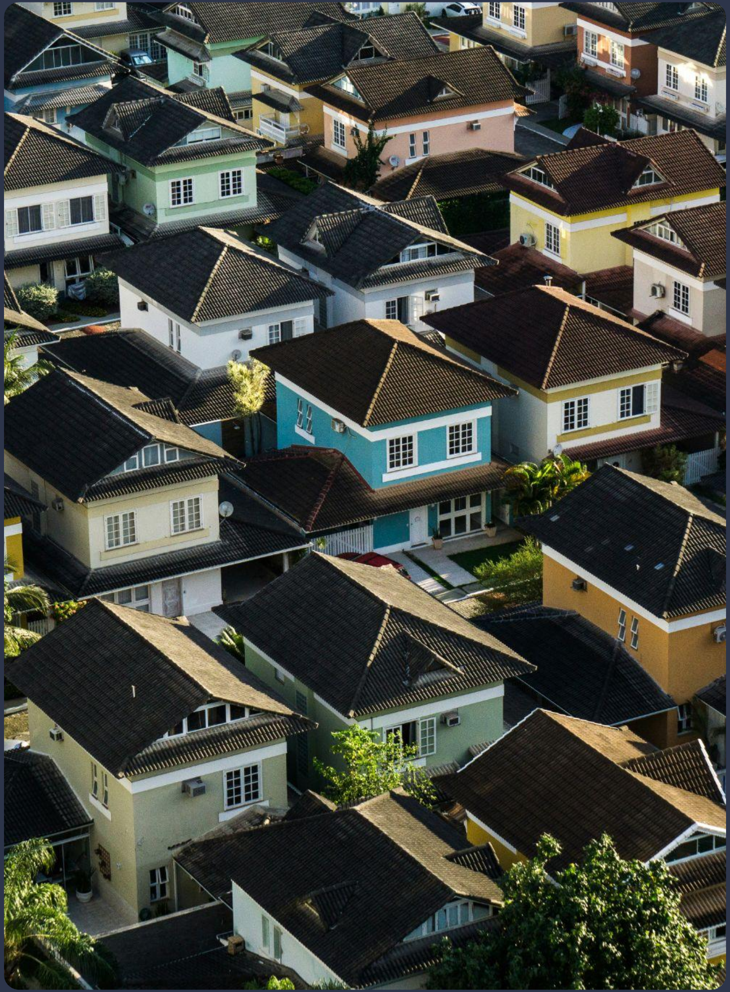
LF ENERGY

OpenSynth

Workshop: How to Train a Generative AI Model to Create Synthetic Smart Meter Data

Gus Chadney & Sheng Chai, Centre for Net Zero



Centre for Net Zero

Powered by **Octopus Energy Group**

Today’s Agenda

1. Training a generative model
2. Evaluating Privacy
3. Evaluating Fidelity and Utility

Faraday Paper

‘Defining Good’ Paper

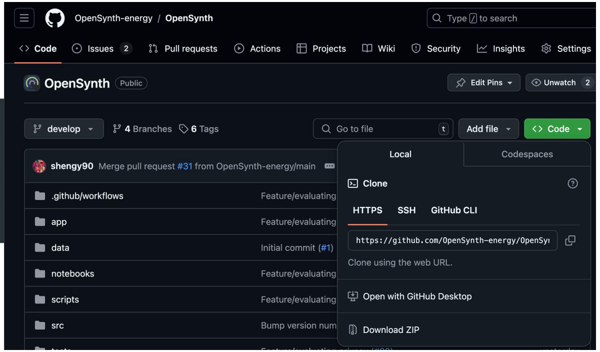
Centre for Net Zero
Powered by Octopus Energy Group

Setting up your environment

🅰️ Clone the OpenSynth repository: https://github.com/OpenSynth-energy/OpenSynth/

