# Invertible Tabular GANs: Killing Two Birds with One Stone for Tabular Data Synthesis

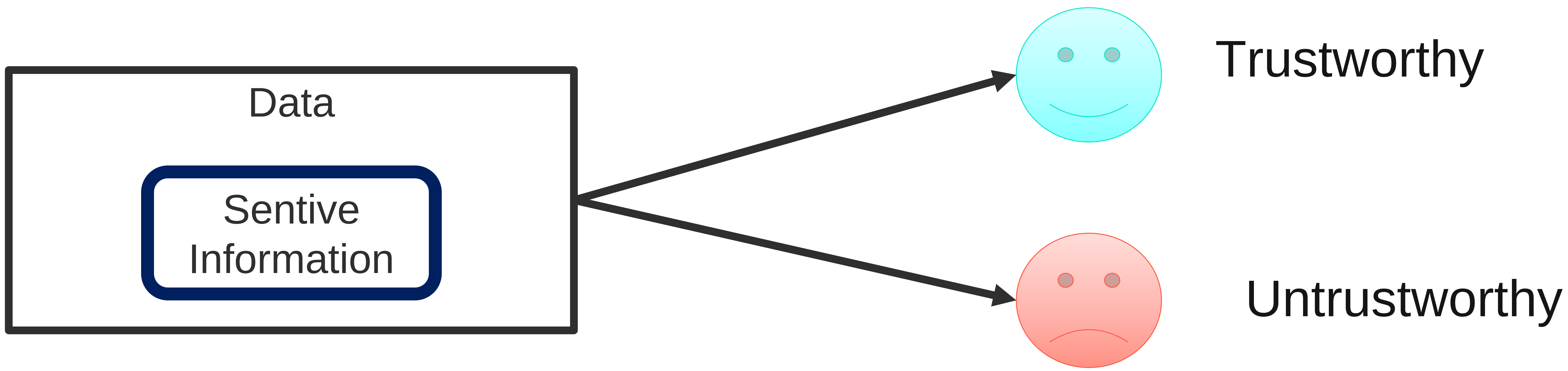
**JAEHOON LEE, Jihyen Hyeong, Jinsung Jeon, Noseong Park, Jihhon Cho**

# Contents

# Introduction

## Motivation

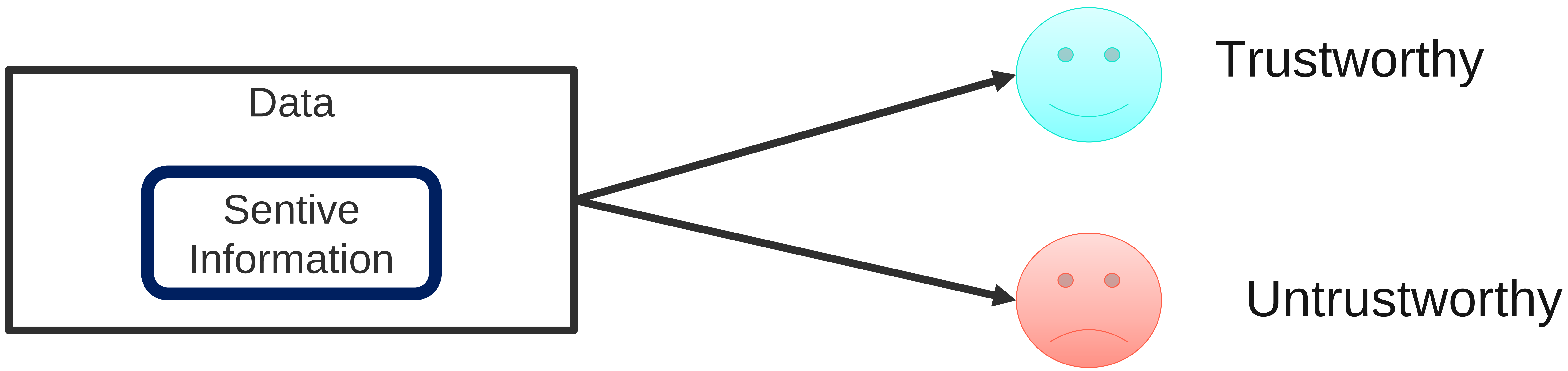
- Tabular data usually has Sensitive Information which can cause privacy problem.
- We share our data with not only Trustworthy people but also Untrustworthy people



# Introduction

## Motivation

- Tabular data usually has Sensitive Information which can cause privacy problem.
- We share our data with not only Trustworthy people but also Untrustworthy people

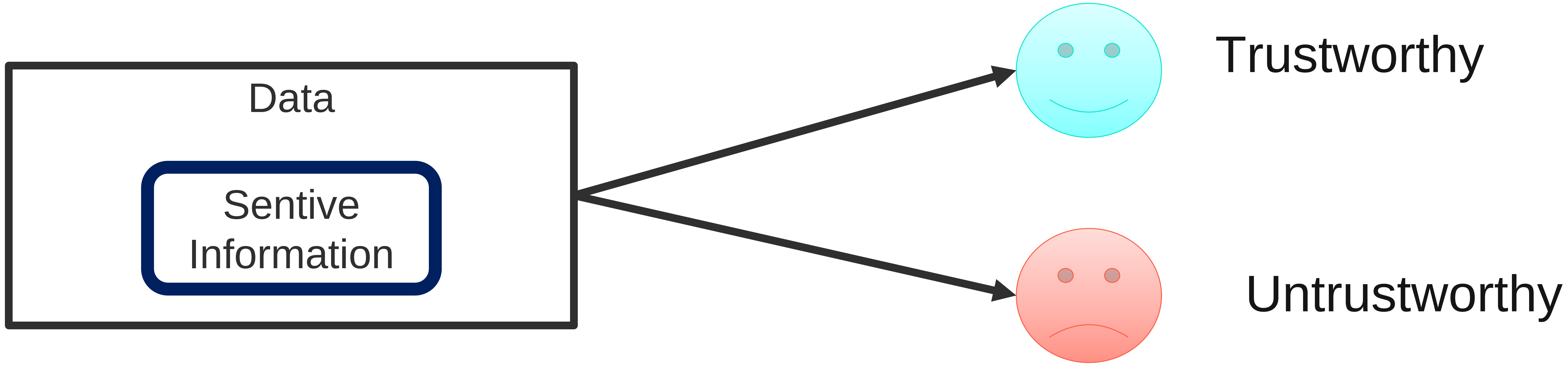



- Privacy Information can be extracted easily from data synthesized by generative models, when the Likelihood(Log-density) of synthesized data is high.

# Introduction

## Motivation

- Tabular data usually has Sensitive Information which can cause privacy problem.
- We share our data with not only Trustworthy people but also Untrustworthy people



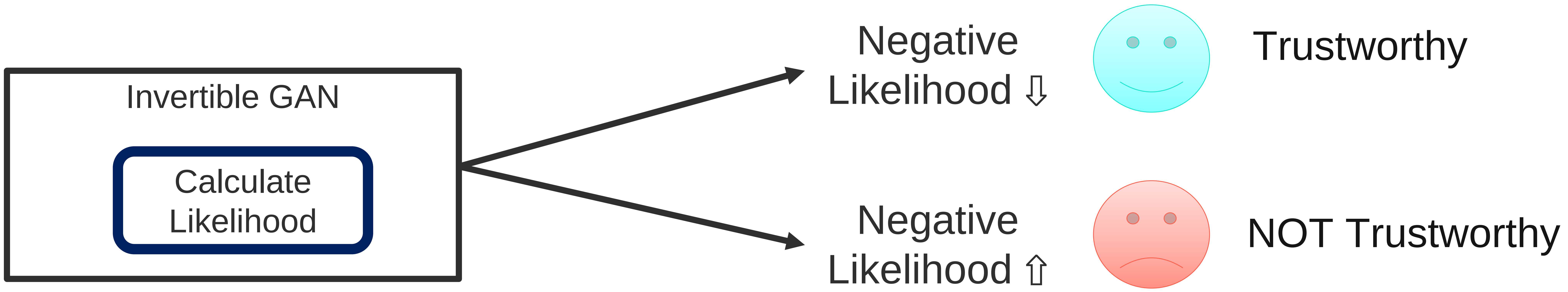

- Privacy Information can be extracted easily from data synthesized by generative models, when the Likelihood(Log-density) of synthesized data is high.

- VAEs generates blurred samples but achieves a better likelihood than GANs.

