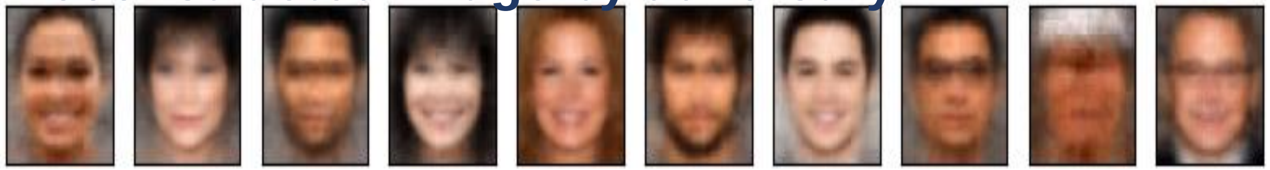
# Privacy leakage

**Reference:** D. Gao, P. Yuan, N. Sun, X. Wu, and Y. Cai, “Face attribute prediction with convolutional neural networks,” in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1294–1299, Dec 2017.

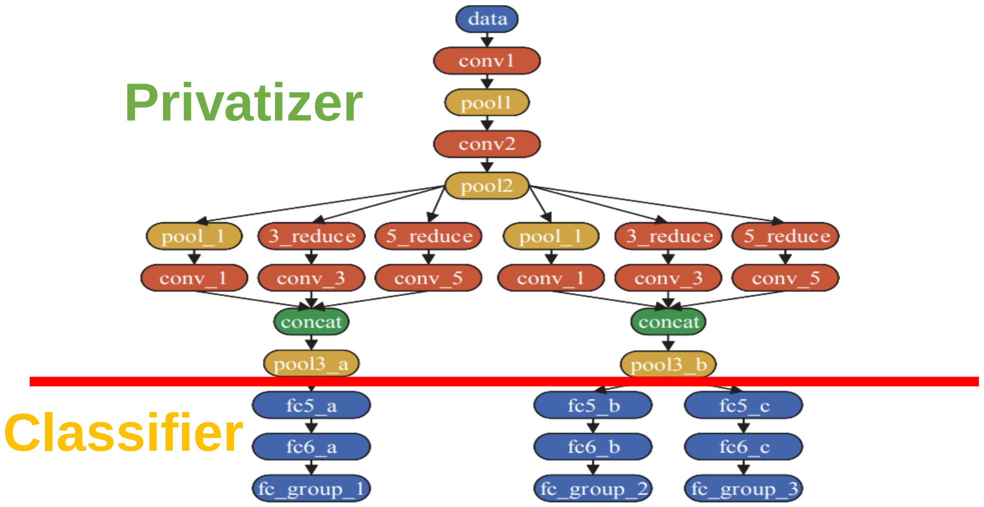
**CELEBA:** **Multiple attribute classification**

**Original images**

**Reconstructed image by adversary**



**Model architecture:**




| Group | Attribute |
|---|---|
| group 1 | black hair, blond hair, blurry, eyeglasses, gray hair, pale skin, straight hair, wearing hat |
| group 2 | attractive, bangs, brown hair, heavy makeup, high cheekbones, mouth slightly open, no beard, oval face, pointy nose, rosy cheeks, smiling, wavy hair, wearing lipstick, young |
| group 3 | 5 o'clock shadow, arched eyebrows, bags under eyes, bald, big lips, big nose, bushy eyebrows, chubby, double chin, goatee, male, mustache, narrow eyes, receding hairline, sideburns, wearing earrings, wearing necklace, wearing necktie |

- Privacy issue:
  - Adversaries are capable of attaining the information corresponding to 40 attributes.
- How to solve?
  - Tune the dimension of the compressing representations.