🅱️ Install opensynth-energy with pip

```
pip install opensynth-energy
```



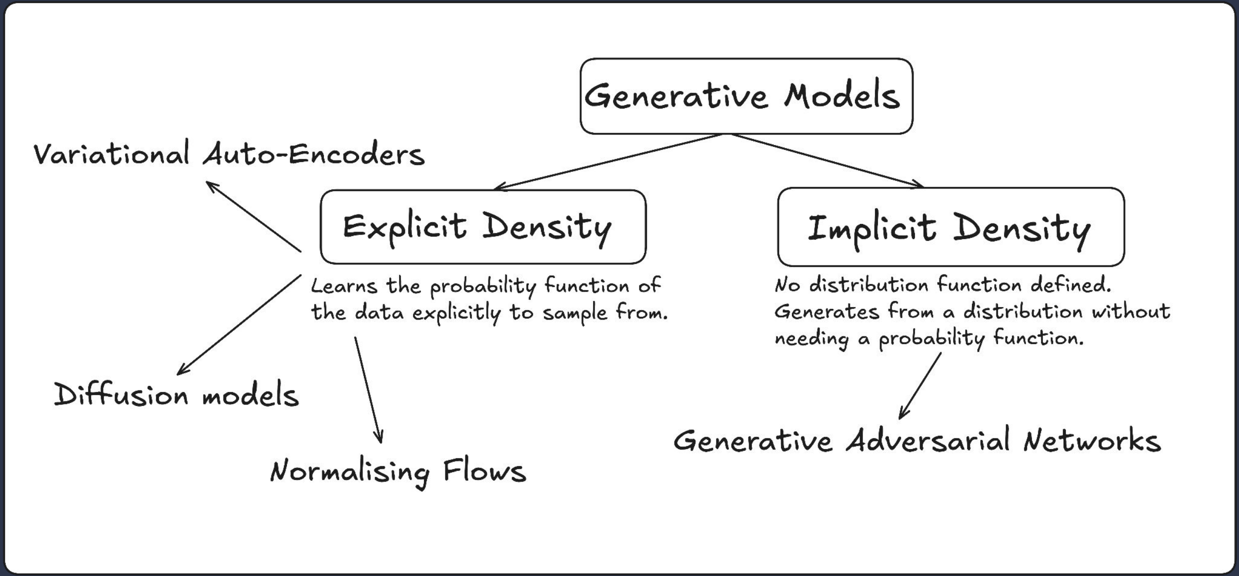
Pre-requisites

1. Python version >= 3.11
2. Numpy version < 2.0.0
3. Pytorch lightning >= 2.3.1
4. Pipenv for managing dependencies

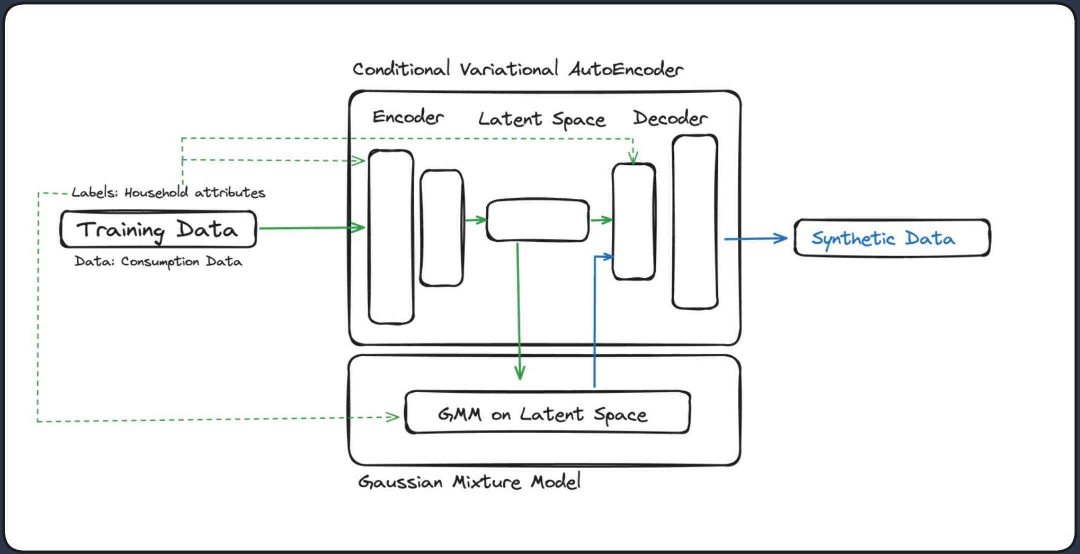
Installing Pipenv + Repo

1. Pip install pipenv
2. Pipenv install torch">=2.3.1"
3. Pipenv sync --dev

1 Training a generative model



Faraday - VAE + GMM



Faraday Paper

Encoder

Maps raw data into a latent space.

GMM

Learns distribution of the latent space.

Decoder

During inference, samples from GMM and decode with decoder.

