Semi-Supervised Learning with GANs for Device-Free Fingerprinting Indoor Localization

Kevin M. Chen and Ronald Y. Chang

Presenter: Kevin M. Chen

Research Center for Information Technology Innovation (CITI), Academia Sinica, Taipei, Taiwan

Presentation Outline

I. Introduction

A. Indoor Localization Problem

B. Our Motivation

II. Proposed Method

A. Intro. to Generative Adversarial Networks (GANs)

B. Semi-Supervised with GANs

III. Experiment & Discussion

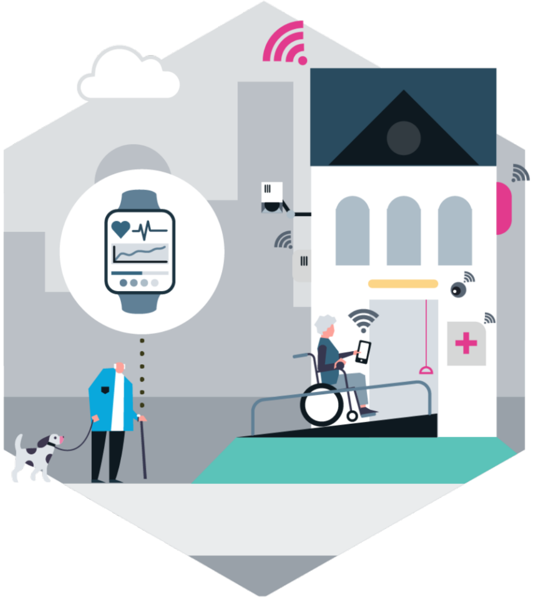
A. Performance Comparison

B. Discussion on Generator model

IV. Conclusion

Indoor Localization Problem

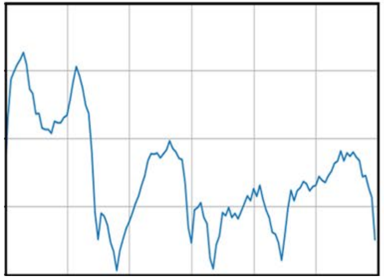
Many current and future **Internet of Things (IoT)** applications are enabled or facilitated by indoor location information.



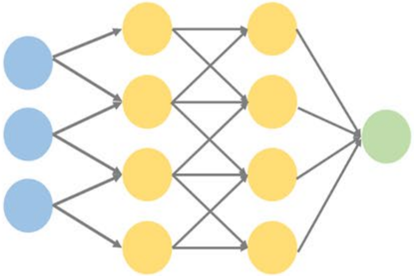
Indoor Localization Problem (cont.)

Solution =

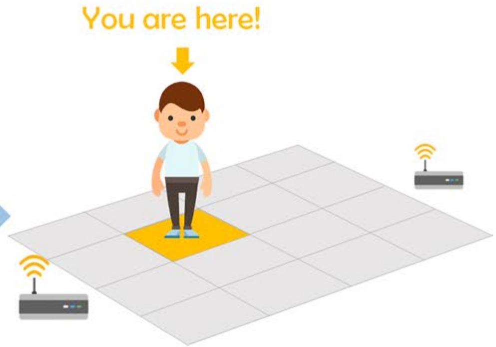
Collect Indoor Wireless Signal (CSI) + **Fingerprint (Deep Learning)**



Wireless Signal Measurement:
Channel State Information (CSI)



Deep Learning:
Neural Network (NN) Model



Online Testing

Motivation

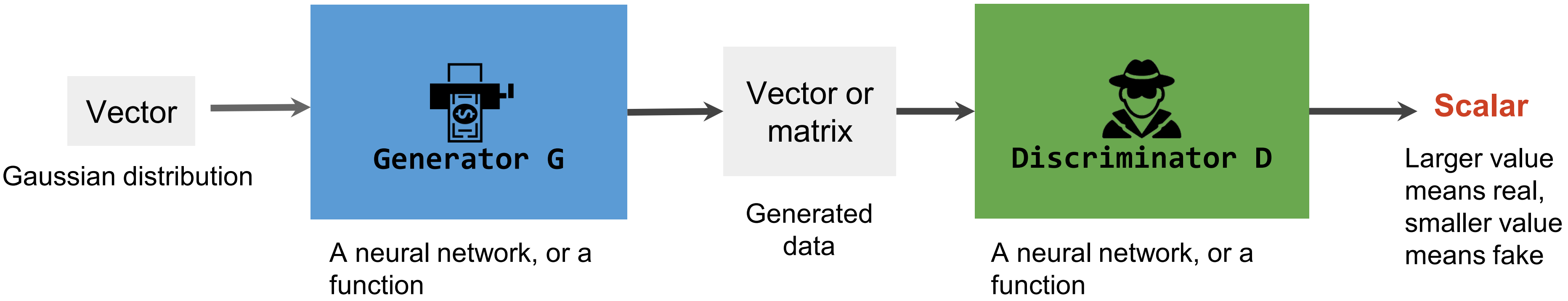
Many proposed deep learning-based solutions are based on **supervised learning**, which requires **numerous labeled data** collected in the site survey to train the fingerprinting localization system. However, data labeling is **labor-intensive** and **time-consuming**.