# Enhance CPGAN

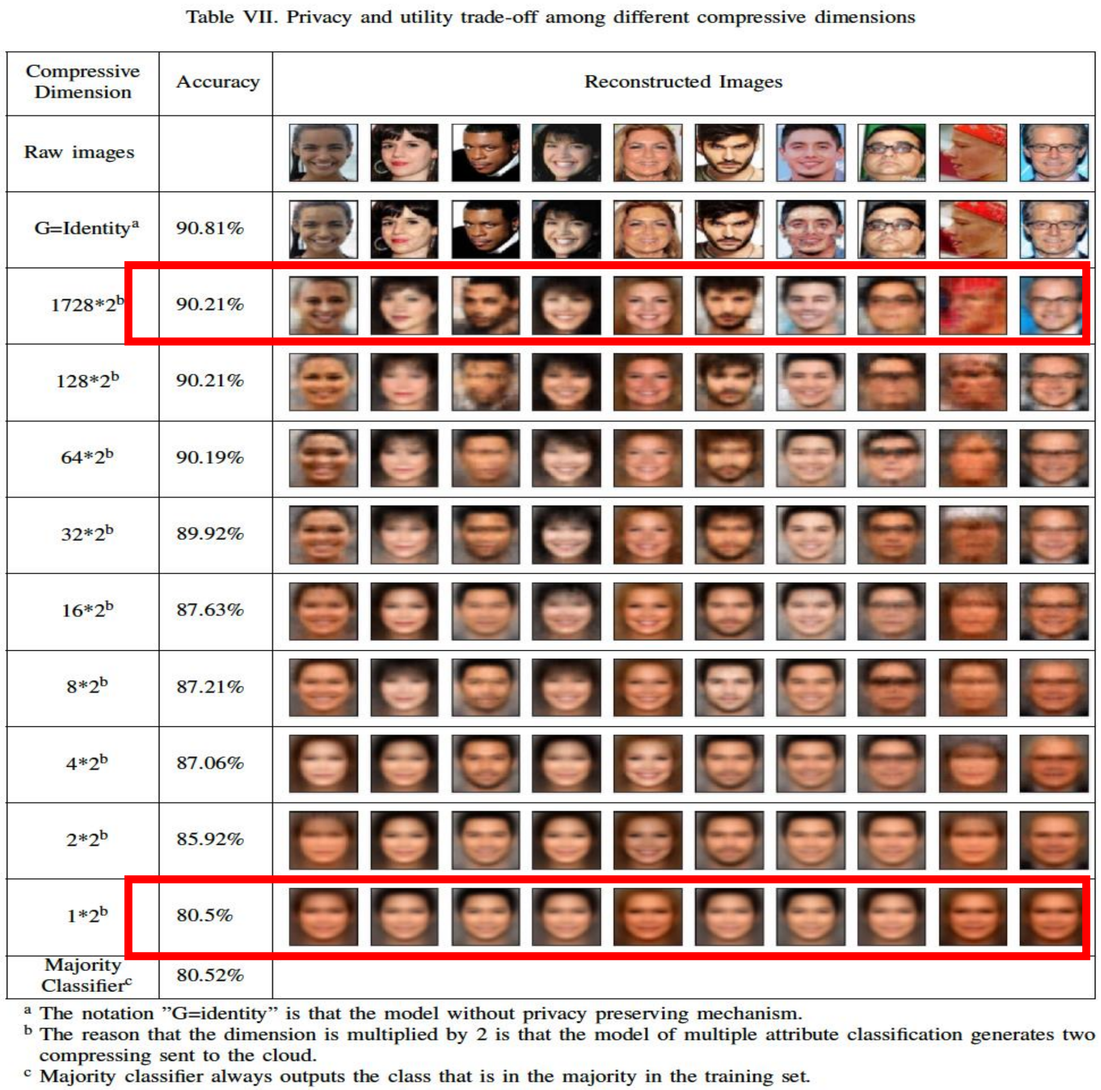


Table VII. Privacy and utility trade-off among different compressive dimensions

| Compressive Dimension | Accuracy | Reconstructed Images |
|---|---|---|
| Raw images | | |
| G=Identity[a] | 90.81% | |
| 1728*2[b] | 90.21% | |
| 128*2[b] | 90.21% | |
| 64*2[b] | 90.19% | |
| 32*2[b] | 89.92% | |
| 16*2[b] | 87.63% | |
| 8*2[b] | 87.21% | |
| 4*2[b] | 87.06% | |
| 2*2[b] | 85.92% | |
| 1*2[b] | 80.5% | |
| Majority Classifier[c] | 80.52% | |

[a] The notation "G=identity" is that the model without privacy preserving mechanism.
[b] The reason that the dimension is multiplied by 2 is that the model of multiple attribute classification generates two compressing sent to the cloud.
[c] Majority classifier always outputs the class that is in the majority in the training set.

