SAMSUNG

ITU AI/ML in 5G challenge: **ITU-ML5G-PS-004**

Presentation Title : Depth Map Estimation in 6G mmWave systems

Team Name: SixG_ISAC

Team Github Repo: ITU-AI-ML-in-5G-Challenge/**ML5G-PS-004-Depth-map-estimation-in-6G-mmWave-systems**

Team Members:

- Shubham Khunteta
- Yeswanth Reddy Guddeti
- Avani Agrawal
- Ashok Kumar Reddy Chavva

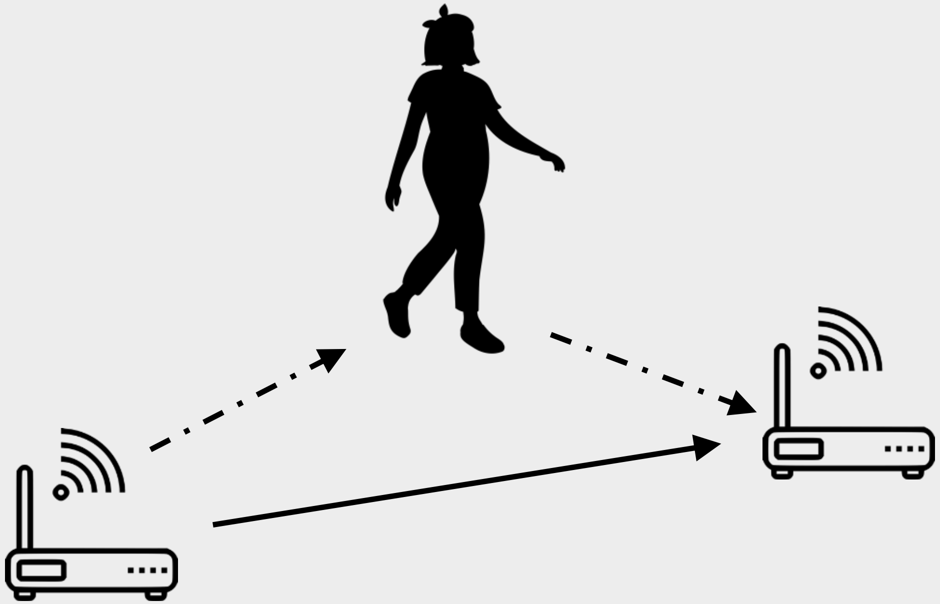
Affiliation *: Beyond 5G Team, Samsung R&D Institute India, Bengaluru.*

Problem Organizer*: NIST, USA.*

Motivation : Communication waves to create Digital world

SAMSUNG

Integrated Sensing & communications



Inference using RF communication signal

Physical world ***Digital world***

Reuse **spectrum, devices and protocols to perform both communication and sensing.**

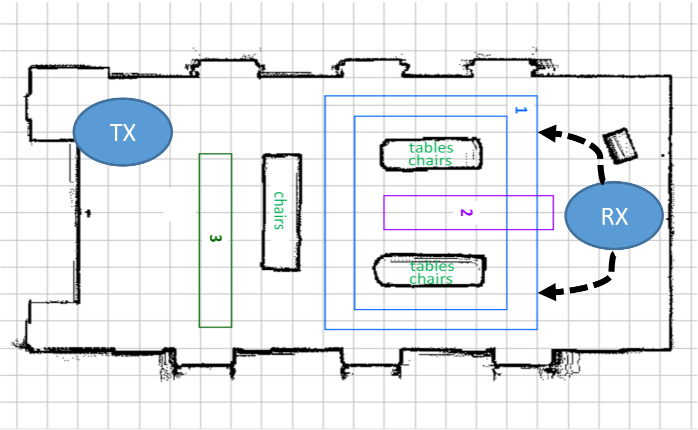
Extra processing required to acquire multi-dimensional data