Motivation (cont.)

- **<u>Semi-Supervised Learning</u>,** which uses a small amount of labeled data and a large amount of unlabeled data for training, gives a possible solution to reduce labor effort.
- **<u>Generative Adversarial Network (GAN)</u>** has been proposed for Semi-Supervised Learning.

Generative Adversarial Network (GAN)

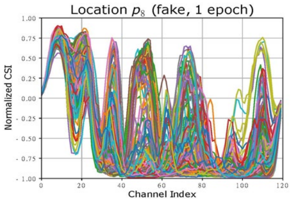
Basic idea of GAN:



Training a GAN is a ***minimax optimization***.

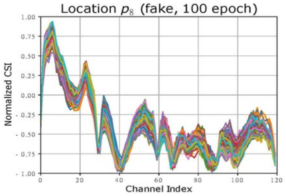
Semi-Supervised with GANs

If we want a GAN to ***generate fake CSI samples***...



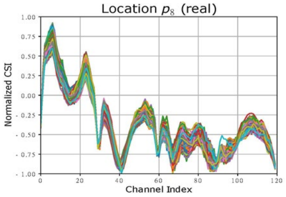
Fake CSI (1 epoch)

Training

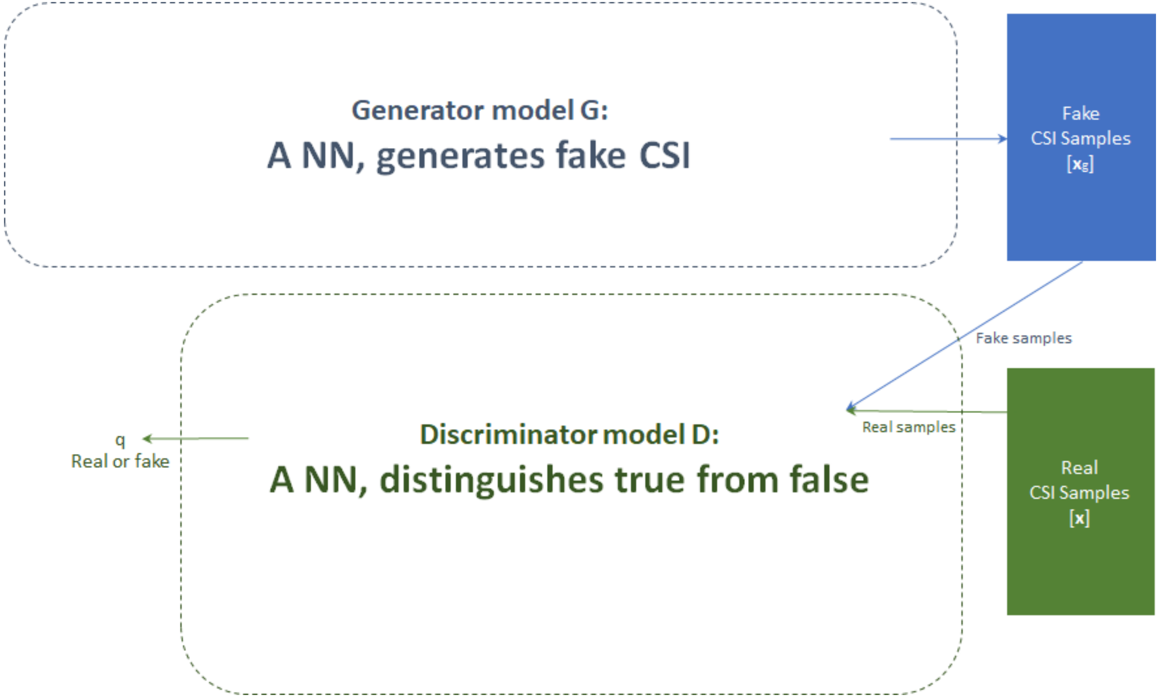


Fake CSI (100 epoch)