***Accuracy***

**90.21%**

**80.5%**

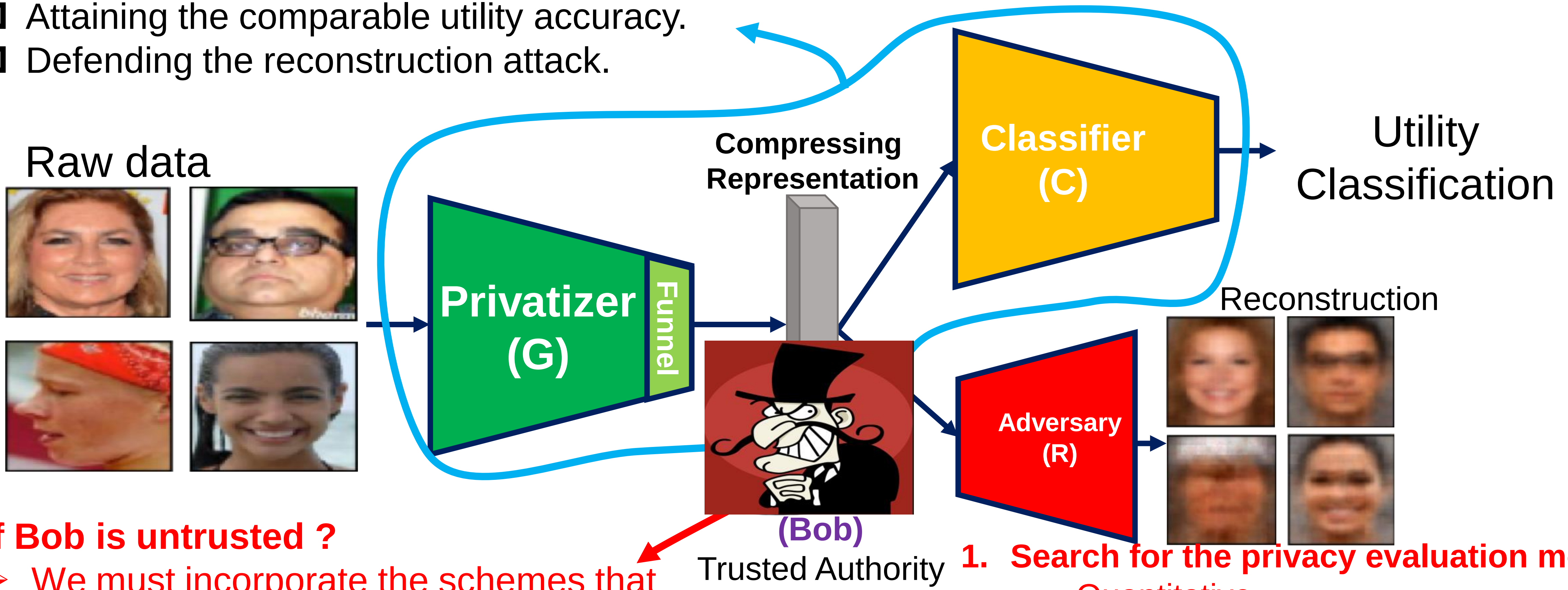
***Reconstruction:***

**Images remaining sensitive information**

**Unrecognizable Images**

# Conclusions and Future Works

- We develop the local compression network that prevents sensitive data from getting exposed to public.
- We confirms that the compressing representation is capable of
  - Attaining the comparable utility accuracy.
  - Defending the reconstruction attack.




**2. If Bob is untrusted ?**

- ➢ We must incorporate the schemes that can protect the training data to CPGAN.

**1. Search for the privacy evaluation metric**

- Quantitative
- Human's vision perception.

# Reference

[1]Sicong Liu et al., “Better accuracy with quantified privacy: representations learned via reconstructive adversarial network,”arXiv, 2017.
[2]Kaiming He et al., “Identity Mappings in Deep Residual Networks,” ECCV, 2016.
[3]Xavier Gastaldi, “Shake-Shake regularization,” arXiv, 2017.
[4]Sergey Zagoruyko et al., “Wide Residual Networks,”arXiv, 2017.
[5]Doudou Gao et al., “Face attribute Prediction with Convolutional Neural Networks,” IEEE conference, 2018.
[6]Hu Han et al., “Heterogeneous Face Attribute Estimation: A Deep Multi-Task Learning Approach,” IEEE, 2018.
[7]Chong Huang et al., “Generative Adversarial Privacy,” ICML workshop, 2018