Centre for Net Zero
Powered by Octopus Energy Group

Lab session: Training Faraday

💾 **Low Carbon London Dataset**

https://shorturl.at/ahe3p

📒 **Notebooks**

Github repo/notebooks/faraday/faraday_tutorial.ipynb

If you need GPU

1. Switch to Google-Colab-Notebooks branch

2. Open notebooks/faraday/Training_Generative_Model.ipynb in Google Colab

3. Download trained model artefacts to your computer!

If you have GPU

Check that Pytorch can access your GPU.

```
import torch

# If on windows (Nvidia GPUs)
torch.cuda.is_available()

# If on Mac machines
torch.backends.mps.is_available()
```

2 Evaluating Privacy

Privacy: Risk of private data leaking, i.e. original training data being recovered from generated samples.

Implicit Protection

During training, limit the model's ability to learn from data to prevent private data from being "memorised".

Explicit Attacks

After training, prove that generated samples are indistinguishable **from unseen training data** to give **"plausible deniability"**

2 Differential Privacy

Inclusion or exclusion of any single data record should not have "disproportionate" impact on the model's output:

- Two datasets, $\mathbf{D_1}$ and $\mathbf{D_2}$ that differs by only 1 row
- **A** is any algorithm, e.g. a 'mean' → A($\mathbf{D_1}$) = Mean of $\mathbf{D_1}$
- **S** is all possible values of $\mathbf{D_1}$
- To be considered **(ϵ, δ)-differentially private:**

$$\Pr(A(D_1) \in S) \leq e^{\epsilon} \Pr(A(D_2) \in S) + \delta.$$

Smaller ϵ = more private.

δ typically set to < 1/N where N is dataset size.

'Defining Good' Paper

2 Differential Private Stochastic Gradient Descent

- **ML Algorithm learns through SGD**

Stochastic gradient descent algorithm: updating model weights iteratively using randomly selected subsets (batches) of the training data to minimize a loss function.

- **DP-SGD modifies SGD to make process differentially private**

DP-SGD introduces **1) gradient clipping** to clip the largest gradient in a batch, typically set norm value to 1. and **2) injecting noise** to 'mask' the largest gradient in a batch and **3) (privacy-accounting)** keep track of how much noise was added.

In PyTorch, commonly implemented using the Opacus library.

Smaller ϵ = more private.

δ typically set to < 1/N where N is dataset size.

'Defining Good' Paper

2 Explicit Privacy Attacks

Membership Inference Attack (MIA)

Distinguish between synthetic data vs unseen holdout data.

Reconstruction attack

Force a generative model to produce output that resembles closely to original training data.

Attribute Inference Attack

Reveal sensitive information from synthetic data.

2 Membership Inference Attack

Consider the scenario. You have:

- **Model A:** predict whether someone has a criminal record
- **Model B:** predict whether someone belongs to the training data of Model A

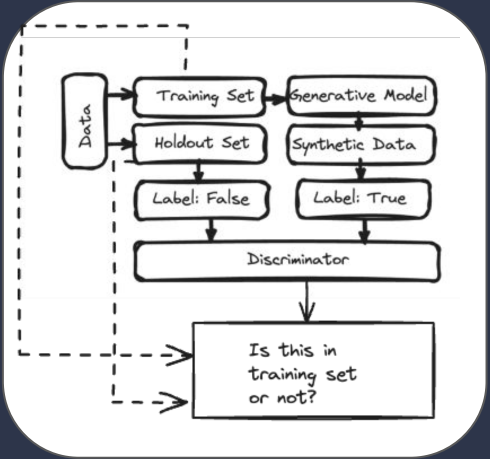
If **Model B** predicts **Person A** belong to **Model B** with high confidence, and **Model A** predicts **Person A** has criminal records with high confidence, then we can have high confidence that **Person A** has criminal record → Leaking the privacy of **Person A**.

Distinguish between synthetic data vs unseen holdout data.

'Defining Good' Paper

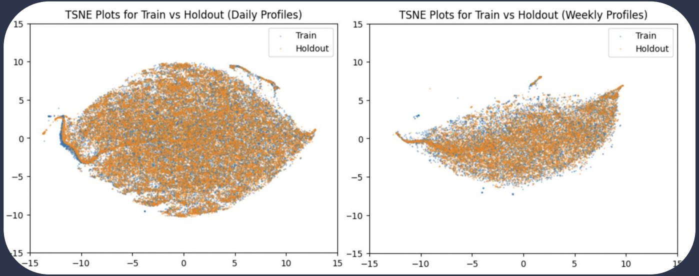
② Membership Inference Attack

1. **Split training data into Train and Holdout.** Train is used to train generative model. Holdout is used for privacy analysis.
2. **Generate a sample of synthetic data.** This data is labelled true. Holdout data is labelled false.
3. **Train a discriminator to distinguish between synthetic vs holdout data.**



Distinguish between synthetic data vs unseen holdout data.