Resemblance



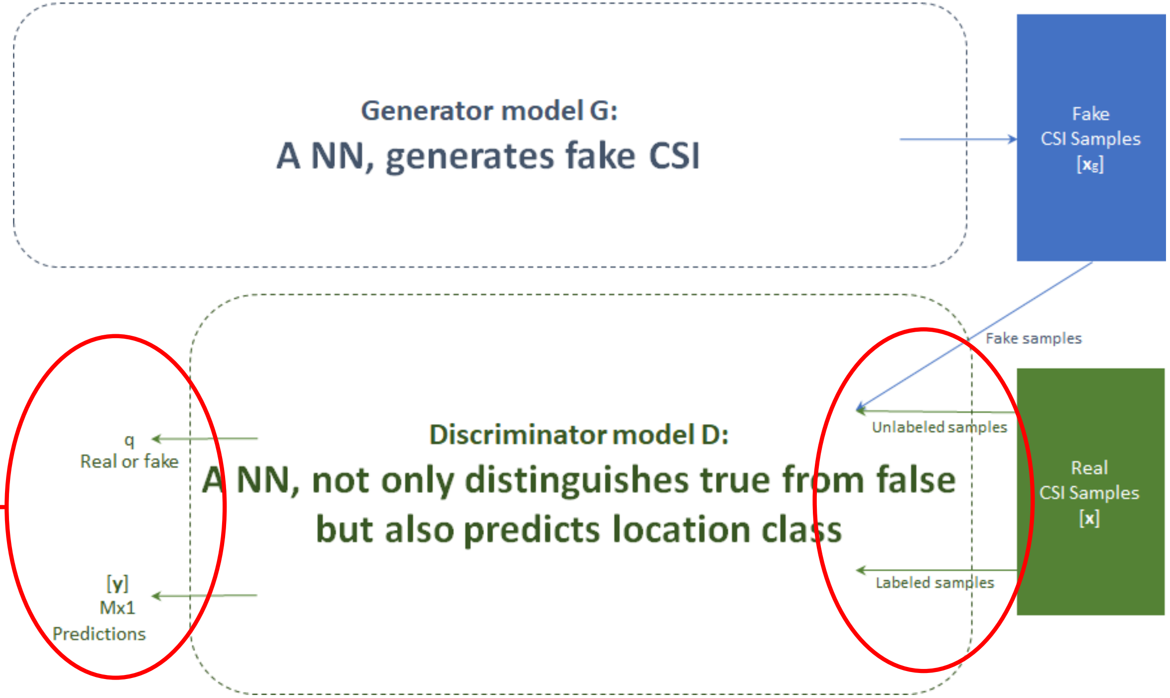
Real CSI



Semi-Supervised with GANs (cont.)

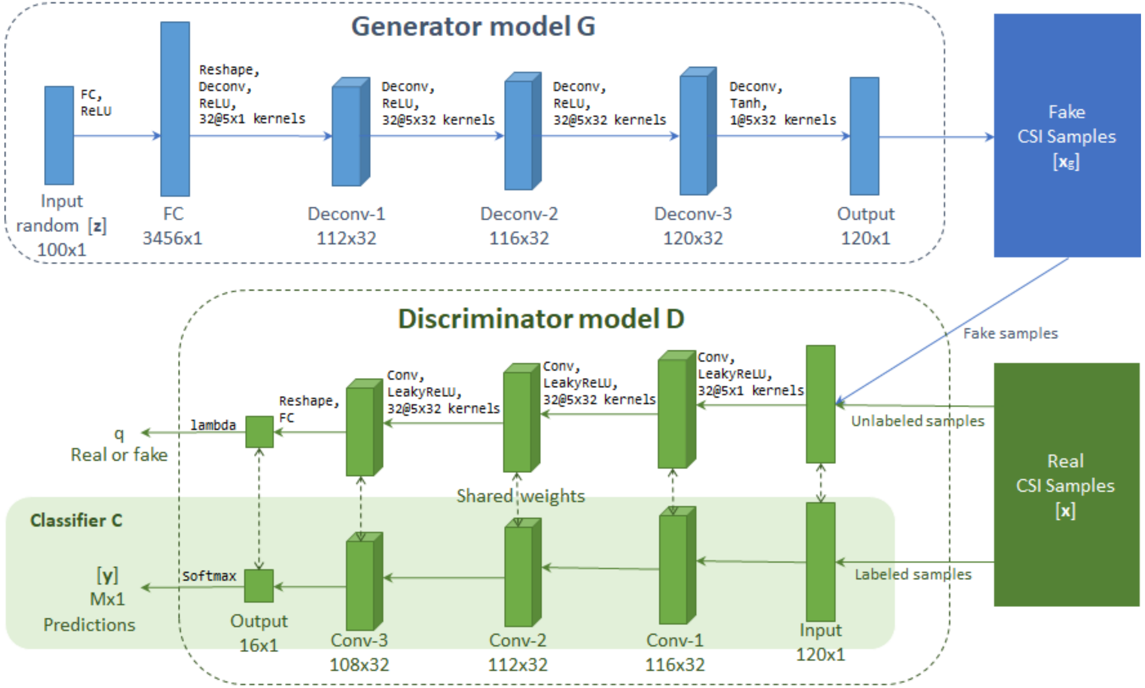
If we want a GAN to do ***semi-supervised learning with CSI samples***...