# Introduction

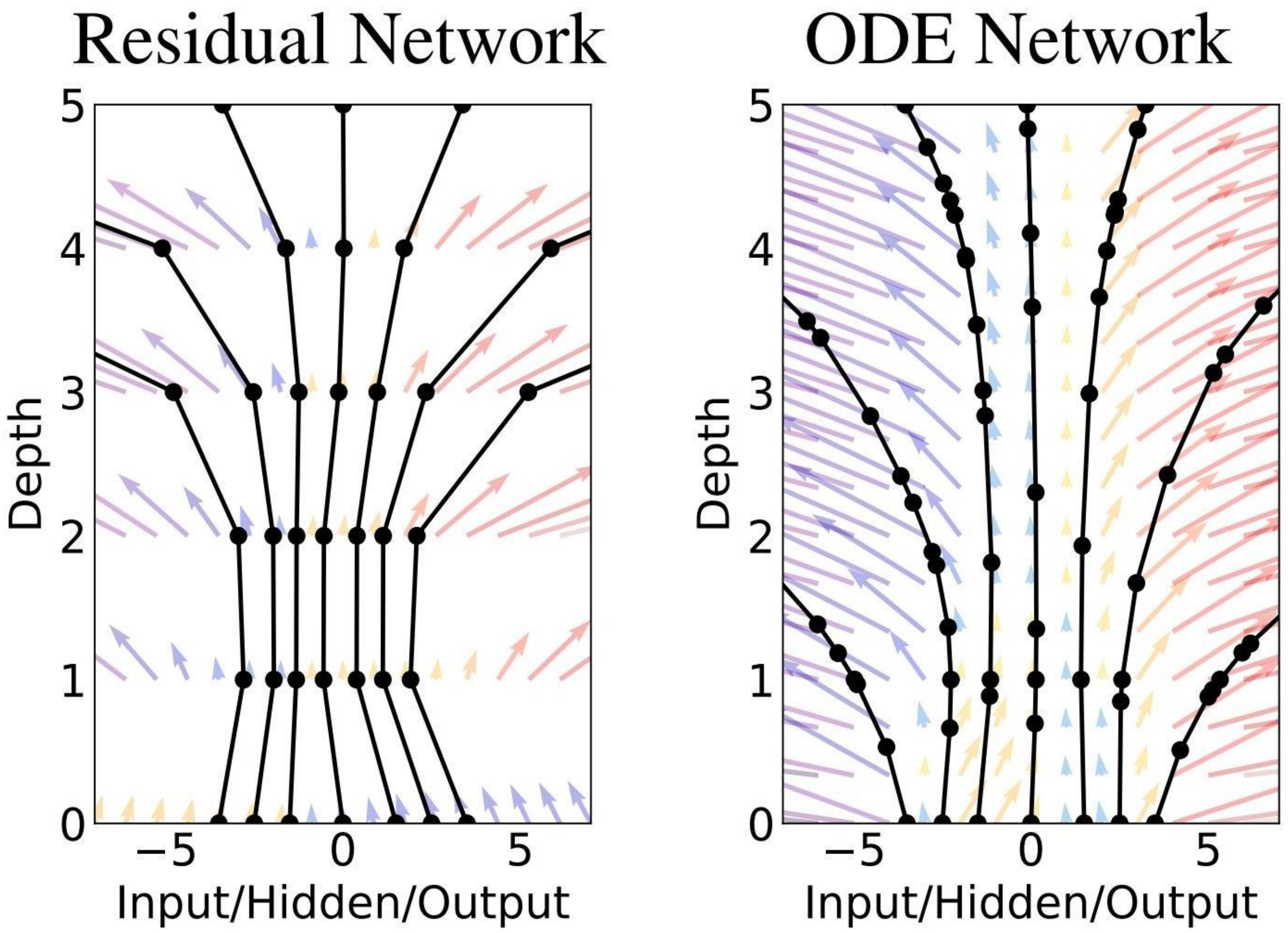
## Proposed Concept

- Making GAN invertible, GAN can be trained with both likelihood loss and GAN loss.
- When sharing with Trustworthy people, decrease GAN loss and the negative likelihood loss
- When sharing with Untrustworthy people, decrease GAN loss, but sacrifice the negative likelihood loss.



# Related Work

## Neural ODE – Invertible Function

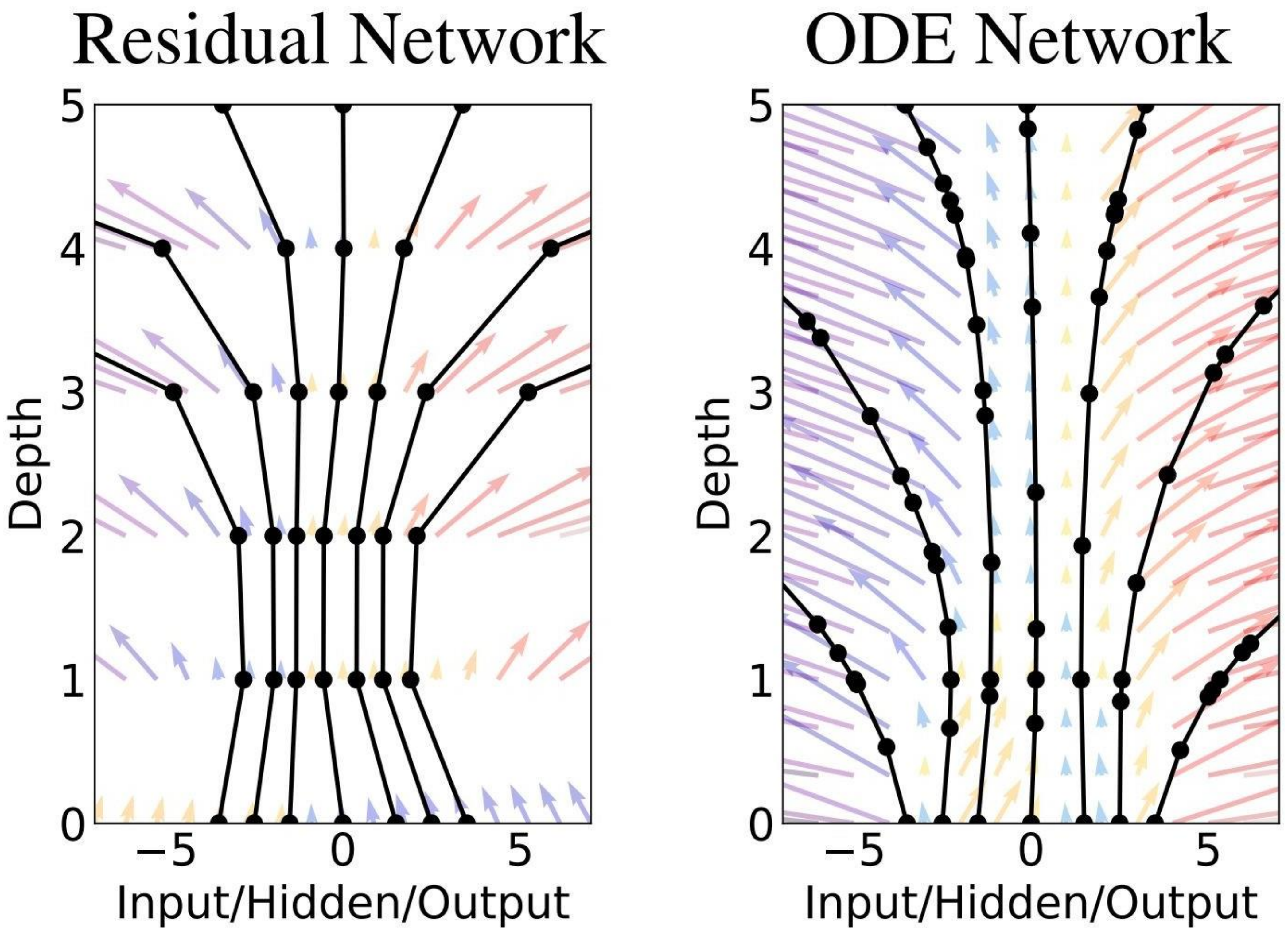



- Solve $z(t_1)$, given the initial condition $z(t_0)$.
- $\frac{\partial z}{\partial t}$ is parameterized by $\theta_f$.

$$z(t_1) = z(t_0) + \int_{t_0}^{t_1} f\big(z(t), t, \theta_f\big)dt, \qquad \frac{\partial z}{\partial t} = f\big(z(t), t, \theta_f\big)$$

# Related Work

## Neural ODE – Invertible Function




- Solve $z(t_1)$, given the initial condition $z(t_0)$.
- $\frac{\partial z}{\partial t}$ is parameterized by $\theta_f$.

$$z(t_1) = z(t_0) + \int_{t_0}^{t_1} f\big(z(t), t, \theta_f\big)dt, \qquad \frac{\partial z}{\partial t} = f\big(z(t), t, \theta_f\big)$$

- Reconstruct $z(t_0)$ from $z(t_1)$.

$$z(t_0) = z(t_1) + \int_{t_1}^{t_0} f\big(z(t), t, \theta_f\big)dt$$

# Related Work

## Neural ODE, Ffjord – Efficient Determinant Calculation

- $|\frac{\partial z(t_1)}{\partial z(t_0)}|$ is main bottle neck of calculating likelihood.
- It usually costs $\mathrm{O}(D^3)$ or $\mathrm{O}(D^2)$, when $D$ is the size of data dimension.