Problem statement

SAMSUNG

- **Setup :**

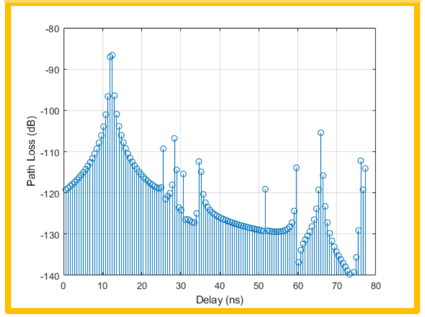
One fixed transmitter



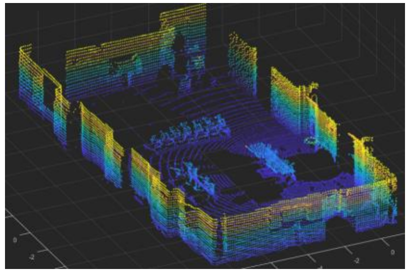
One moving receiver

- **ML Model :**

Channel Impulse response



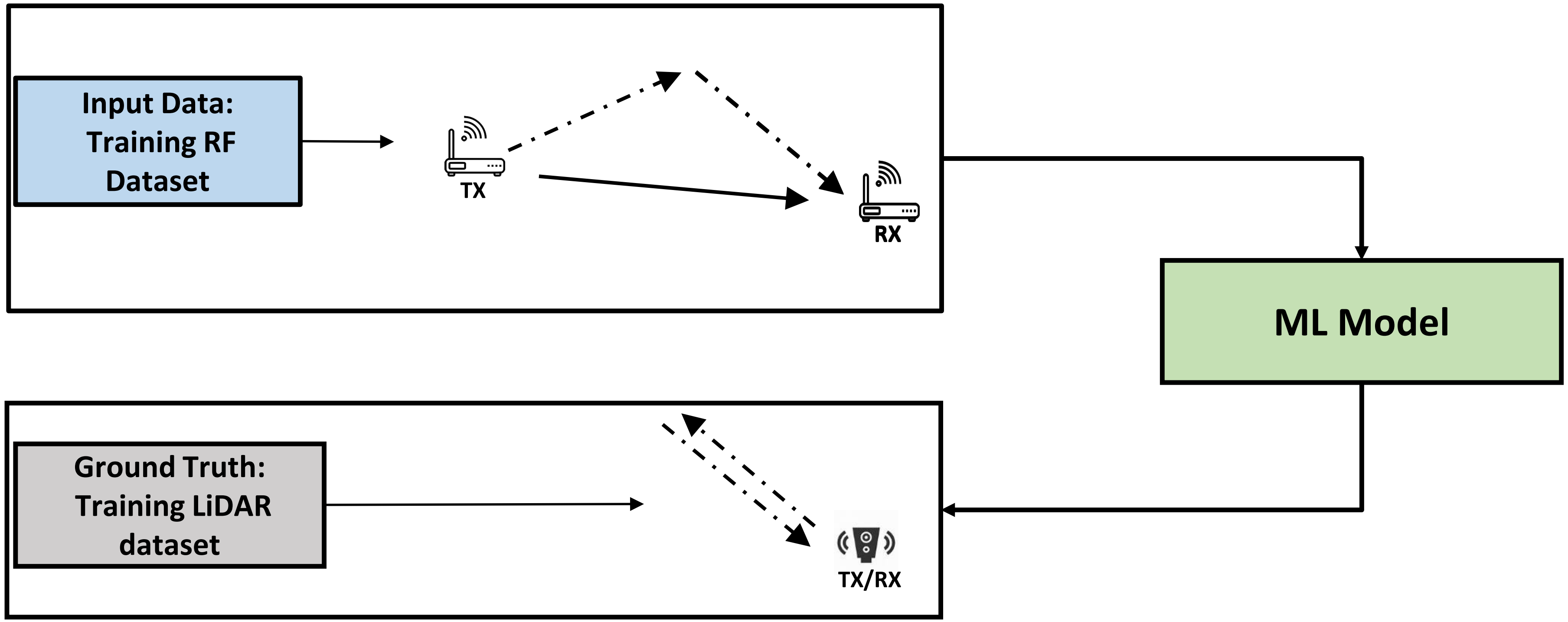
TX/RX
Location & Orientation
Unknown



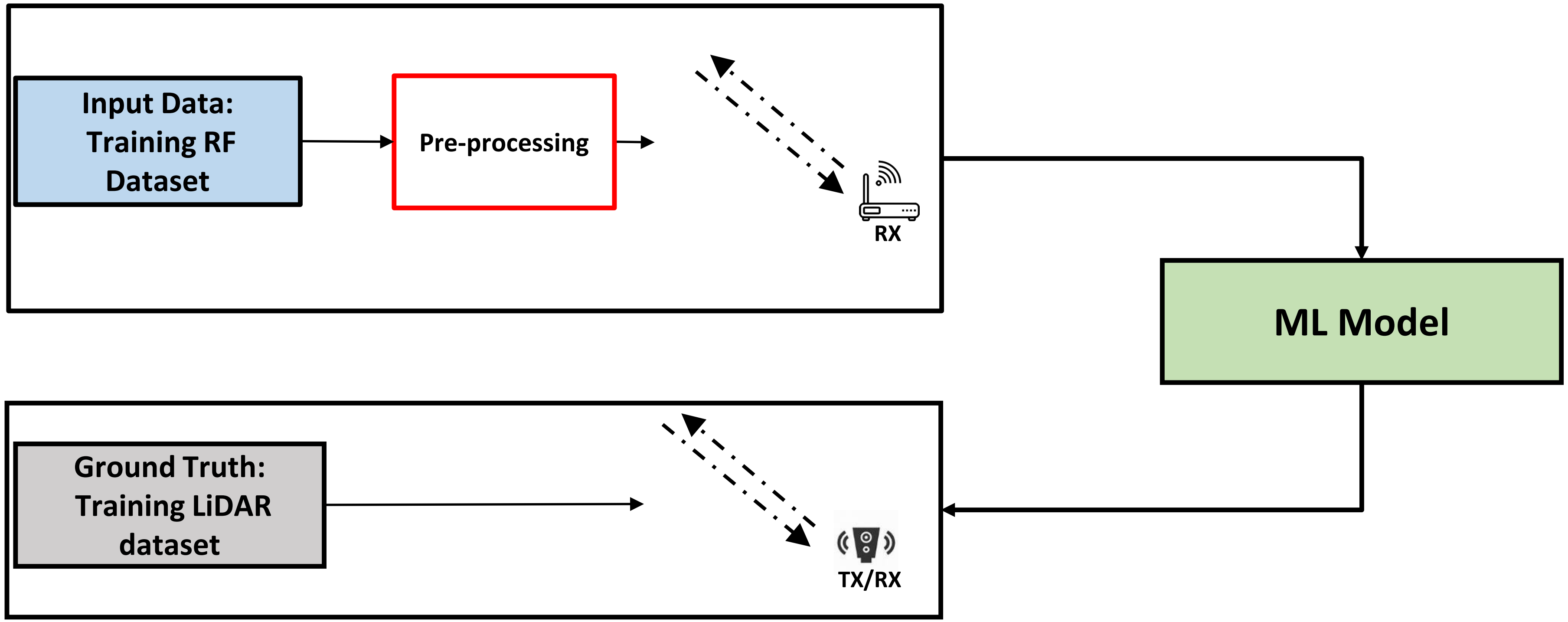
Input Data: Training RF Dataset → **Trained Model** → **Output: LiDAR Point Clouds**