Make standard GAN's output from <u>1 to M+1</u> by making D become a dual model with shared weights.



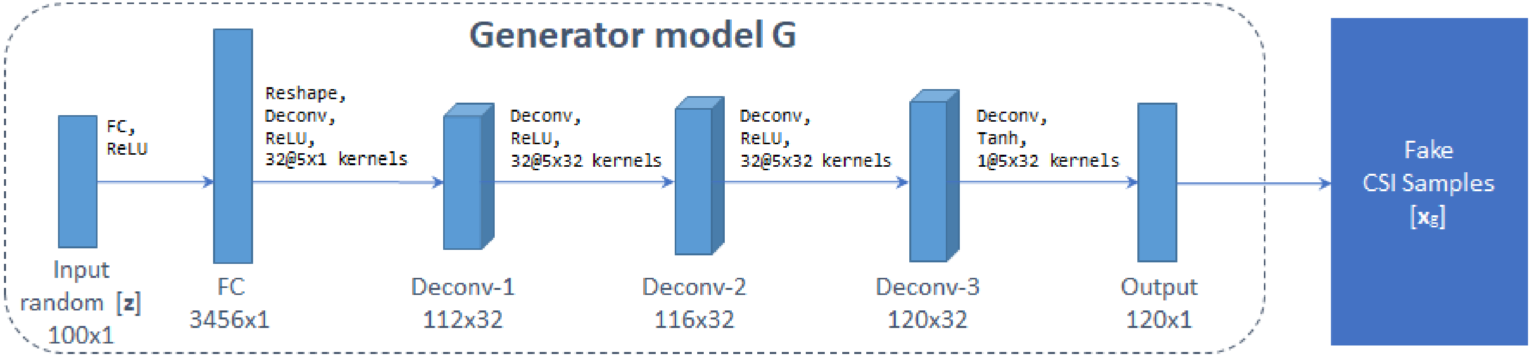
Semi-Supervised with GANs (cont.)

We apply **Deep Convolutional Generative Adversarial Network** (DCGAN) as our model's architecture.



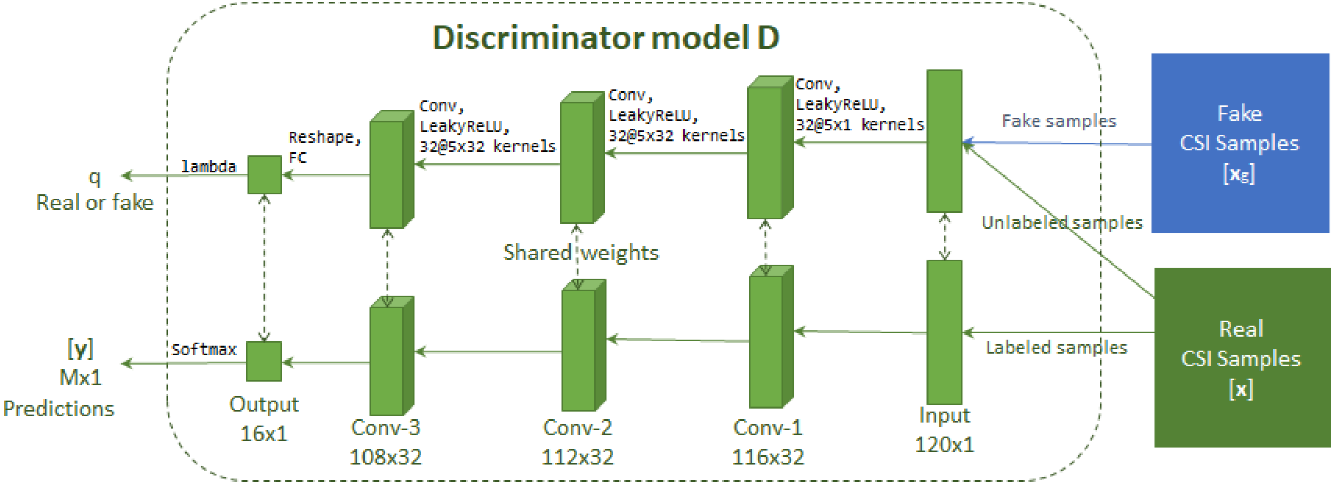
Note: The parameters of the model is determined by cross-validating the model performance with different configurations.