# Compressive Privacy Generative Adversarial Networks

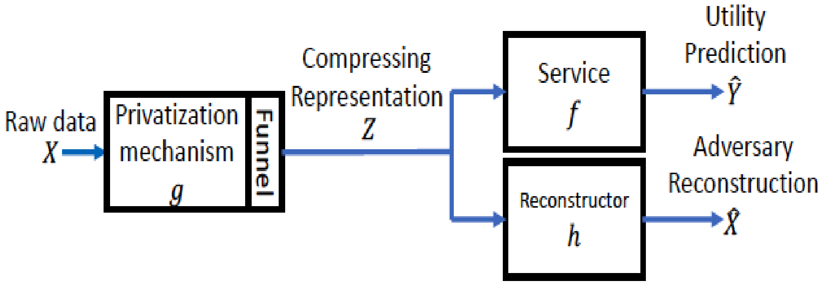
*Bo-Wei Tseng, Pei-Yuan Wu*

***Why?*** **-> Solve the privacy issue (reconstruction attack) occurring in the MLaaS model.**

***What?*** **-> Develop the local privacy preserving mechanism (privatizer) to prevent the sensitive data from getting exposed to the cloud.**

***How?*** **-> Incorporate the multiple adversaries strategy to adversarial learning scheme.**

## *CPGAN architecture:*




## *CPGAN objective function:*

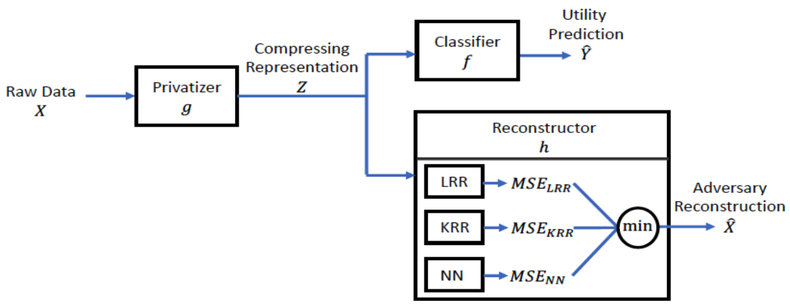
$$\max_{g}(\min_{h} L_{adv}(g,h) - \lambda \min_{f} L_{util}(g,f))$$

$$L_{adv} = \mathbb{E}_{\hat{X}\sim P_h(\cdot|Z)}[\|X-\hat{X}\|_2^2]$$

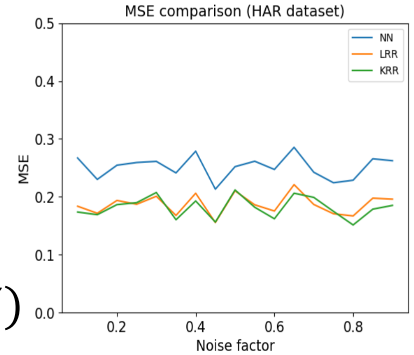
$$L_{\text{util}}(\text{P}_{\text{f}}(\cdot\,|Z),Y) = \mathbb{E}[-\log\text{P}_{\text{f}}(\text{Y}|\text{Z})]$$

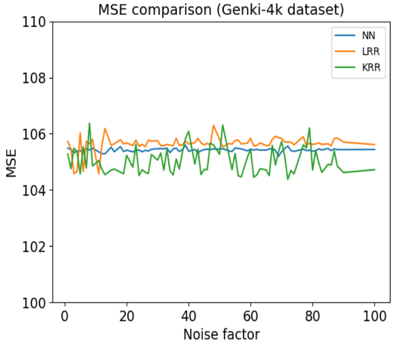
$$(X,Y)\sim P_{\mathcal{X},\mathcal{Y}},\ Z|X\sim P_{g_\theta}(\cdot\,|X),\ \hat{Y}|Z\sim P_{f_\tau}(\cdot\,|Z)$$