Challenge : Estimate the depth map of the environment at each receiver position, using mm-wave signals + ML.

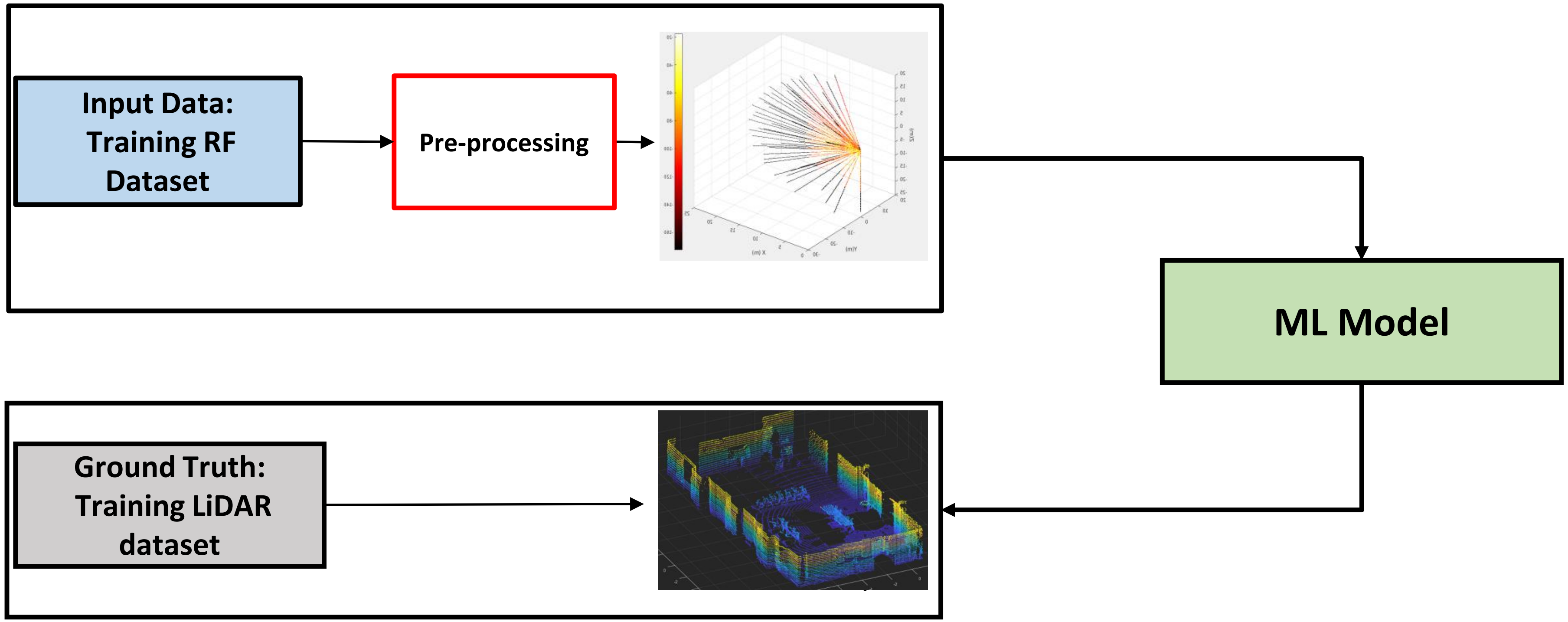
Solution outline



Solution outline



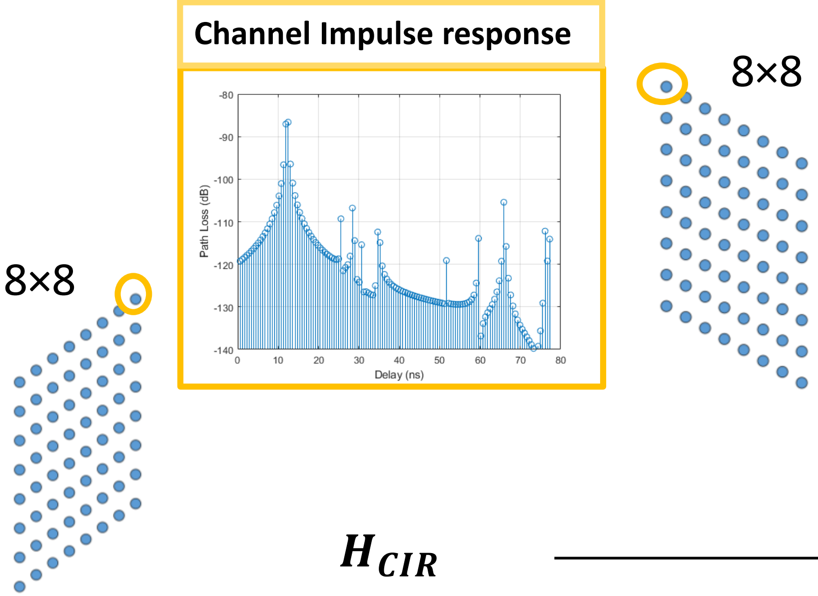
Solution outline

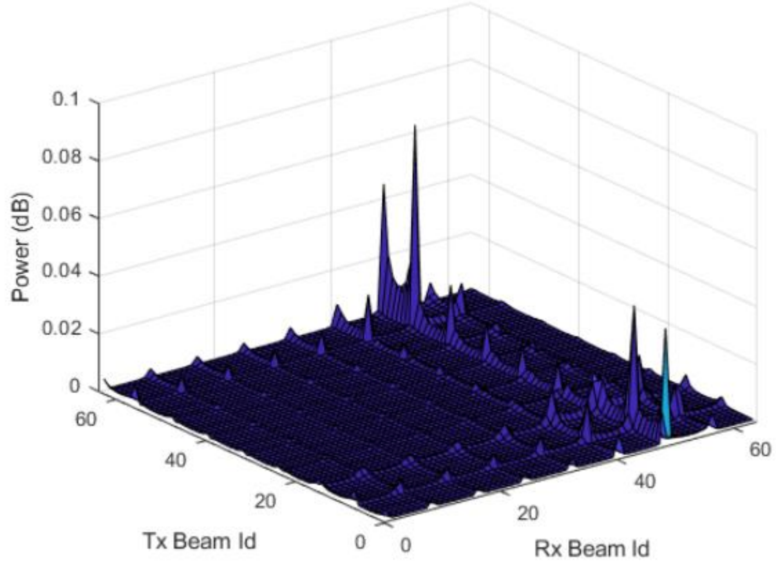


Solution : At each RX location

- Transform bi-static RF data to mono-static format.
- Train ML model by fitting it to similarly structured LiDAR ground truth.

Pre-processing : Spatial Transform





H_{CIR} (TX Ant x RX Ant x Delay) → Spatial transform → H_{beam} (AOD x AOA x Delay)

Delay T_s $\frac{1}{1.76}ns$ ⇔ Distance resolution of $0.17\ m$.

Horz.Ant 8 ⇔ Azimuth resolution of 22.5 º (Avg.)

Vert.Ant 8 ⇔ Elevation resolution of 22.5 º (Avg.)