$$log^{p(z(t_1))} = log^{p(z(t_0))} - log^{|\frac{\partial z(t_1)}{\partial z(t_0)}|}$$

# Related Work

## Neural ODE, Ffjord – Efficient Determinant Calculation

- $|\frac{\partial z(t_1)}{\partial z(t_0)}|$ is main bottle neck of calculating likelihood.
- It usually costs $\mathrm{O}(D^3)$ or $\mathrm{O}(D^2)$, when $D$ is the size of data dimension.

$$log^{p(z(t_1))} = log^{p(z(t_0))} - log^{|\frac{\partial z(t_1)}{\partial z(t_0)}|}$$

- By instantaneous change of variables formula, (Neural ODE)

$$\frac{\partial log^{p(z(t))}}{\partial t} = -Tr\left(\frac{\partial f}{\partial z(t)}\right), \qquad \frac{\partial z}{\partial t} = f\big(z(t), t, \theta_f\big)$$

$$log^{p(z(t_1))} = log^{p(z(t_0))} - \int_{t_0}^{t_1} Tr\left(\frac{\partial f}{\partial z(t)}\right) dt$$

# Related Work

## Neural ODE, Ffjord – Efficient Determinant Calculation

- $|\frac{\partial z(t_1)}{\partial z(t_0)}|$ is main bottle neck of calculating likelihood.
- It usually costs $O(D^3)$ or $O(D^2)$, when $D$ is the size of data dimension.

$$log^{p(z(t_1))} = log^{p(z(t_0))} - log^{|\frac{\partial z(t_1)}{\partial z(t_0)}|}$$

- By instantaneous change of variables formula, (Neural ODE)

$$\frac{\partial log^{p(z(t))}}{\partial t} = -Tr\left(\frac{\partial f}{\partial z(t)}\right), \qquad \frac{\partial z}{\partial t} = f\left(z(t), t, \theta_f\right)$$

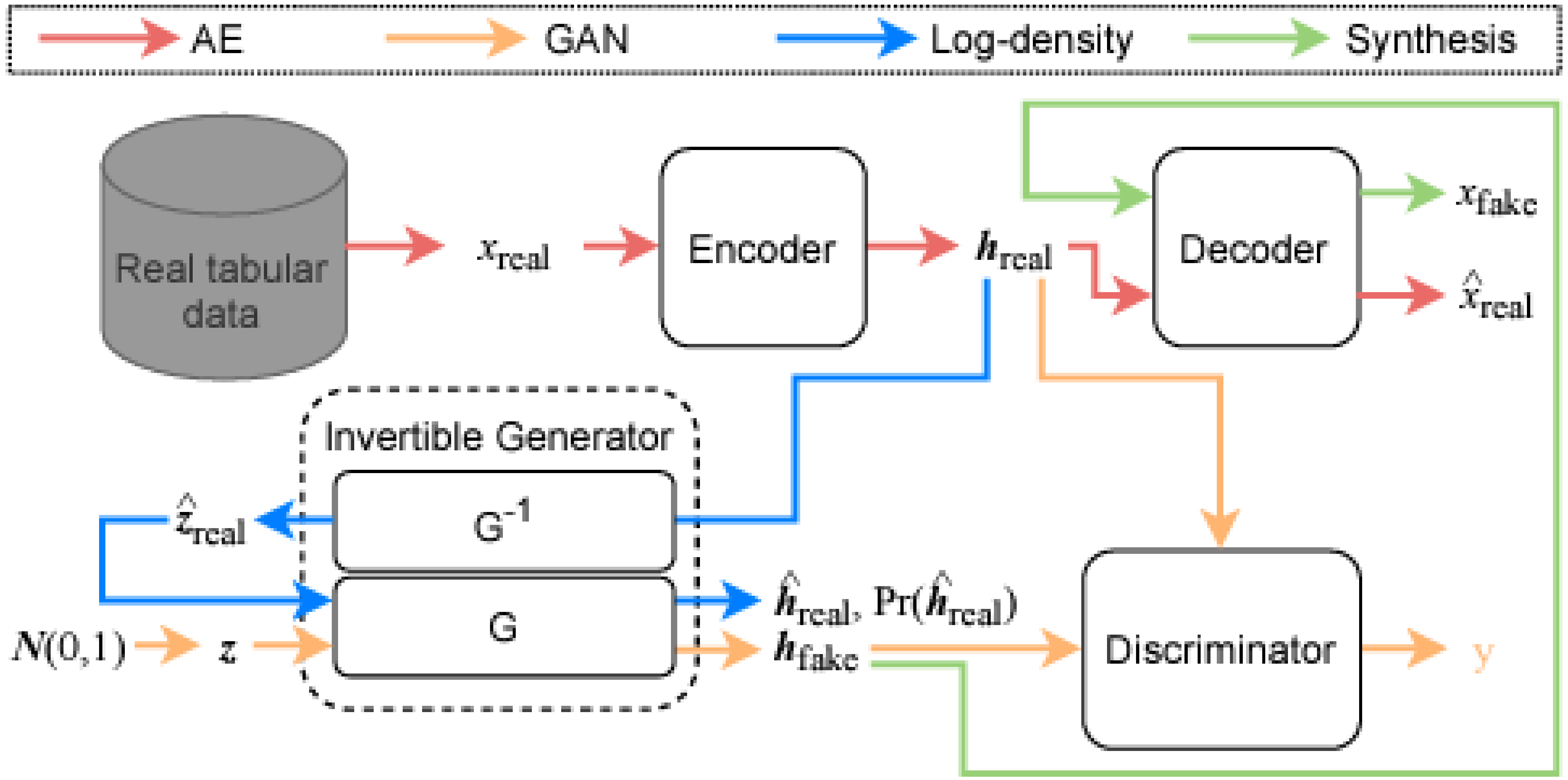
$$log^{p(z(t_1))} = log^{p(z(t_0))} - \int_{t_0}^{t_1} Tr\left(\frac{\partial f}{\partial z(t)}\right) dt$$

- By Hutchinson estimator, $log^{p(z(t_1))}$ can be efficiently calculated. (Ffjord)
- The cost of calculating Hutchinson estimator is slightly larger than that of evaluating $f$, since calculating $\epsilon^T \frac{\partial f}{\partial z(t)}$ has the same cost as $f$, using the reverse-mode automatic differentiation.

$$\frac{\partial log^{p(z(t))}}{\partial t} = -Tr\left(\frac{\partial f}{\partial z(t)}\right) = -E_{p(\epsilon)}[\epsilon^T \frac{\partial f}{\partial z(t)} \epsilon]$$

# Proposed Model

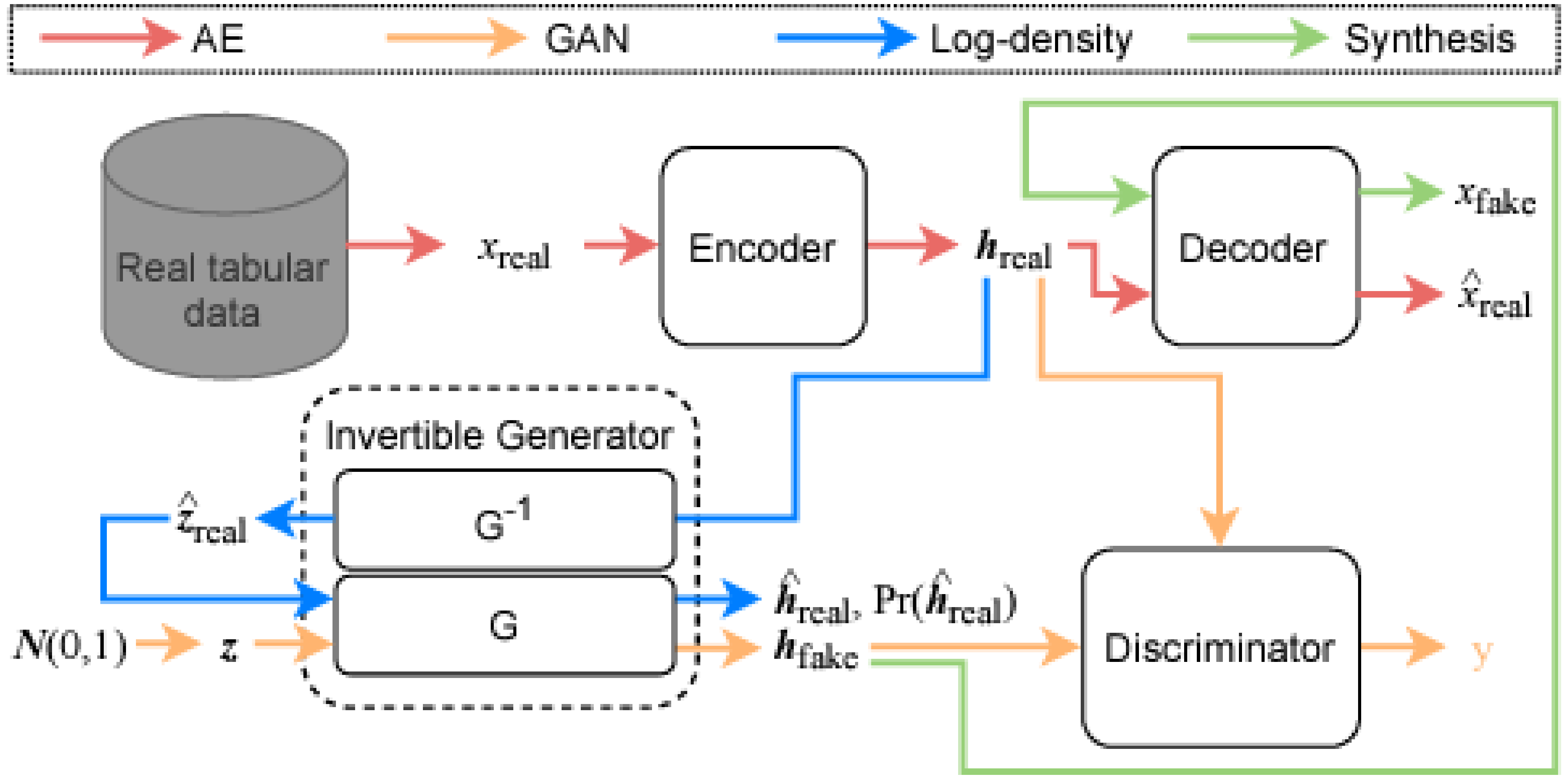
## ITGAN(Invertible Tabular GAN)