## *Multiple adversaries strategy for training/evaluation:*




## *Verification:*




## *Results on GENKI-4K dataset:*

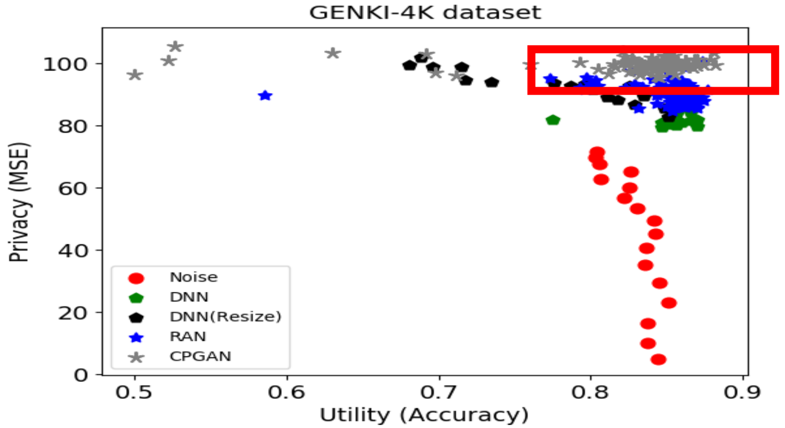
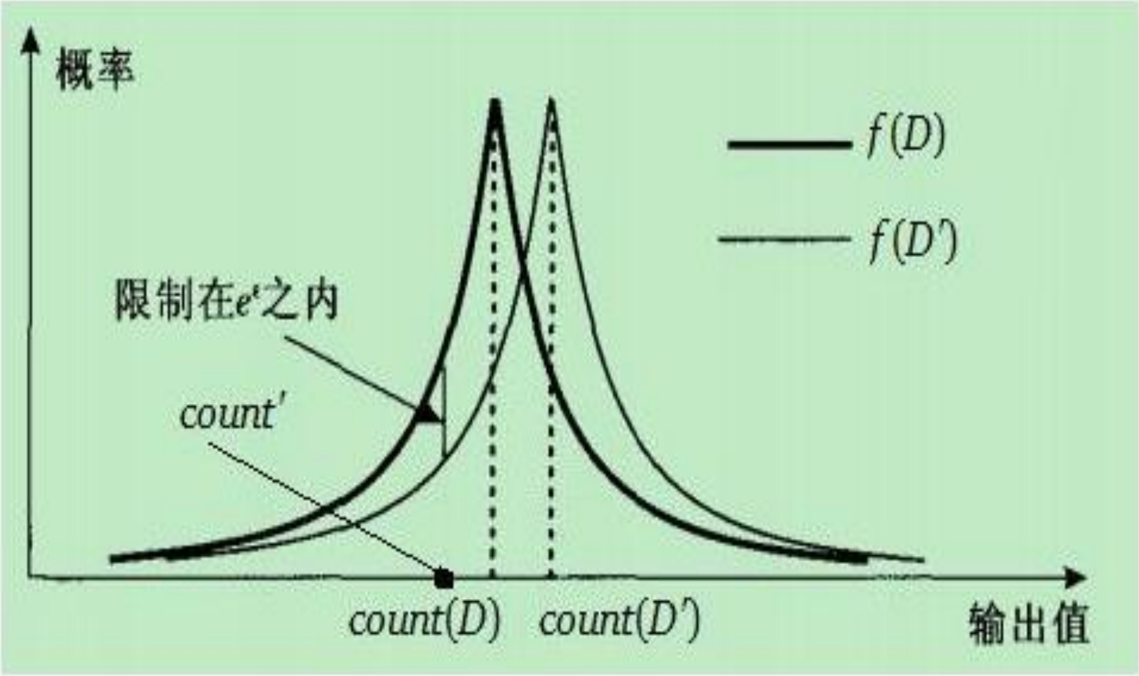



Table II. Reconstructed images from five privacy preserving mechanisms on GENKI-4K dataset.

| | Original | Noise | DNN | DNN (Resize) | RAN | CPGAN |
|---|---|---|---|---|---|---|
| Average Accuracy (%) | | 69.83 | 85.19 | 77.70 | 84.89 | 84.93 |
| Image1 | | | | | | |
| Image2 | | | | | | |

# Thank You !!!

# Differential Privacy




$$\Pr[\mathcal{A}(D_1) \in S] \leq e^{\epsilon} \times \Pr[\mathcal{A}(D_2) \in S],$$

## Differential privacy in Deep Learning:

The general steps for adding differential privacy to any learning algorithm are as follows:

1. Initialize learning parameters randomly.
2. Take a random sample.
3. Compute gradient on that random sample.
4. Clip the gradient.
5. Add noise.
6. Descent.
7. Compute the overall privacy cost using a privacy accountant.