Generator Model



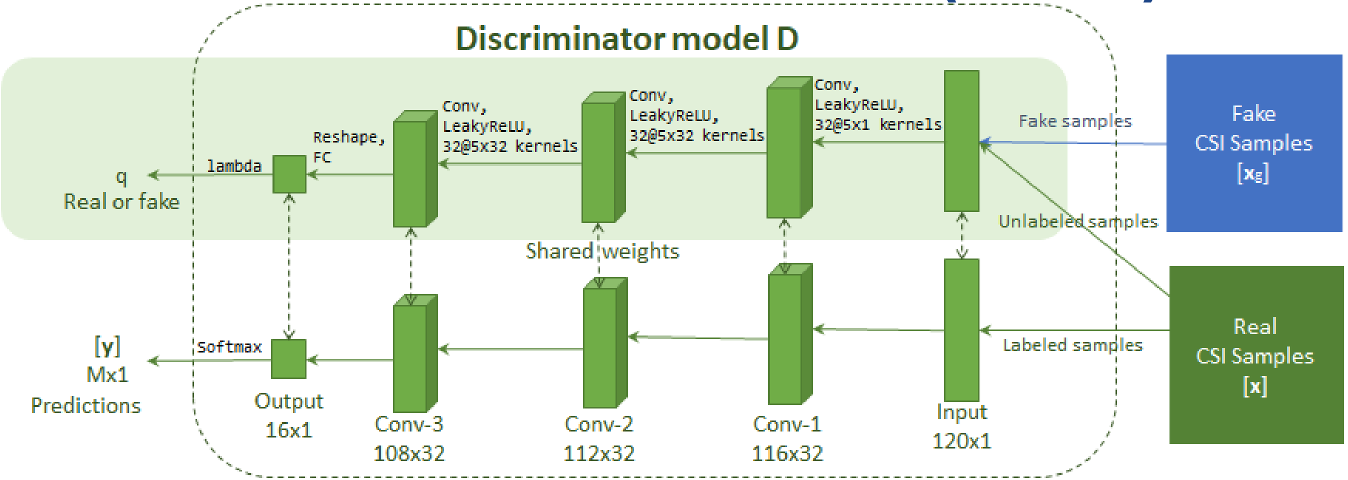
- **Input:** Random Gaussian distribution noise **z** (size 100x1)
- **Output:** Fake Generated CSI sample $\mathbf{x}_g$ (size 120x1)

Discriminator Model



- **Input:** Generated fake sample $\mathbf{x_g}$ (size 120x1); Real CSI sample **x** (size 120x1)
- **Output:** q, possibility the input is real (scalar); **y**, predictions on input (size Mx1)

Discriminator Model (cont.)