- ITGAN synthesizes the data on hidden space, and Decoder recover the real data from that. (Green)
- There are 3 parts in ITGAN. (Red: AutoEncoder, Orange: GAN, Blue: Likelihood(Log-density))

# Proposed Model

## ITGAN(Invertible Tabular GAN) - AutoEncoder

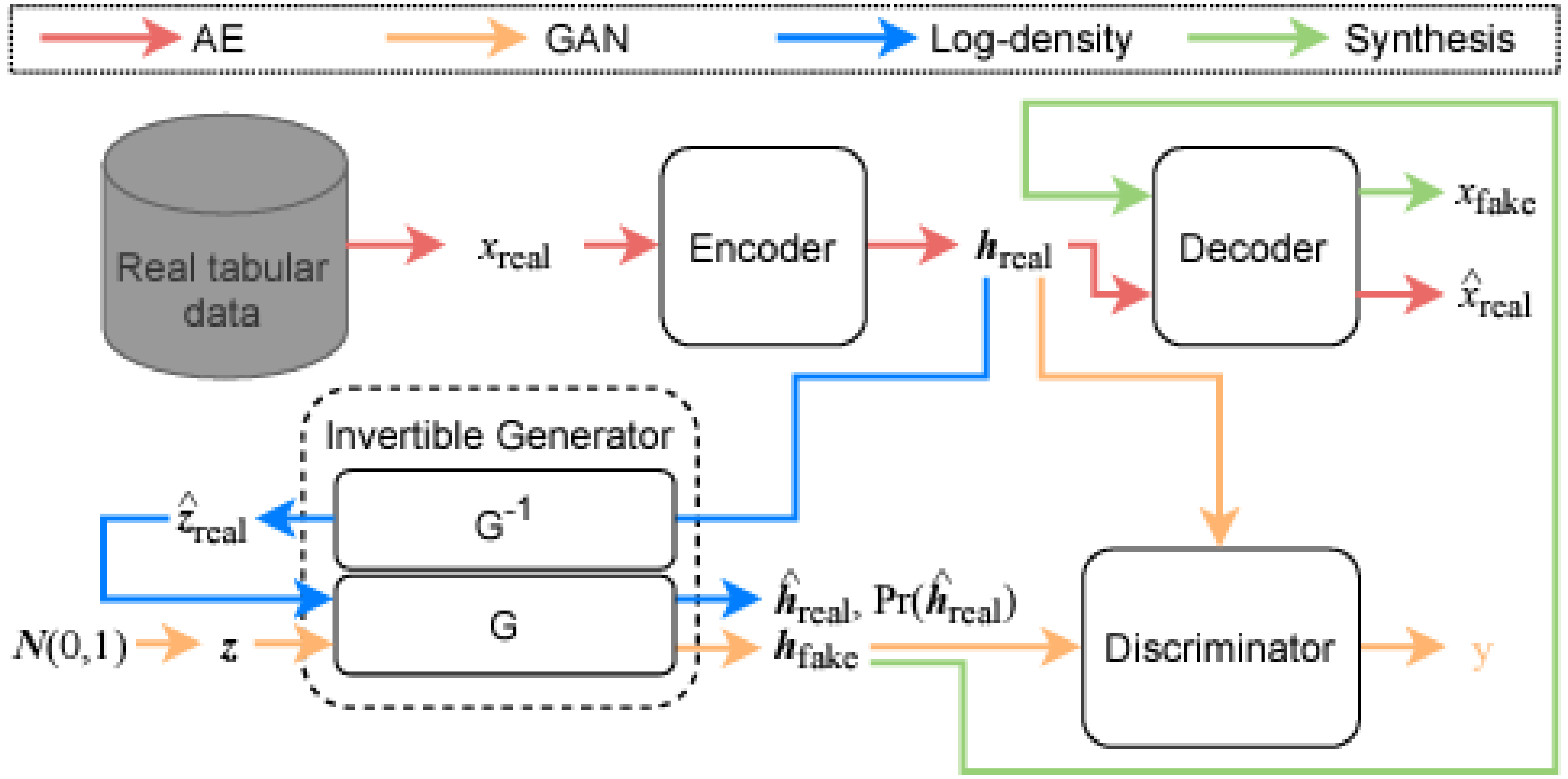



$$L_{AE} = L_{Reconstruct} + \frac{1}{2}||h_{real}||^2 + \frac{1}{2}||h_{fake} - \hat{h}_{fake}||^2$$

- AutoEncoder makes hidden space, and GAN operates on that.
- Using AutoEncoder meets the invariant dimensionality requirement of NODEs and, relieves the burden of GAN by separating the labor.
- AutoEncoder is learned by $L_{AE}$, where $\hat{h}_{fake}$ is a reconstructed hidden vector by $\mathrm{Encoder}(\mathrm{Decoder}(h_{fake}))$.

# Proposed Model

## ITGAN(Invertible Tabular GAN) – GAN, Likelihood(Log-density)




$$h_{fake} = z(0) + \int_0^1 f\big(z(t), t; \theta_g\big)dt$$

$$R_{density} \overset{\text{def}}{=} \gamma E[-log\hat{p}(E(x))]_{x\sim p_{data}}$$

- GAN generates hidden vector with Neural ODE. The integral time is 0 ~ 1.
- GAN is the same with the original WGAN-GP model, except the invertible structure and the operation on hidden space made by autoencoder.
- GAN is trained with $L_{GAN}$ and $R_{density}$, where $L_{GAN}$ is the WGAN-GP loss, and $R_{density}$ is negative log-density regularization calculated using Hutchison Estimator.

# Proposed Model

## Training Algorithm

**Algorithm 1:** How to train IT-GAN

**Input:** Training data $D_{train}$, Validating data $D_{val}$, Maximum iteration number $max\_iter$, The training periods $period_D, period_G, period_L$

Initialize $\boldsymbol{\theta}_e$, $\boldsymbol{\theta}_r$, $\boldsymbol{\theta}_g$, $\boldsymbol{\theta}_d$, and $k \leftarrow 0$;
**while** $k < max\_iter$ **do**
  $k \leftarrow k + 1$;
  Train $\boldsymbol{\theta}_e$ and $\boldsymbol{\theta}_r$ with $L_{AE}$ and $L_{GAN}$;
  **if** $k \mod period_D = 0$ **then**
    Train $\boldsymbol{\theta}_d$ with $L_{GAN}$;
  **end**
  **if** $k \mod period_G = 0$ **then**
    Train $\boldsymbol{\theta}_g$ with $L_{GAN}$;
  **end**
  **if** $k \mod period_L = 0$ **then**
    Train $\boldsymbol{\theta}_g$ with $R_{density}$;
  **end**
  Validate and update the best parameters, $\boldsymbol{\theta}_e^*$, $\boldsymbol{\theta}_r^*$, $\boldsymbol{\theta}_g^*$, and $\boldsymbol{\theta}_d^*$, with $D_{val}$;
**end**
**return** $\boldsymbol{\theta}_e^*$, $\boldsymbol{\theta}_r^*$, $\boldsymbol{\theta}_g^*$, and $\boldsymbol{\theta}_d^*$;

- AutoEncoder
  - Train with $L_{AE}$ every iteration
  - Training the Encoder with $L_{GAN}$ helps the discriminator better distinguish real and fake hidden vectors by learning a hidden vector in favor of the discriminator.