# Gradient reconstruction

- Take one neural network for example (i.e. regression problem), then the objective function is:
- Its derivations:

$$J(W, b, x, y) \stackrel{\text{def}}{=} (h_{W,b}(x) - y)^2$$

$$\eta_k \stackrel{\text{def}}{=} \frac{\delta J(W, b, x, y)}{\delta W_k} = 2(h_{W,b}(x) - y)\frac{\delta h_{W,b}(x)}{\delta W_k} = 2(h_{W,b}(x) - y)\frac{\delta f(\sum_{i=1}^{d} W_i x_i + b)}{\delta W_k}$$

$$= 2(h_{W,b}(x) - y) f'(\sum_{i=1}^{d} W_i x_i + b) \cdot x_k$$

$$\eta \stackrel{\text{def}}{=} \frac{\delta J(W, b, x, y)}{\delta b} = 2(h_{W,b}(x) - y)\frac{\delta h_{W,b}(x)}{\delta b} = 2(h_{W,b}(x) - y)\frac{\delta f(\sum_{i=1}^{d} W_i x_i + b)}{\delta b}$$

$$= 2(h_{W,b}(x) - y) f'(\sum_{i=1}^{d} W_i x_i + b) \cdot 1.$$

**Thus,** $\eta_k / \eta = x_k.$

# DCA Formulation

**From derivation maximum utility mutual information**

**Derivative:**

$$
\begin{aligned}
I(u;y) &= H(u) - H(u|y) \\
&= \frac{1}{2}\log_2|\Sigma_u| + \frac{\mu}{2}\log_2 2\pi e - \frac{1}{2}\log_2|\Sigma_{\hat{u}}| + \frac{\mu}{2}\log_2 2\pi e \\
&= \frac{1}{2}\log_2|\Sigma_u| - |\Sigma_{\hat{u}}| \\
&= \frac{-1}{2}\log_2|\Sigma_{\hat{u}}| - |\Sigma_u| \\
&= \frac{-1}{2}\log_2(|\Sigma_u + (\Sigma_{\hat{u}} - \Sigma_u)| - |\Sigma_u|) \\
&\cong \frac{-1}{2}Tr(\Sigma_u^{-1}(\Sigma_{\hat{u}} - \Sigma_u)) \\
&= \frac{1}{2}Tr(\Sigma_u^{-1}(U^T\Sigma_x U - U^T\Sigma_{\hat{x}}U)) \\
&= \frac{1}{2}Tr(\Sigma_u^{-1}(U^T(\Sigma_x - \Sigma_{\hat{x}})U)) \\
&= \frac{1}{2}Tr(\Sigma_u^{-1}(U^T(\Sigma_x U^T(F^T(\Sigma_x + \Sigma_\epsilon)F)^{-1}F^T)\Sigma_x U) \\
&= \frac{1}{2}Tr((F^T(\Sigma_x + \Sigma_\epsilon)F)^{-1}F^T\Sigma_x U\Sigma_u^{-1}U^T\Sigma_x F)) \\
&= \frac{1}{2}Tr((F^T(\Sigma_x + \Sigma_\epsilon)F)^{-1}F^T\Omega F))
\end{aligned}
\qquad (8)
$$

# RFF theory

**Theorem 1** (Bochner [13]). *A continuous kernel $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y})$ on $\mathcal{R}^d$ is positive definite if and only if $k(\delta)$ is the Fourier transform of a non-negative measure.*

If a shift-invariant kernel $k(\delta)$ is properly scaled, Bochner's theorem guarantees that its Fourier transform $p(\omega)$ is a proper probability distribution. Defining $\zeta_\omega(\mathbf{x}) = e^{j\omega'\mathbf{x}}$, we have

$$k(\mathbf{x} - \mathbf{y}) = \int_{\mathcal{R}^d} p(\omega) e^{j\omega'(\mathbf{x}-\mathbf{y})}\, d\omega = E_\omega[\zeta_\omega(\mathbf{x})\zeta_\omega(\mathbf{y})^*], \tag{2}$$

so $\zeta_\omega(\mathbf{x})\zeta_\omega(\mathbf{y})^*$ is an unbiased estimate of $k(\mathbf{x}, \mathbf{y})$ when $\omega$ is drawn from $p$.