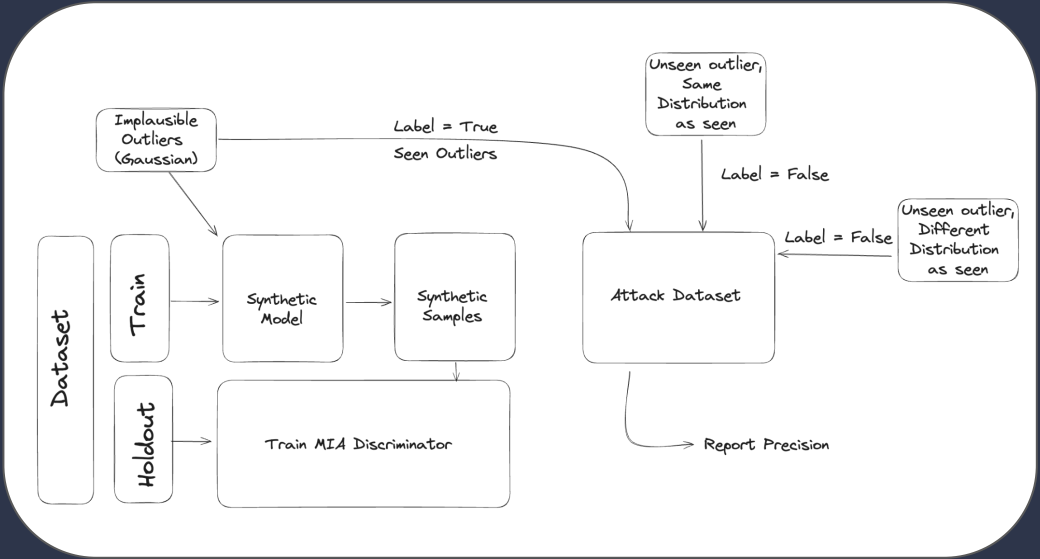
Holdout and Training data distribution however is too similar for traditional MIA to work!



Centre for Net Zero
Powered by Octopus Energy Group

'Defining Good' Paper

2 Membership Inference Attack with Outliers



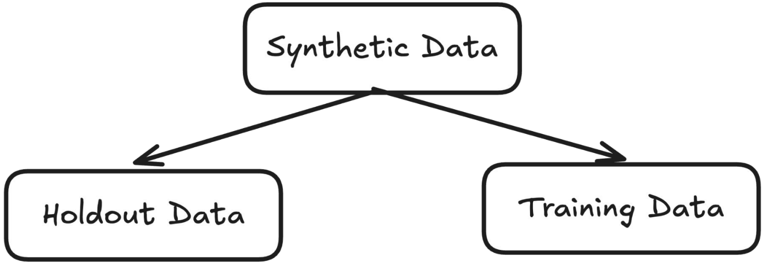
Inject implausible outliers to training data for generative model. Then launch MIA directly on injected outliers.

Distinguish between synthetic data vs unseen holdout data.

Centre for Net Zero
Powered by **Octopus Energy Group**

'Defining Good' Paper

2 Reconstruction Attack



One-tailed Two-sample KS Test:

Null Hypothesis: Synthetic Data is closer to Holdout than to training data.

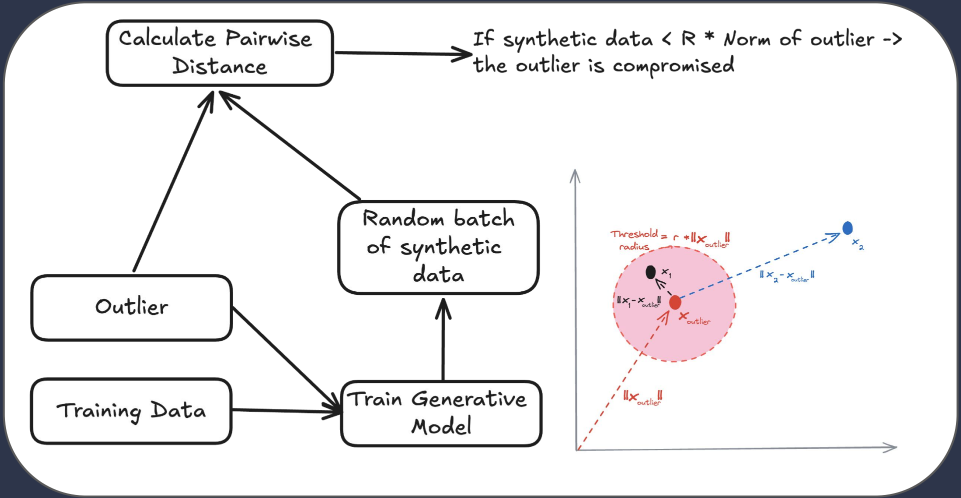
Alt Hypothesis: SYnthetic Data is not closer to Holdout than to training data.