# Proposed Model

## Training Algorithm

**Algorithm 1:** How to train IT-GAN

**Input:** Training data $D_{train}$, Validating data $D_{val}$, Maximum iteration number $max\_iter$, The training periods $period_D, period_G, period_L$

Initialize $\boldsymbol{\theta}_e$, $\boldsymbol{\theta}_r$, $\boldsymbol{\theta}_g$, $\boldsymbol{\theta}_d$, and $k \leftarrow 0$;
**while** $k < max\_iter$ **do**
  $k \leftarrow k + 1$;
  Train $\boldsymbol{\theta}_e$ and $\boldsymbol{\theta}_r$ with $L_{AE}$ and $L_{GAN}$;
  **if** $k \mod period_D = 0$ **then**
    Train $\boldsymbol{\theta}_d$ with $L_{GAN}$;
  **end**
  **if** $k \mod period_G = 0$ **then**
    Train $\boldsymbol{\theta}_g$ with $L_{GAN}$;
  **end**
  **if** $k \mod period_L = 0$ **then**
    Train $\boldsymbol{\theta}_g$ with $R_{density}$;
  **end**
  Validate and update the best parameters, $\boldsymbol{\theta}_e^*$, $\boldsymbol{\theta}_r^*$, $\boldsymbol{\theta}_g^*$, and $\boldsymbol{\theta}_d^*$, with $D_{val}$;
**end**
**return** $\boldsymbol{\theta}_e^*$, $\boldsymbol{\theta}_r^*$, $\boldsymbol{\theta}_g^*$, and $\boldsymbol{\theta}_d^*$;

- AutoEncoder
  - Train with $L_{AE}$ every iteration
  - Training the Encoder with $L_{GAN}$ helps the discriminator better distinguish real and fake hidden vectors by learning a hidden vector in favor of the discriminator.

- GAN, Likelihood(Log-density)
  - generator, discriminator are trained with $L_{GAN}$ every each period, where AutoEncoder is trained every iterations. Because Generator and Discriminator rely on the hidden vector by the AutoEncoder.
  - log-density regularization also has a period, since frequent log-density regularization negatively affects the entire training progress.

# Experiments

## Evaluation Score(Binary Classification)

Table 1: Classification in Adult

| Method | F1 | ROCAUC | Dist. |
|---|---|---|---|
| Real | 0.66±0.00 | 0.88±0.00 | 0.00±0.00 |
| PrivBN | 0.43±0.02 | 0.84±0.01 | 0.35±0.05 |
| TVAE | 0.62±0.01 | 0.84±0.01 | 0.16±0.04 |
| TGAN | <u>0.63±0.01</u> | 0.85±0.01 | 0.22±0.02 |
| TableGAN | 0.46±0.03 | 0.81±0.01 | 0.42±0.03 |
| IT-GAN(Q) | **0.64±0.01** | **0.86±0.00** | 0.33±0.03 |
| IT-GAN(L) | **0.64±0.01** | 0.85±0.01 | 0.41±0.09 |
| IT-GAN | **0.64±0.01** | <u>0.86±0.01</u> | 0.30±0.01 |

Table 2: Classification in Census

| Method | F1 | ROCAUC | Dist. |
|---|---|---|---|
| Real | 0.47±0.01 | 0.90± 0.00 | 0.00±0.00 |
| PrivBN | 0.23±0.03 | 0.81±0.03 | 0.57±0.05 |
| TVAE | 0.44±0.01 | 0.86±0.01 | 0.14±0.01 |
| TGAN | 0.38±0.03 | 0.86±0.02 | 0.14±0.02 |
| TableGAN | 0.31±0.06 | 0.81±0.03 | 0.30±0.06 |
| IT-GAN(Q) | <u>0.45±0.01</u> | **0.89±0.00** | 0.32±0.05 |
| IT-GAN(L) | **0.46±0.01** | 0.88±0.01 | 0.49±0.10 |
| IT-GAN | <u>0.45±0.01</u> | <u>0.88±0.00</u> | 0.32±0.05 |

Test score of Regression/Classification Machine Learning Model
trained with synthesized samples of each generative model

# Experiments

## Evaluation Score(Multi-Class Classification)

Table 3: Classification in Credit

| Method | Macro F1 | Micro F1 | ROCAUC | Dist. |
|---|---|---|---|---|
| Real | 0.48±0.01 | 0.61±0.00 | 0.67±0.00 | 0.00±0.00 |
| Ind | 0.27±0.01 | 0.44±0.01 | 0.51±0.01 | 0.75±0.04 |
| PrivBN | 0.32±0.02 | 0.51±0.01 | **0.60±0.00** | 0.99±0.11 |
| TVAE | 0.39±0.00 | **0.57±0.00** | 0.58±0.00 | 0.27±0.03 |
| TGAN | 0.40±0.00 | 0.55±0.01 | 0.59±0.00 | 0.44±0.03 |
| MedGAN | 0.37±0.02 | 0.51±0.03 | 0.56±0.01 | 26.37±2.87 |
| IT-GAN(Q) | **0.41±0.01** | 0.54±0.01 | **0.60±0.00** | 0.52±0.06 |
| IT-GAN(L) | 0.40±0.01 | 0.55±0.01 | 0.60±0.01 | 0.69±0.04 |
| IT-GAN | 0.40±0.00 | 0.54±0.01 | 0.59±0.01 | 0.59±0.04 |

Table 4: Classification in Cabs

| Method | Macro F1 | Micro F1 | ROCAUC | Dist. |
|---|---|---|---|---|
| Real | 0.65±0.00 | 0.68±0.00 | 0.78±0.00 | 0.00±0.00 |
| PrivBN | 0.64±0.00 | 0.67±0.00 | 0.77±0.00 | 0.52±0.02 |
| TVAE | 0.60±0.02 | 0.66±0.01 | 0.74±0.01 | 0.33±0.01 |
| TGAN | 0.64±0.00 | 0.67±0.00 | 0.76±0.00 | 0.49±0.02 |
| VeeGAN | 0.54±0.06 | 0.60±0.05 | 0.71±0.02 | 1.46±0.12 |
| IT-GAN(Q) | **0.66±0.00** | **0.69±0.00** | 0.79±0.01 | 0.56±0.06 |
| IT-GAN(L) | 0.66±0.01 | 0.68±0.01 | **0.79±0.00** | 0.64±0.05 |
| IT-GAN | 0.64±0.01 | 0.67±0.01 | 0.77±0.00 | 0.53±0.04 |

Test score of Regression/Classification Machine Learning Model
trained with synthesized samples of each generative model

# Experiments

## Evaluation Score(Regression)

Table 5: Regression in King

| Method | $R^2$ | Ex. Var. | MSE | MAE | Dist. |
|---|---|---|---|---|---|
| Real | 0.50±0.11 | 0.61±0.02 | 0.14±0.03 | 0.30±0.03 | 0.00±0.00 |
| TVAE | 0.44±0.01 | 0.52±0.04 | 0.16±0.00 | 0.32±0.01 | 0.29±0.02 |
| TGAN | 0.43±0.01 | **0.60±0.00** | 0.16±0.00 | 0.32±0.00 | 0.55±0.02 |
| TableGAN | 0.41±0.02 | 0.46±0.03 | 0.17±0.01 | 0.33±0.01 | 0.61±0.03 |
| VeeGAN | 0.25±0.15 | 0.32±0.14 | 0.21±0.04 | 0.37±0.03 | 4.61±2.22 |
| IT-GAN(Q) | **0.59±0.00** | **0.60±0.00** | **0.12±0.00** | 0.28±0.00 | 0.60±0.02 |
| IT-GAN(L) | 0.53±0.01 | 0.56±0.01 | 0.13±0.00 | 0.29±0.00 | 1.09±0.19 |
| IT-GAN | 0.59±0.01 | 0.60±0.01 | **0.12±0.00** | **0.27±0.00** | 0.64±0.06 |

Table 6: Regression in News (Ex. Var. means explained variance.)

| Method | $R^2$ | Ex.Var | MSE | MAE | Dist. |
|---|---|---|---|---|---|
| Real | 0.15±0.01 | 0.15±0.00 | 0.69± 0.00 | 0.63±0.01 | 0.00±0.00 |
| TVAE | -0.09±0.03 | 0.03±0.04 | 0.88±0.03 | 0.67±0.01 | 1.84±0.04 |
| TGAN | 0.06±0.02 | 0.07±0.01 | 0.76±0.01 | 0.66±0.02 | 1.97±0.05 |
| IT-GAN(Q) | **0.09±0.01** | 0.09±0.01 | **0.74±0.01** | 0.65±0.01 | 2.31±0.02 |
| IT-GAN(L) | 0.03±0.03 | 0.06±0.02 | 0.78±0.03 | 0.65±0.01 | 2.45±0.06 |
| IT-GAN | 0.09±0.02 | **0.10±0.01** | **0.74±0.01** | **0.64±0.00** | 2.37±0.03 |

Test score of Regression/Classification Machine Learning Model
trained with synthesized samples of each generative model