# MAP (In detail.)

## 1 Assumption

Let $\mathbf{x} = \boldsymbol{\tau} + \boldsymbol{\xi}$, where $\boldsymbol{\tau} \in \{-\boldsymbol{\mu}, \boldsymbol{\mu}\}$, $\boldsymbol{z} = \mathbf{A}\boldsymbol{x} + \boldsymbol{\epsilon}$, $\hat{\boldsymbol{x}} = \mathbf{B}\boldsymbol{z}$, where $\boldsymbol{\tau}, \boldsymbol{\xi}, \boldsymbol{\epsilon}$ are independent r.v.s with zero mean, and $\boldsymbol{\xi}, \boldsymbol{\epsilon}$ is sampled from Gaussian distribution. Thus, $\mathbf{R}_\tau = \mathbb{E}[\boldsymbol{\tau}\boldsymbol{\tau}^T]$, $\mathbf{R}_\xi = \mathbb{E}[\boldsymbol{\xi}\boldsymbol{\xi}^T]$. Note that if s=0 then $\boldsymbol{\tau} = \boldsymbol{\mu}$, s=0 is on the contrary, where $s$ denotes the utility label (binary):

- Since the diagonal covariance matrix can make the analysis simpler, and shift the mean vector to the form (such as $[\alpha, 0, 0, 0, 0, \ldots]$). Note that we assume the mean vector is $(\boldsymbol{\mu}, -\boldsymbol{\mu})$ in the following discussion.

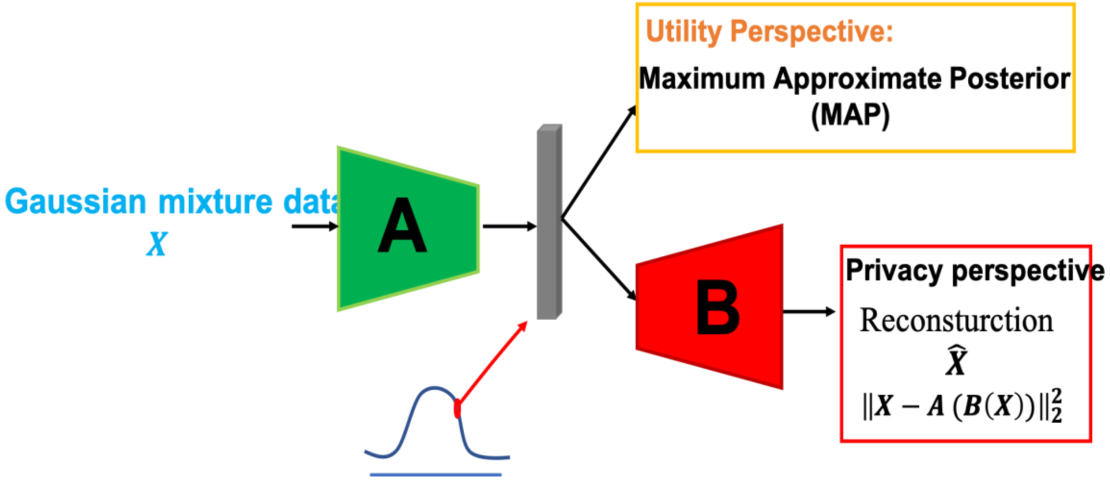
$$
\begin{aligned}
\frac{Q(X|s=0)}{Q(X|s=1)} &\underset{s_0}{\overset{s_1}{\gtrless}} \frac{q}{1-q} \\
\frac{e^{-(\vec{x}-\vec{\mu})^T \Sigma_D^{-1} (x-\mu)}}{e^{-(\vec{x}+\vec{\mu})^T \Sigma_D^{-1} (\vec{x}+\vec{\mu})}} &= \frac{q}{1-q} \\
2x^T \Sigma_D^{-1} \vec{\mu} + 2\vec{\mu}^T \Sigma_D^{-1} \vec{x} &= 2ln(\frac{q}{1-q}) \\
Let\ c &= \Sigma_D^{-1} \vec{\mu} \\
\therefore c^T \vec{x} &= \frac{ln(\frac{q}{1-q})}{2}
\end{aligned}
\tag{1}
$$

therefore, the $\vec{\boldsymbol{x}} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots]$ only has the deterministic solution in $\boldsymbol{x_1}$, the other has infinite solution. The following is the 2-dimension example:

$$
\int_{x_1}^{\infty} e^{\frac{-(x_1+\alpha)^2}{2}} dx_1 \int_{-\infty}^{\infty} e^{\frac{-(x_2)^2}{2}} dx_2 \tag{2}
$$