Transforms Communication data to Sensing framework

Pre-processing : Spatial Transform

H_{CIR}

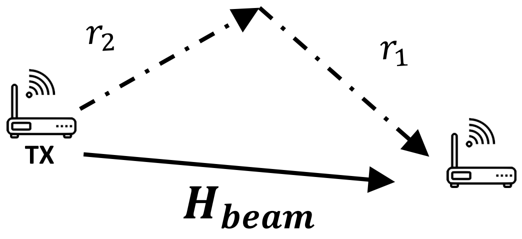
↓

Spatial transform

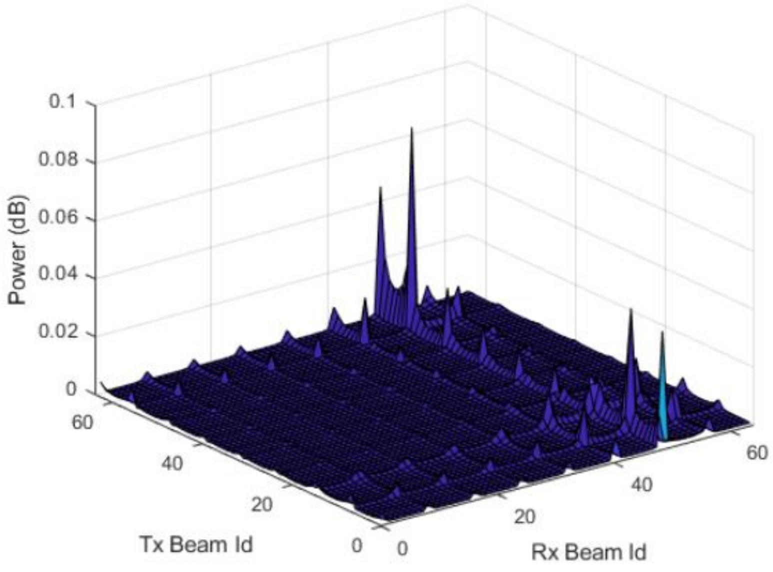
↓ H_{beam}

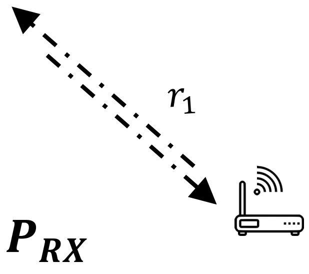
↓

P_{RX}

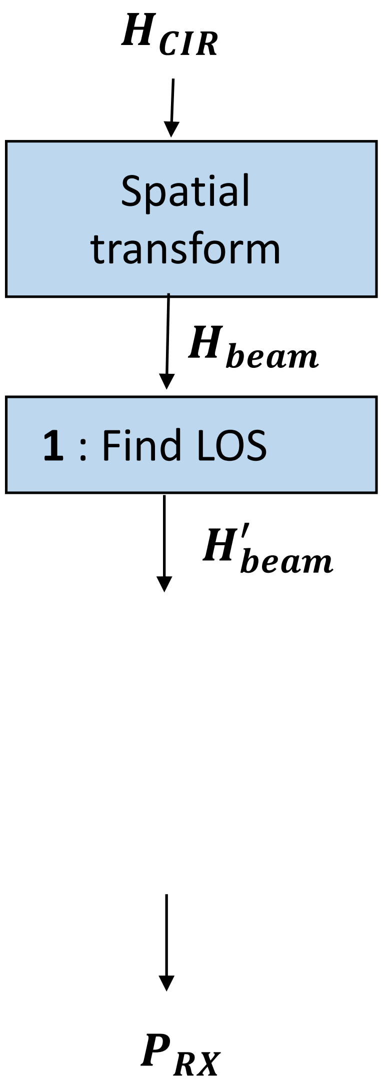


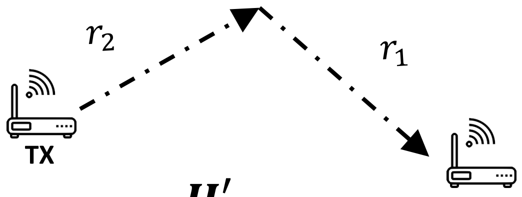
(AOD x AOA x Delay)