# Experiments

## Evaluation Score

Table 1: Classification in `Adult`

| Method | F1 | ROCAUC | Dist. |
|---|---|---|---|
| Real | 0.66±0.00 | 0.88±0.00 | 0.00±0.00 |
| PrivBN | 0.43±0.02 | 0.84±0.01 | 0.35±0.05 |
| TVAE | 0.62±0.01 | 0.84±0.01 | 0.16±0.04 |
| TGAN | 0.63±0.01 | 0.85±0.01 | 0.22±0.02 |
| TableGAN | 0.46±0.03 | 0.81±0.01 | 0.42±0.03 |
| IT-GAN(Q) | **0.64±0.01** | **0.86±0.00** | 0.33±0.03 |
| IT-GAN(L) | **0.64±0.01** | 0.85±0.01 | 0.41±0.09 |
| IT-GAN | **0.64±0.01** | 0.86±0.01 | 0.30±0.01 |

Table 2: Classification in `Census`

| Method | F1 | ROCAUC | Dist. |
|---|---|---|---|
| Real | 0.47±0.01 | 0.90± 0.00 | 0.00±0.00 |
| PrivBN | 0.23±0.03 | 0.81±0.03 | 0.57±0.05 |
| TVAE | 0.44±0.01 | 0.86±0.01 | 0.14±0.01 |
| TGAN | 0.38±0.03 | 0.86±0.02 | 0.14±0.02 |
| TableGAN | 0.31±0.06 | 0.81±0.03 | 0.30±0.06 |
| IT-GAN(Q) | 0.45±0.01 | **0.89±0.00** | 0.32±0.05 |
| IT-GAN(L) | **0.46±0.01** | 0.88±0.01 | 0.49±0.10 |
| IT-GAN | 0.45±0.01 | 0.88±0.00 | 0.32±0.05 |

Table 3: Classification in `Credit`

| Method | Macro F1 | Micro F1 | ROCAUC | Dist. |
|---|---|---|---|---|
| Real | 0.48±0.01 | 0.61±0.00 | 0.67±0.00 | 0.00±0.00 |
| Ind | 0.27±0.01 | 0.44±0.01 | 0.51±0.01 | 0.75±0.04 |
| PrivBN | 0.32±0.02 | 0.51±0.01 | **0.60±0.00** | 0.99±0.11 |
| TVAE | 0.39±0.00 | **0.57±0.00** | 0.58±0.00 | 0.27±0.03 |
| TGAN | 0.40±0.00 | 0.55±0.01 | 0.59±0.00 | 0.44±0.03 |
| MedGAN | 0.37±0.02 | 0.51±0.03 | 0.56±0.01 | 26.37±2.87 |
| IT-GAN(Q) | **0.41±0.01** | 0.54±0.01 | **0.60±0.00** | 0.52±0.06 |
| IT-GAN(L) | 0.40±0.01 | 0.55±0.01 | 0.60±0.01 | 0.69±0.04 |
| IT-GAN | 0.40±0.00 | 0.54±0.01 | 0.59±0.01 | 0.59±0.04 |

Table 4: Classification in `Cabs`

| Method | Macro F1 | Micro F1 | ROCAUC | Dist. |
|---|---|---|---|---|
| Real | 0.65±0.00 | 0.68±0.00 | 0.78±0.00 | 0.00±0.00 |
| PrivBN | 0.64±0.00 | 0.67±0.00 | 0.77±0.00 | 0.52±0.02 |
| TVAE | 0.60±0.02 | 0.66±0.01 | 0.74±0.01 | 0.33±0.01 |
| TGAN | 0.64±0.00 | 0.67±0.00 | 0.76±0.00 | 0.49±0.02 |
| VeeGAN | 0.54±0.06 | 0.60±0.05 | 0.71±0.02 | 1.46±0.12 |
| IT-GAN(Q) | **0.66±0.00** | **0.69±0.00** | 0.79±0.01 | 0.56±0.06 |
| IT-GAN(L) | 0.66±0.01 | 0.68±0.01 | **0.79±0.00** | 0.64±0.05 |
| IT-GAN | 0.64±0.01 | 0.67±0.01 | 0.77±0.00 | 0.53±0.04 |

Table 5: Regression in `King`

| Method | $R^2$ | Ex. Var. | MSE | MAE | Dist. |
|---|---|---|---|---|---|
| Real | 0.50±0.11 | 0.61±0.02 | 0.14±0.03 | 0.30±0.03 | 0.00±0.00 |
| TVAE | 0.44±0.01 | 0.52±0.04 | 0.16±0.00 | 0.32±0.01 | 0.29±0.02 |
| TGAN | 0.43±0.01 | **0.60±0.00** | 0.16±0.00 | 0.32±0.00 | 0.55±0.02 |
| TableGAN | 0.41±0.02 | 0.46±0.03 | 0.17±0.01 | 0.33±0.01 | 0.61±0.03 |
| VeeGAN | 0.25±0.15 | 0.32±0.14 | 0.21±0.04 | 0.37±0.03 | 4.61±2.22 |
| IT-GAN(Q) | **0.59±0.00** | **0.60±0.00** | **0.12±0.00** | 0.28±0.00 | 0.60±0.02 |
| IT-GAN(L) | 0.53±0.01 | 0.56±0.01 | 0.13±0.00 | 0.29±0.00 | 1.09±0.19 |
| IT-GAN | 0.59±0.01 | 0.60±0.01 | **0.12±0.00** | **0.27±0.00** | 0.64±0.06 |

Table 6: Regression in `News` (Ex. Var. means explained variance.)

| Method | $R^2$ | Ex.Var | MSE | MAE | Dist. |
|---|---|---|---|---|---|
| Real | 0.15±0.01 | 0.15±0.00 | 0.69± 0.00 | 0.63±0.01 | 0.00±0.00 |
| TVAE | -0.09±0.03 | 0.03±0.04 | 0.88±0.03 | 0.67±0.01 | 1.84±0.04 |
| TGAN | 0.06±0.02 | 0.07±0.01 | 0.76±0.01 | 0.66±0.02 | 1.97±0.05 |
| IT-GAN(Q) | **0.09±0.01** | 0.09±0.01 | **0.74±0.01** | 0.65±0.01 | 2.31±0.02 |
| IT-GAN(L) | 0.03±0.03 | 0.06±0.02 | 0.78±0.03 | 0.65±0.01 | 2.45±0.06 |
| IT-GAN | 0.09±0.02 | **0.10±0.01** | **0.74±0.01** | **0.64±0.00** | 2.37±0.03 |

- ITGAN(Q) is a proposed model decreasing the negative log-density, where ITGAN(L) is that sacrificing the negative log-density. ITGAN is without log-density regularizer.
- Dist.(real-fake distance) is the average of the distance from each fake record to its closest real record

# Experiments

## Evaluation Score

Table 1: Classification in Adult

| Method | F1 | ROCAUC | Dist. |
|---|---|---|---|
| Real | 0.66±0.00 | 0.88±0.00 | 0.00±0.00 |
| PrivBN | 0.43±0.02 | 0.84±0.01 | 0.35±0.05 |
| TVAE | 0.62±0.01 | 0.84±0.01 | 0.16±0.04 |
| TGAN | 0.63±0.01 | 0.85±0.01 | 0.22±0.02 |
| TableGAN | 0.46±0.03 | 0.81±0.01 | 0.42±0.03 |
| IT-GAN(Q) | **0.64±0.01** | **0.86±0.00** | 0.33±0.03 |
| IT-GAN(L) | **0.64±0.01** | 0.85±0.01 | 0.41±0.09 |
| IT-GAN | **0.64±0.01** | 0.86±0.01 | 0.30±0.01 |

Table 2: Classification in Census

| Method | F1 | ROCAUC | Dist. |
|---|---|---|---|
| Real | 0.47±0.01 | 0.90± 0.00 | 0.00±0.00 |
| PrivBN | 0.23±0.03 | 0.81±0.03 | 0.57±0.05 |
| TVAE | 0.44±0.01 | 0.86±0.01 | 0.14±0.01 |
| TGAN | 0.38±0.03 | 0.86±0.02 | 0.14±0.02 |
| TableGAN | 0.31±0.06 | 0.81±0.03 | 0.30±0.06 |
| IT-GAN(Q) | 0.45±0.01 | **0.89±0.00** | 0.32±0.05 |
| IT-GAN(L) | **0.46±0.01** | 0.88±0.01 | 0.49±0.10 |
| IT-GAN | 0.45±0.01 | 0.88±0.00 | 0.32±0.05 |