# Privacy Loss

$$
\begin{aligned}
L_{rec} &= \mathbb{E}_{X\sim P_{data}} \left\| X - \hat{X} \right\|_2^2 \\
&= \mathbb{E}_{X\sim P_{data}} \| X - B(AX+\epsilon) \|_2^2 \\
&= \mathbb{E}_{x\sim P_{data}} \| (I - BA)X - B\epsilon \|_2^2 \\
&= Tr(\mathbb{E}_x[(I-BA)X - B\epsilon][(I-BA)X - B\epsilon]^T)
\end{aligned}
$$




- Zero gradient with respect to B:

$$-2R_x A^T + 2BR_\epsilon + 2B(AR_x A^T) = 0$$

$$\therefore B = R_x A^T (AR_x A^T + R_\epsilon)^{-1}$$

Substitute the solution into B, than find A follows the loss below:

$$\boxed{\max_A Tr(R_x) - Tr(R_x A^T (AR_x A^T + R_\epsilon)^{-1} AR_x)}$$

# Theory MSE

- Since the Q function is increasing with the alpha. Out Optimization becomes:

### 1.3 Combination

- The combination of the alpha and MSE loss above is:

$$\max_{A} Tr(R_{\mathrm{x}}) - Tr(R_{\mathrm{x}} A^T (A R_{\mathrm{x}} A^T + \mathbf{R}_\epsilon)^{-1} A R_{\mathrm{x}}) - \lambda (2A\mu)^T (A R_\xi A^T + \mathbf{R}_\epsilon)^{-1} (2A\mu) \tag{7}$$

- And zero gradient with respect to A (assuming $\mathbf{R}_\epsilon = \mathbf{0}$)

$$\begin{aligned} 0 = -3((A R_{\mathrm{x}} A^T)^{-1} A (R_{\mathrm{x}})^2 + (R_{\mathrm{x}})^2 A^T (A R_{\mathrm{x}} A^T)^{-1} + \\ -2(A R_{\mathrm{x}} A^T)^{-1} A R_{\mathrm{x}}^2 A^T (A R_{\mathrm{x}} A^T)^{-1} A R_{\mathrm{x}} + \\ 4\lambda((A \Sigma_\xi A^T)^{-1} A \vec{\mu} \vec{\mu}^T + 2 \vec{\mu} \vec{\mu}^T (A \Sigma_\xi A^T)^{-1} A A^T (A \Sigma_\xi A^T)^{-1} A \Sigma_\xi) \end{aligned}$$

**This is really intractable !!!**

# LRR and KRR formulation

Assuming that training data matrix $\mathbf{X}$ and output value matrix $\mathbf{Y}$ are both zero mean. Thus, $\mathbf{S} = \mathbf{X}\mathbf{X}^T$ and the bias term is no longer useful.

$$\begin{aligned} \boldsymbol{E_{LSE}} &= \left\| \boldsymbol{X^T W - Y} \right\|_2^2 \\ &= \boldsymbol{Tr((X^T W - Y)(X^T W - Y)^T)} \end{aligned} \tag{11}$$

zero gradient with respect to $\mathbf{W}$, we get:

$$\begin{aligned} \mathbf{0} &= \mathbf{2XX}^T \boldsymbol{W} - \mathbf{2XY} + \mathbf{2}\rho\mathbf{W} \\ \mathbf{W} &= (\mathbf{S} + \rho\mathbf{I})^{-1}\mathbf{XY} \end{aligned} \tag{12}$$

# LFW Accuracy Comparison

| | Raw images | Reconstruction images |
|---|---|---|
| Recognition Accuracy | 67.2% (train)<br>7% (validation)<br>1% (testing) | 0% (train)<br>0% (validation)<br>0% (testing) |
| LFW accuracy | 91.83% | 72% |