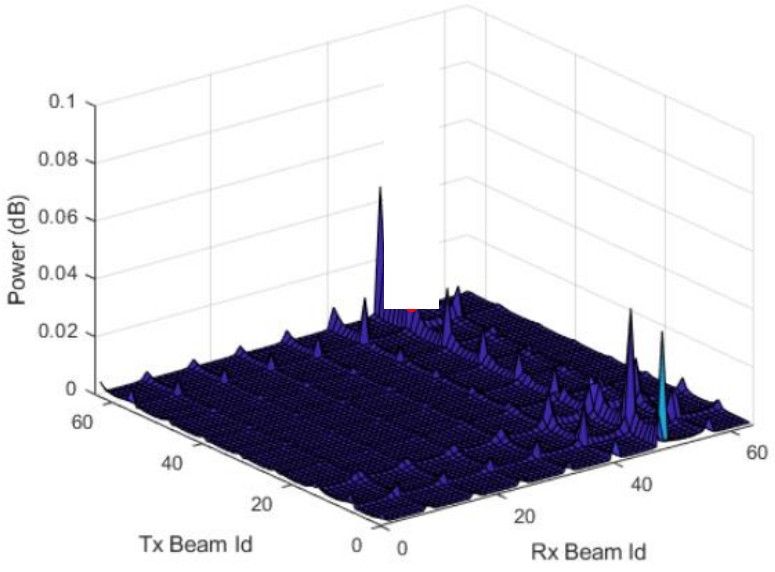




Pre-processing : Find LOS

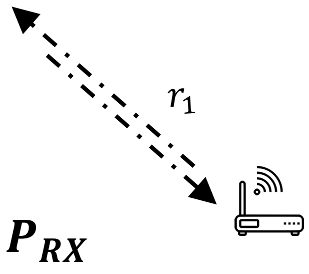






Select (TX,RX) Beam with

- Higher Power
- Lower Delay
- Tighter Beamwidth



Remove LOS from H_{beam} & Store LOS parameters

Pre-processing : Remove TX

SAMSUNG

H_{CIR}

↓

Spatial transform

↓ H_{beam}

1 : Find LOS

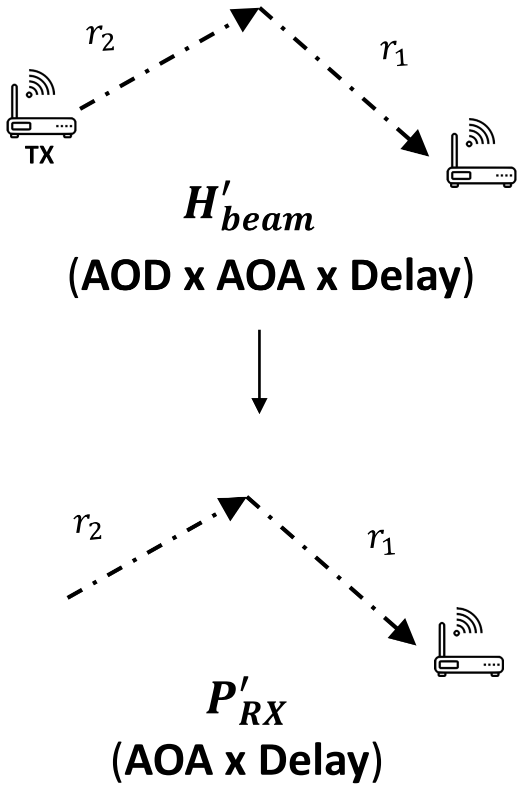
↓ H'_{beam}

2 : Remove TX

↓ P'_{RX}

↓

P_{RX}



Remove effect of TX by projecting AOD to Delay

Pre-processing : Remove TX

SAMSUNG

H_{CIR}

↓

Spatial transform

↓ H_{beam}

1 : Find LOS

↓ H'_{beam}

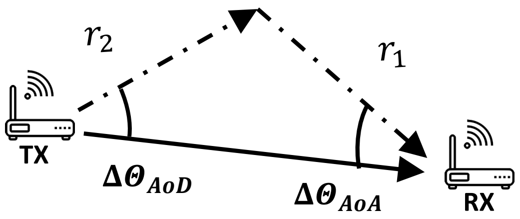
2 : Remove TX

↓ P'_{RX} **(AOA x Delay)**

↓

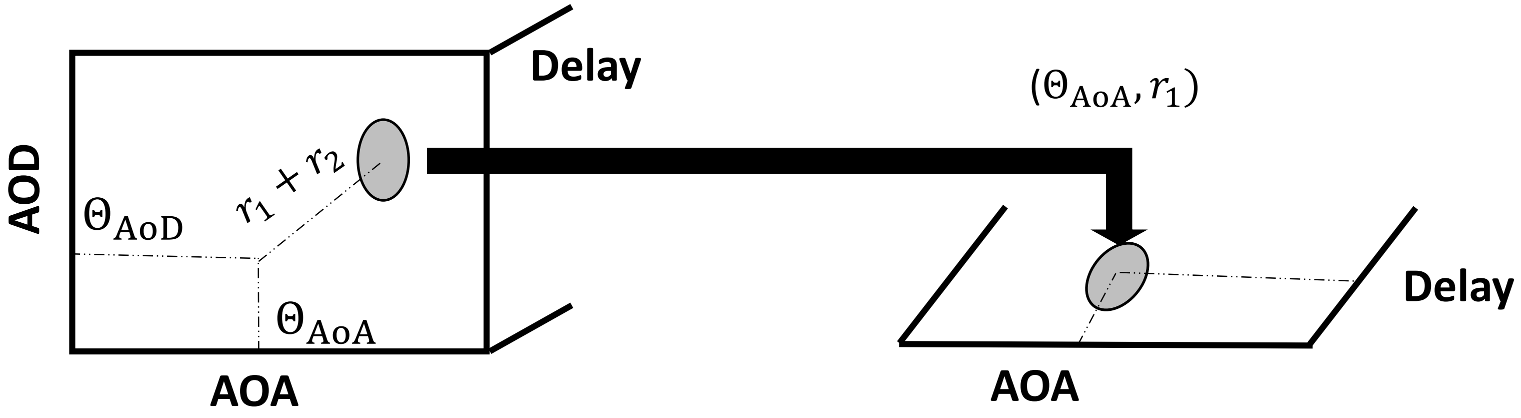
P_{RX}

Calculate $\frac{r_2}{r_1} = \frac{\sin(\Delta\Theta_{\mathrm{AoA}})}{\sin(\Delta\Theta_{\mathrm{AoD}})}$