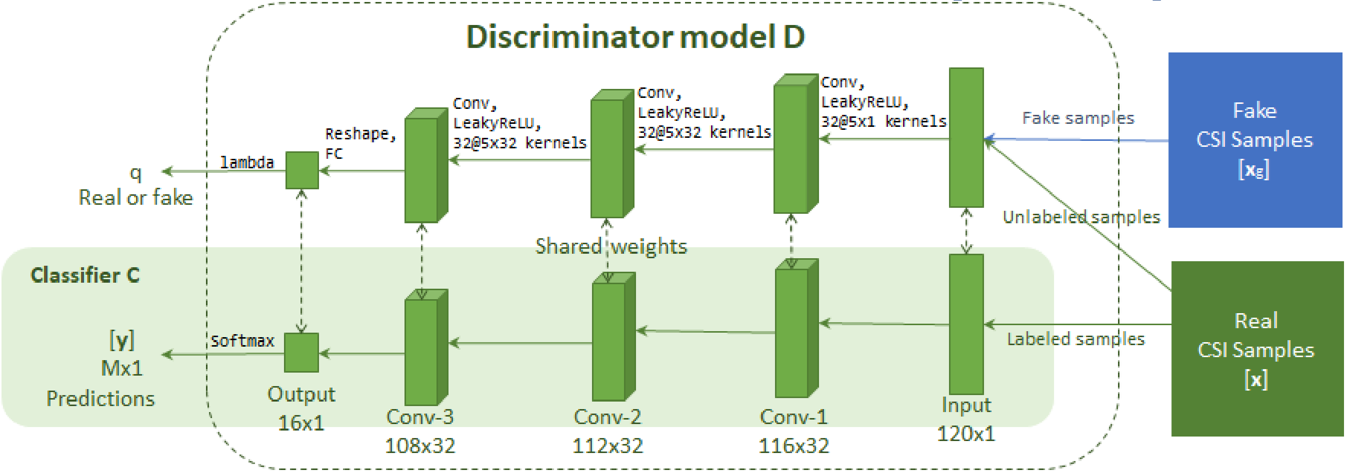


- **Seen as *Discriminator D***: Produce a scalar q which represents the probability of the input CSI sample being a real sample with a customized lambda function:

$$\lambda(\mathbf{c}) = \frac{\sum_{m=1}^{M} \exp(c_m)}{\sum_{m=1}^{M} \exp(c_m)+1}$$

[T. Salimans et al, "Improved techniques for training GANs, NIPS2016]

Discriminator Model (cont.)

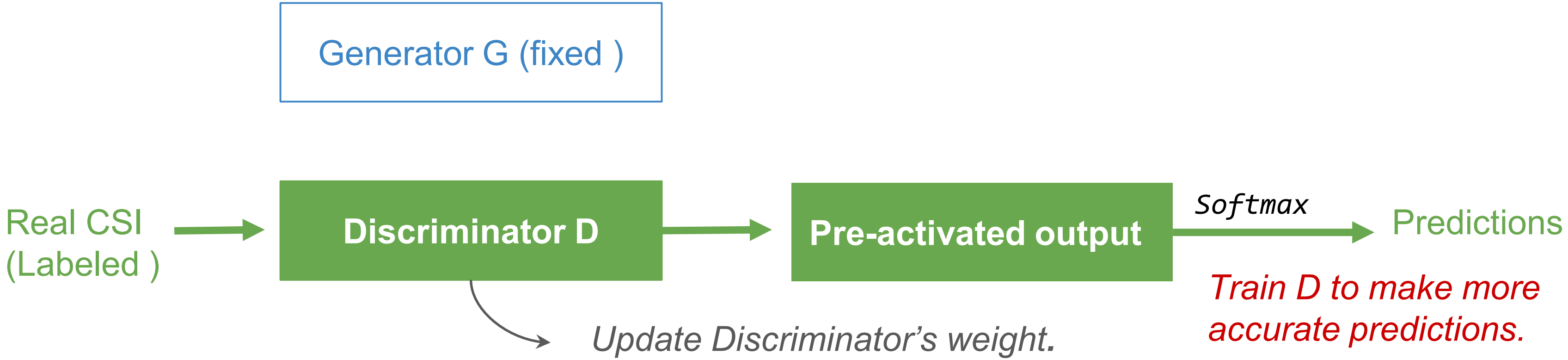


- **Seen as *Classifier C***: Produce a prediction vector **y** with Softmax activation and index of the largest component in **y** is the class prediction.
- ***<u>This model C is saved as the fingerprint</u>***.