Table 3: Classification in Credit

| Method | Macro F1 | Micro F1 | ROCAUC | Dist. |
|---|---|---|---|---|
| Real | 0.48±0.01 | 0.61±0.00 | 0.67±0.00 | 0.00±0.00 |
| Ind | 0.27±0.01 | 0.44±0.01 | 0.51±0.01 | 0.75±0.04 |
| PrivBN | 0.32±0.02 | 0.51±0.01 | **0.60±0.00** | 0.99±0.11 |
| TVAE | 0.39±0.00 | **0.57±0.00** | 0.58±0.00 | 0.27±0.03 |
| TGAN | 0.40±0.00 | 0.55±0.01 | 0.59±0.00 | 0.44±0.03 |
| MedGAN | 0.37±0.02 | 0.51±0.03 | 0.56±0.01 | 26.37±2.87 |
| IT-GAN(Q) | **0.41±0.01** | 0.54±0.01 | **0.60±0.00** | 0.52±0.06 |
| IT-GAN(L) | 0.40±0.01 | 0.55±0.01 | 0.60±0.01 | 0.69±0.04 |
| IT-GAN | 0.40±0.00 | 0.54±0.01 | 0.59±0.01 | 0.59±0.04 |

Table 4: Classification in Cabs

| Method | Macro F1 | Micro F1 | ROCAUC | Dist. |
|---|---|---|---|---|
| Real | 0.65±0.00 | 0.68±0.00 | 0.78±0.00 | 0.00±0.00 |
| PrivBN | 0.64±0.00 | 0.67±0.00 | 0.77±0.00 | 0.52±0.02 |
| TVAE | 0.60±0.02 | 0.66±0.01 | 0.74±0.01 | 0.33±0.01 |
| TGAN | 0.64±0.00 | 0.67±0.00 | 0.76±0.00 | 0.49±0.02 |
| VeeGAN | 0.54±0.06 | 0.60±0.05 | 0.71±0.02 | 1.46±0.12 |
| IT-GAN(Q) | **0.66±0.00** | **0.69±0.00** | 0.79±0.01 | 0.56±0.06 |
| IT-GAN(L) | 0.66±0.01 | 0.68±0.01 | **0.79±0.00** | 0.64±0.05 |
| IT-GAN | 0.64±0.01 | 0.67±0.01 | 0.77±0.00 | 0.53±0.04 |

Table 5: Regression in King

| Method | $R^2$ | Ex. Var. | MSE | MAE | Dist. |
|---|---|---|---|---|---|
| Real | 0.50±0.11 | 0.61±0.02 | 0.14±0.03 | 0.30±0.03 | 0.00±0.00 |
| TVAE | 0.44±0.01 | 0.52±0.04 | 0.16±0.00 | 0.32±0.01 | 0.29±0.02 |
| TGAN | 0.43±0.01 | **0.60±0.00** | 0.16±0.00 | 0.32±0.00 | 0.55±0.02 |
| TableGAN | 0.41±0.02 | 0.46±0.03 | 0.17±0.01 | 0.33±0.01 | 0.61±0.03 |
| VeeGAN | 0.25±0.15 | 0.32±0.14 | 0.21±0.04 | 0.37±0.03 | 4.61±2.22 |
| IT-GAN(Q) | **0.59±0.00** | **0.60±0.00** | **0.12±0.00** | 0.28±0.00 | 0.60±0.02 |
| IT-GAN(L) | 0.53±0.01 | 0.56±0.01 | 0.13±0.00 | 0.29±0.00 | 1.09±0.19 |
| IT-GAN | 0.59±0.01 | 0.60±0.01 | **0.12±0.00** | **0.27±0.00** | 0.64±0.06 |

Table 6: Regression in News (Ex. Var. means explained variance.)

| Method | $R^2$ | Ex.Var | MSE | MAE | Dist. |
|---|---|---|---|---|---|
| Real | 0.15±0.01 | 0.15±0.00 | 0.69± 0.00 | 0.63±0.01 | 0.00±0.00 |
| TVAE | -0.09±0.03 | 0.03±0.04 | 0.88±0.03 | 0.67±0.01 | 1.84±0.04 |
| TGAN | 0.06±0.02 | 0.07±0.01 | 0.76±0.01 | 0.66±0.02 | 1.97±0.05 |
| IT-GAN(Q) | **0.09±0.01** | 0.09±0.01 | **0.74±0.01** | 0.65±0.01 | 2.31±0.02 |
| IT-GAN(L) | 0.03±0.03 | 0.06±0.02 | 0.78±0.03 | 0.65±0.01 | 2.45±0.06 |
| IT-GAN | 0.09±0.02 | **0.10±0.01** | **0.74±0.01** | **0.64±0.00** | 2.37±0.03 |

- ITGAN(Q) is a proposed model decreasing the negative log-density, where ITGAN(L) is that sacrificing the negative log-density. ITGAN is without log-density regularizer.
- Dist.(real-fake distance) is the average of the distance from each fake record to its closest real record

- All kinds of ITGAN achieve the best score in almost cases.
- Among ITGAN, ITGAN(Q) is the best.

# Experiments

## Evaluation Score

Table 1: Classification in Adult

| Method | F1 | ROCAUC | Dist. |
|---|---|---|---|
| Real | 0.66±0.00 | 0.88±0.00 | 0.00±0.00 |
| PrivBN | 0.43±0.02 | 0.84±0.01 | 0.35±0.05 |
| TVAE | 0.62±0.01 | 0.84±0.01 | 0.16±0.04 |
| TGAN | 0.63±0.01 | 0.85±0.01 | 0.22±0.02 |
| TableGAN | 0.46±0.03 | 0.81±0.01 | 0.42±0.03 |
| IT-GAN(Q) | **0.64±0.01** | **0.86±0.00** | 0.33±0.03 |
| IT-GAN(L) | **0.64±0.01** | 0.85±0.01 | 0.41±0.09 |
| IT-GAN | **0.64±0.01** | 0.86±0.01 | 0.30±0.01 |

Table 2: Classification in Census

| Method | F1 | ROCAUC | Dist. |
|---|---|---|---|
| Real | 0.47±0.01 | 0.90± 0.00 | 0.00±0.00 |
| PrivBN | 0.23±0.03 | 0.81±0.03 | 0.57±0.05 |
| TVAE | 0.44±0.01 | 0.86±0.01 | 0.14±0.01 |
| TGAN | 0.38±0.03 | 0.86±0.02 | 0.14±0.02 |
| TableGAN | 0.31±0.06 | 0.81±0.03 | 0.30±0.06 |
| IT-GAN(Q) | 0.45±0.01 | **0.89±0.00** | 0.32±0.05 |
| IT-GAN(L) | **0.46±0.01** | 0.88±0.01 | 0.49±0.10 |
| IT-GAN | 0.45±0.01 | 0.88±0.00 | 0.32±0.05 |

Table 3: Classification in Credit

| Method | Macro F1 | Micro F1 | ROCAUC | Dist. |
|---|---|---|---|---|
| Real | 0.48±0.01 | 0.61±0.00 | 0.67±0.00 | 0.00±0.00 |
| Ind | 0.27±0.01 | 0.44±0.01 | 0.51±0.01 | 0.75±0.04 |
| PrivBN | 0.32±0.02 | 0.51±0.01 | **0.60±0.00** | 0.99±0.11 |
| TVAE | 0.39±0.00 | **0.57±0.00** | 0.58±0.00 | 0.27±0.03 |
| TGAN | 0.40±0.00 | 0.55±0.01 | 0.59±0.00 | 0.44±0.03 |
| MedGAN | 0.37±0.02 | 0.51±0.03 | 0.56±0.01 | 26.37±2.87 |
| IT-GAN(Q) | **0.41±0.01** | 0.54±0.01 | **0.60±0.00** | 0.52±0.06 |
| IT-GAN(L) | 0.40±0.01 | 0.55±0.01 | 0.60±0.01 | 0.69±0.04 |
| IT-GAN | 0.40±0.00 | 0.54±0.01 | 0.59±0.01 | 0.59±0.04 |

Table 4: Classification in Cabs

| Method | Macro F1 | Micro F1 | ROCAUC | Dist. |
|---|---|---|---|---|
| Real | 0.65±0.00 | 0.68±0.00 | 0.78±0.00 | 0.00±0.00 |
| PrivBN | 0.64±0.00 | 0.67±0.00 | 0.77±0.00 | 0.52±0.02 |
| TVAE | 0.60±0.02 | 0.66±0.01 | 0.74±0.01 | 0.33±0.01 |
| TGAN | 0.64±0.00 | 0.67±0.00 | 0.76±0.00 | 0.49±0.02 |
| VeeGAN | 0.54±0.06 | 0.60±0.05 | 0.71±0.02 | 1.46±0.12 |
| IT-GAN(Q) | **0.66±0.00** | **0.69±0.00** | 0.79±0.01 | 0.56±0.06 |
| IT-GAN(L) | 0.66±0.01 | 0.68±0.01 | **0.79±0.00** | 0.64±0.05 |
| IT-GAN | 0.64±0.01 | 0.67±0.01 | 0.77±0.00 | 0.53±0.04 |