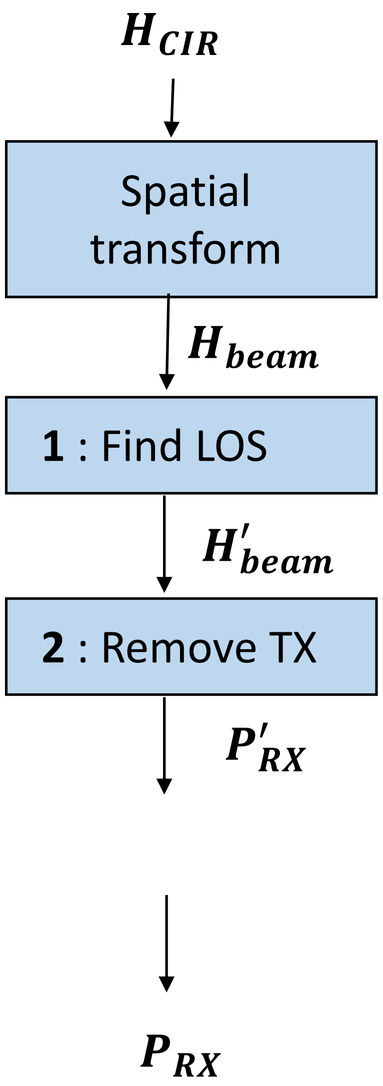


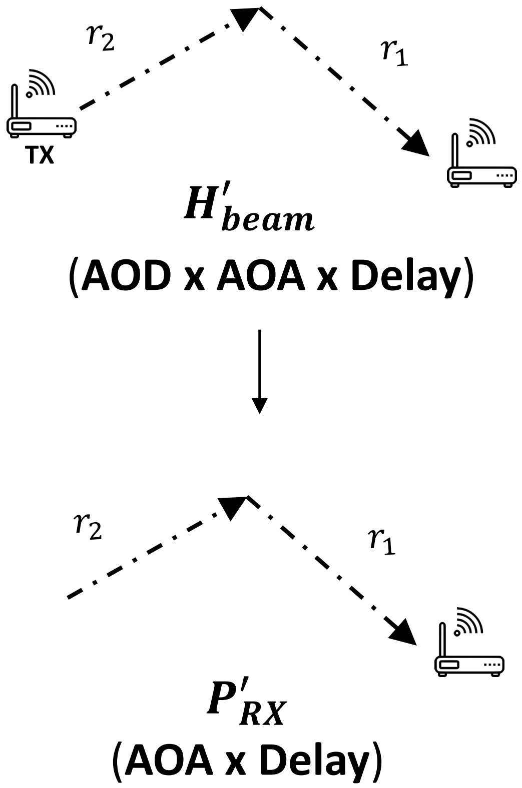
Project AOD to Delay

P'_{RX}



Pre-processing : Remove TX





Remove effect of TX by projecting AOD to Delay

Pre-processing : Rescale Power

SAMSUNG

$\boldsymbol{H_{CIR}}$

↓

Spatial transform

↓ $\boldsymbol{H_{beam}}$

1 : Find LOS

↓ $\boldsymbol{H'_{beam}}$

2 : Remove TX

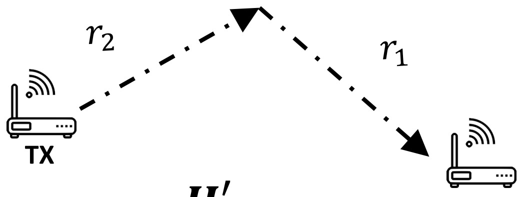
↓ $\boldsymbol{P'_{RX}}$

3 : Rescale power

↓

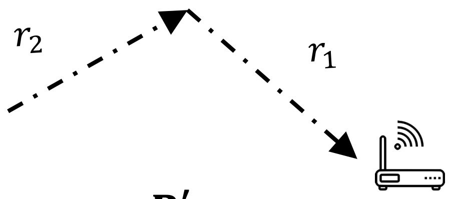
$\boldsymbol{P_{RX}}$

(AOA x Delay)



$\boldsymbol{H'_{beam}}$

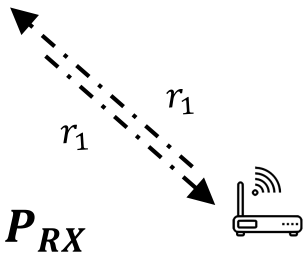
(AOD x AOA x Delay)



$\boldsymbol{P'_{RX}}$

(AOA x Delay)

$$\boldsymbol{P_{RX} = P'_{RX} * \frac{r_2^2}{r_1^2}}$$



$\boldsymbol{P_{RX}}$

(AOA x Delay)

Rescale to get monostatic power

Pre-processing :