Distance-based attack to prove that synthetic data is not more similar to training than to holdout data.

Force a generative model to produce output that resembles closely to original training data.

‘Defining Good’ Paper

2 Reconstruction Attack with Outliers

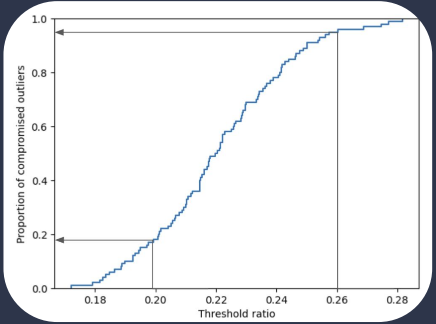


Stakeholders decide on a threshold radius.

Force a generative model to produce output that resembles closely to original training data.

Calculate % of outliers that were compromised under the decided threshold.

Ratio of 0.3 ~ Vector norm with 30 kWh: outlier with norm of 30±9 kWh would compromise it.



Centre for Net Zero
Powered by Octopus Energy Group

'Defining Good' Paper

Lab session: Evaluating Privacy

📒 **Notebooks**

Github repo/notebooks/faraday/faraday_evaluation.ipynb

‘Defining Good’ Paper

You won’t need GPU for this.

Repo won’t work on Google Colab because of python compatibility.

Google colab uses 3.10.
OpenSynth requires 3.11.

If you have GPU

Check that Pytorch can access your GPU.

```python
import torch

# If on windows (Nvidia GPUs)
torch.cuda.is_available()

# If on Mac machines
torch.backends.mps.is_available()
```

3 Evaluating Fidelity and Utility

Fidelity

How similar synthetic data is statistically to original training data. Evaluated with statistical metrics, e.g. mean, quantiles, autocorrelation etc.

Utility

Would we still get similar results if we used synthetic data in place of real data for downstream applications?

‘Defining Good’ Paper

3 Proposed Fidelity Metrics

1. Standard statistical and distance measures: Calculate the mean and 95th of each load profile. Compare the distributions of mean/ quantile values between Real and Synthetic data.

2. Distribution of autocorrelation function (ACF) : Calculate the ACF coefficients of each load profile. Compare the distributions of ACF coefficients between Real and Synthetic data.

3. Distribution of magnitude and timing of peaks : Mask each load profile with 0 and preserving only the 'Top K' values. Compare the distribution of masked dataset between Real and Synthetic data.

4. Distribution of clusters: Fit N arbitrary clusters using Gaussian Mixture Model on real data to predict clusters, and obtained predicted clusters of synthetic data. Compare distribution of clusters between Real and Synthetic Data.

Smart meter data is hierarchical. Repeat 1-3 on a cluster level using clusters from 4.

'Defining Good' Paper

Centre for Net Zero
Powered by Octopus Energy Group

3 Proposed Utility Metrics

Train on Synthetic, Test on Real Framework. For a downstream task:

1. Train Model A on Real data
2. Train Model B on Synthetic Data
3. Compare performances of Model A and Model B on unseen real data.

Classification task

Predict season using smart meter data.

Forecasting task

Predict next-time step consumption using historical data.

'Defining Good' Paper

Centre for Net Zero

Powered by Octopus Energy Group

Lab session: Evaluating Fidelity

📒 **Implement your own fidelity and utility metrics!**

💾 **Prepared Dataset**

https://drive.google.com/drive/u/0/folders/1RJV-OFgelPwWVYkTK6YPQ2TSwu4LVreN

‘Defining Good’ Paper

If you need GPU

1. Create a new google colab notebook

2. Download the prepared dataset for fidelity/ utility analysis, upload it manually to Google Colab and have a go!

If you have GPU

Check that Pytorch can access your GPU.

```python
import torch

# If on windows (Nvidia GPUs)
torch.cuda.is_available()

# If on Mac machines
torch.backends.mps.is_available()
```