Training DCGAN

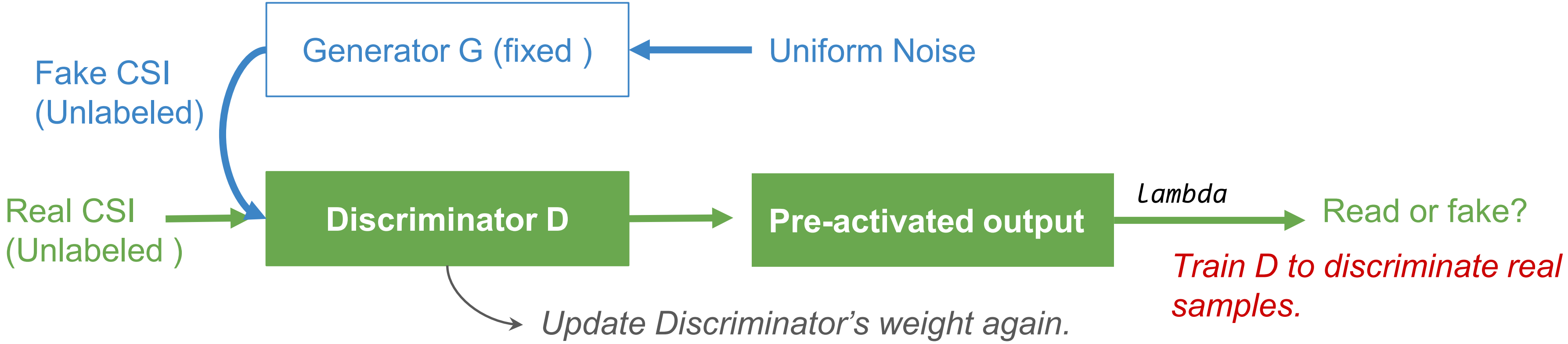
The training of the proposed DCGAN involves a **two-step** iterative process: (1)Training ***Discriminator*** and (2)Training ***Generator***. For each iteration:

First, we train Discriminator *as Classifier*, with ***few*** labeled CSI



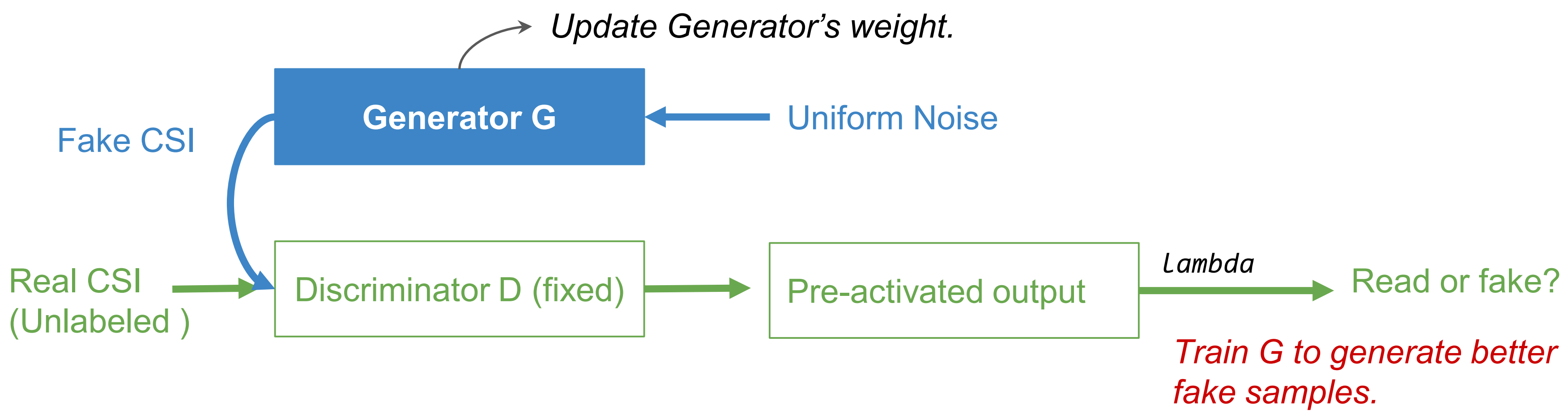
Training DCGAN (cont.)

Next, we train Discriminator D again, but with the fake CSI samples from ***fixed G*** plus real CSI samples:



Training DCGAN (cont.)

Finally, we turn to train Generator G, with the ***fixed and updated Discriminator D***:

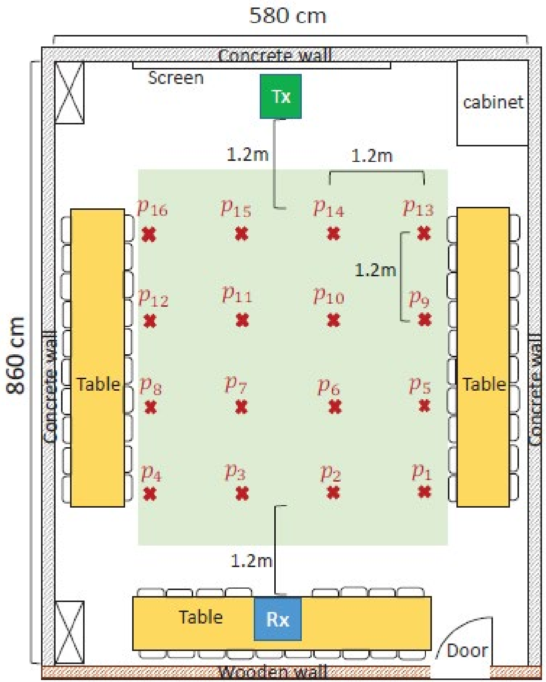


Note: The model is trained with Adam optimizer and categorical cross-entropy loss function in Tensorflow.