Table 5: Regression in King

| Method | $R^2$ | Ex. Var. | MSE | MAE | Dist. |
|---|---|---|---|---|---|
| Real | 0.50±0.11 | 0.61±0.02 | 0.14±0.03 | 0.30±0.03 | 0.00±0.00 |
| TVAE | 0.44±0.01 | 0.52±0.04 | 0.16±0.00 | 0.32±0.01 | 0.29±0.02 |
| TGAN | 0.43±0.01 | **0.60±0.00** | 0.16±0.00 | 0.32±0.00 | 0.55±0.02 |
| TableGAN | 0.41±0.02 | 0.46±0.03 | 0.17±0.01 | 0.33±0.01 | 0.61±0.03 |
| VeeGAN | 0.25±0.15 | 0.32±0.14 | 0.21±0.04 | 0.37±0.03 | 4.61±2.22 |
| IT-GAN(Q) | **0.59±0.00** | **0.60±0.00** | **0.12±0.00** | 0.28±0.00 | 0.60±0.02 |
| IT-GAN(L) | 0.53±0.01 | 0.56±0.01 | 0.13±0.00 | 0.29±0.00 | 1.09±0.19 |
| IT-GAN | 0.59±0.01 | 0.60±0.01 | **0.12±0.00** | **0.27±0.00** | 0.64±0.06 |

Table 6: Regression in News (Ex. Var. means explained variance.)

| Method | $R^2$ | Ex.Var | MSE | MAE | Dist. |
|---|---|---|---|---|---|
| Real | 0.15±0.01 | 0.15±0.00 | 0.69± 0.00 | 0.63±0.01 | 0.00±0.00 |
| TVAE | -0.09±0.03 | 0.03±0.04 | 0.88±0.03 | 0.67±0.01 | 1.84±0.04 |
| TGAN | 0.06±0.02 | 0.07±0.01 | 0.76±0.01 | 0.66±0.02 | 1.97±0.05 |
| IT-GAN(Q) | **0.09±0.01** | 0.09±0.01 | **0.74±0.01** | 0.65±0.01 | 2.31±0.02 |
| IT-GAN(L) | 0.03±0.03 | 0.06±0.02 | 0.78±0.03 | 0.65±0.01 | 2.45±0.06 |
| IT-GAN | 0.09±0.02 | **0.10±0.01** | **0.74±0.01** | **0.64±0.00** | 2.37±0.03 |

- ITGAN(Q) is a proposed model decreasing the negative log-density, where ITGAN(L) is that sacrificing the negative log-density. ITGAN is without log-density regularizer.
- Dist.(real-fake distance) is the average of the distance from each fake record to its closest real record

- All kinds of ITGAN achieve the best score in almost cases.
- Among ITGAN, ITGAN(Q) is the best.

- Despite sacrificing the negative log-density, ITGAN(L) achieves the reasonable scores.
- Dist. of ITGAN(L) is the biggest.

# Experiments

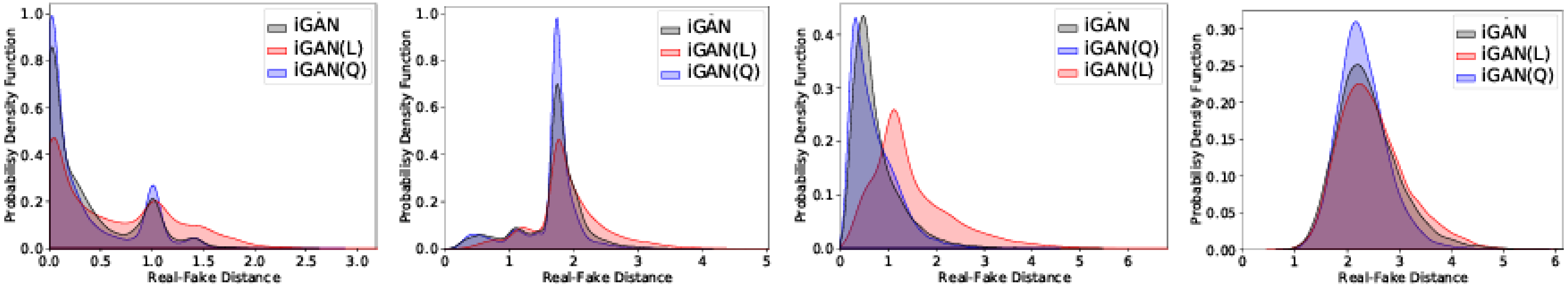
## Privacy Attack

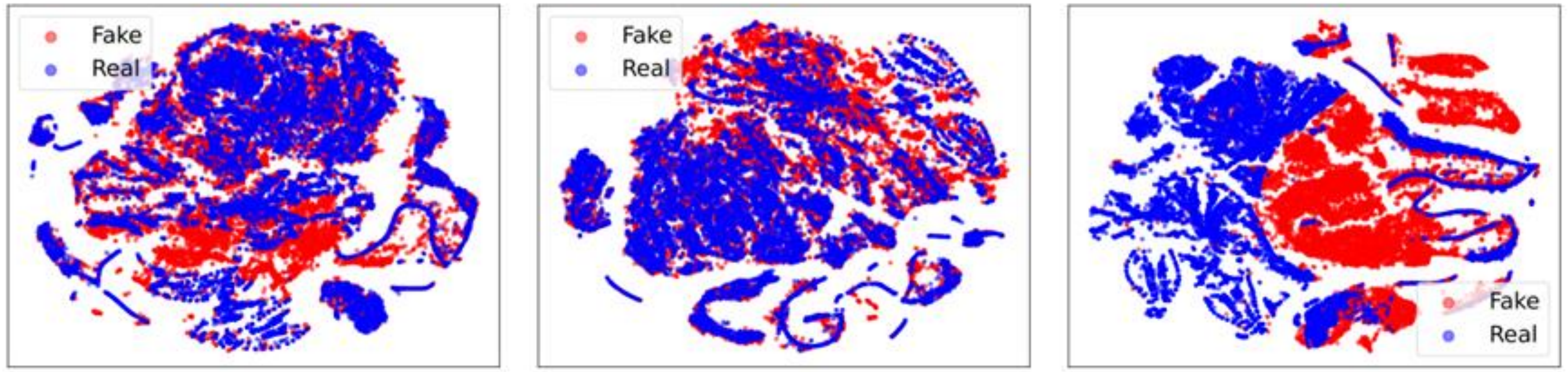
| Model | Adult | Census | Credit | Cabs | King | News |
|---|---|---|---|---|---|---|
| IT-GAN(Q) | 0.612±0.008 | 0.833±0.011 | 0.710±0.012 | 0.659±0.016 | 0.761±0.025 | 0.791±0.003 |
| IT-GAN(L) | **0.599±0.016** | **0.741±0.027** | **0.656±0.027** | **0.630±0.011** | **0.703±0.032** | **0.783±0.010** |
| IT-GAN | 0.618±0.003 | 0.816±0.019 | 0.688±0.058 | 0.654±0.033 | 0.742±0.003 | 0.788±0.007 |

- Table shows privacy attack success scores. High score means being more vulnerable to privacy attack.
- ITGAN(L) achieves the lowest privacy attack sucess scores, however before results prove that ITGAN(L) has reasonable machine learning evaluation scores.

# Experiments

## Ablation Study: ITGAN vs ITGAN(Q,L)




(a) IT-GAN (b) IT-GAN(Q) (c) IT-GAN(L)

- The above figure(figure1) shows real-fake distance and the below figure (figure2) shows t-SNE visualization.
- It shows the effect of log-density
- ITGAN(Q) generates the synthesized data more similar to oringal data than ITGAN.
- However, ITGAN(L)'s samples are very different with original data.
- These show that log-density regularizer works as intended.

# Experiments

## Sensitivity Analyses

Table 9: Sensitivity in News

| $\gamma$ | $R^2$ | Ex.Var | MSE | MAE | Dist. |
|---|---|---|---|---|---|
| -0.0105 | 0.05 | 0.07 | 0.77 | 0.66 | 2.53 |
| -0.0100 | 0.06 | 0.07 | 0.76 | 0.65 | 2.49 |
| 0.0000 | 0.07 | 0.10 | 0.75 | **0.64** | 2.41 |
| 0.0100 | **0.10** | **0.11** | **0.73** | **0.64** | 2.44 |
| 0.0500 | **0.10** | 0.10 | **0.73** | 0.65 | 2.34 |

Table 10: Sensitivity of full black box attack w.r.t. $\gamma$ in News

| $\gamma$ | FBB ROCAUC |
|---|---|
| -0.012 | 0.762 |
| -0.011 | **0.752** |
| 0.000 | 0.784 |
| 0.050 | 0.787 |
| 0.100 | 0.792 |

- $\gamma$ is the coefficient of negative log-density regularizer
- When $\gamma = 0.01$, the evaluation scores(Machine Learning score) are the best.
- As $\gamma$ decreases, real-fake distance (Dist.) increases.
- Similarly the lower $\gamma$ achieves the lower attack success score.

# Conclusion

## ITGAN

- Successfully combine GAN and log-density regularization
- **ITGAN can adjust trade-off between synthesis quality and the real-fake distance.**
- In some Multi-class or/and imbalanced datasets, there is a room for improving.

# Thank You !