Experimental Environment

Dataset:

- **Training Set**: 400 CSI samples for each location (6400 for all locations)
- **Testing Set**: 200 CSI samples for each location (3400 for all locations)
- Unlabeled Set: the CSI is same as training set but without label



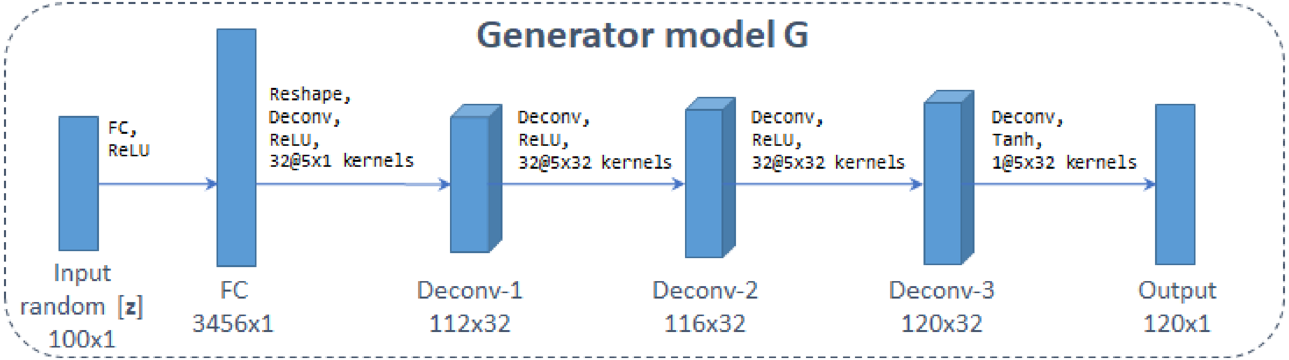
Conference room at the Research Center for Information Technology Innovation, Academia Sinica.

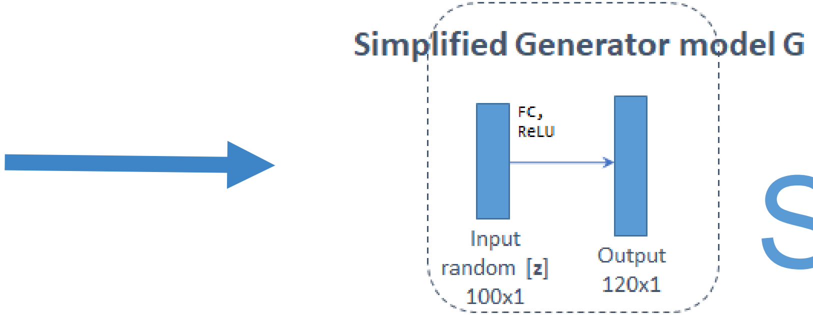
Experiment Result

Labeled CSI samples	Semi-Supervised DCGAN (%)	Supervised CNN (%) (same architecture as D of DCGAN)
16	**85.75**	**58.87**
32	85.78	68.78
64	87.28	82.47
128	87.41	81.25
1600	87.09	86.87
3200	86.72	88.31
6400	**87.84**	**87.71**

With semi-supervised DCGAN, the classification accuracy retains with reduced amount of labeled data.

Simplified G





Still Work ?

Simplified G (cont.)

Labeled CSI samples	Semi-Supervised DCGAN (%)	Semi-Supervised DCGAN With a Simplified G (%)
16	**85.75**	**64.40**
32	85.78	72.94
64	87.28	79.25
128	87.41	79.41
1600	87.09	81.41
3200	86.72	86.63
6400	**87.84**	**87.06**

A sophisticated G can help train a good D, and consequently a good C, in the considered DCGAN architecture.

Conclusion

1. We have presented a **GAN-based semi-supervised approach** to the device-free fingerprinting indoor localization problem.
2. We showed that the proposed scheme **achieves an increasingly advantageous performance when trained with an increasingly reduced number of labeled training samples**, as compared to the supervised approach.
3. The **interactions between the G, D, and C** of the proposed model were discussed.

Thank you for your attention!