SAMSUNG

$\boldsymbol{H_{CIR}}$

↓

Spatial transform

↓ $\boldsymbol{H_{beam}}$

1 : Find LOS

↓ $\boldsymbol{H'_{beam}}$

2 : Remove TX

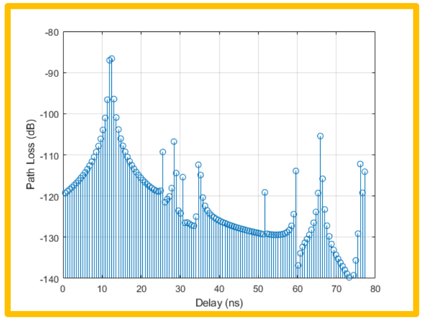
↓ $\boldsymbol{P'_{RX}}$

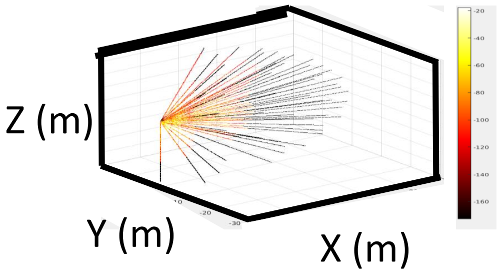
3 : Rescale power

↓

$\boldsymbol{P_{RX}}$

Input: CIR for 64x64 MIMO

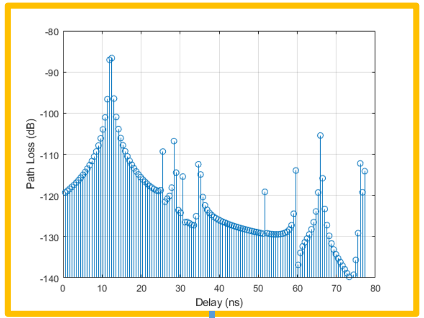




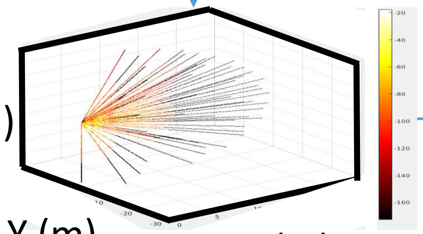
Input and Output for ML Model Training

SAMSUNG

Given Input: CIR for 64x64 MIMO



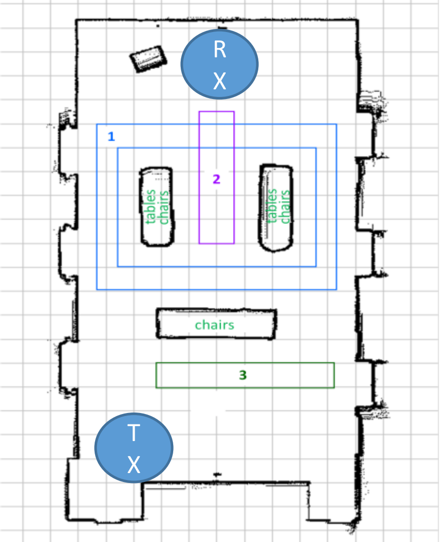
Pre-processing



Z (m)

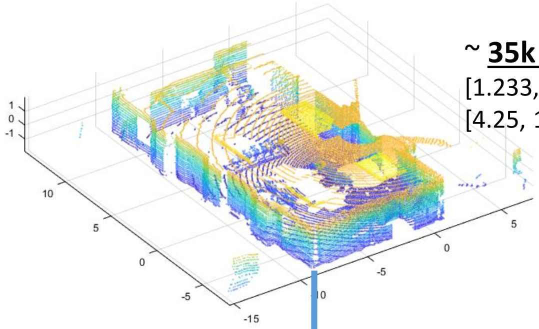
Y (m)

X (m)



ML Model

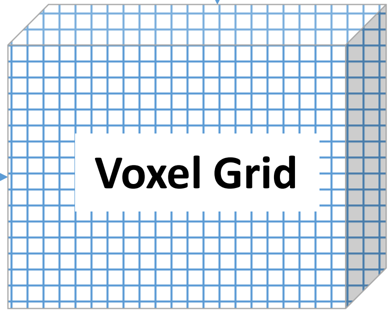
Given Output (ground truth): LIDAR PCD



~ **35k points**
[1.233, 8.342, -1.20]
[4.25, 10.57, -0.25] ….

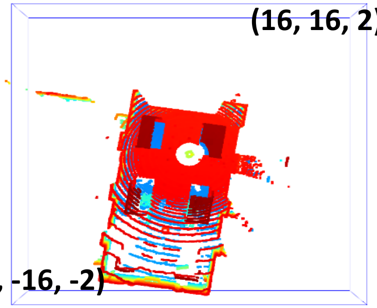
If a point is in voxel:
Voxel value =1

(16, 16, 2)



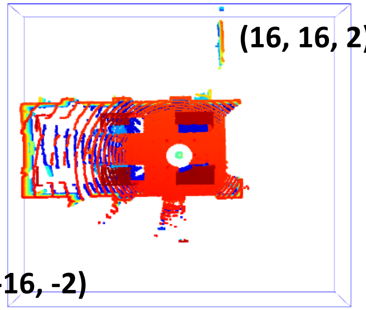
(-16, -16, -2)

(16, 16, 2)



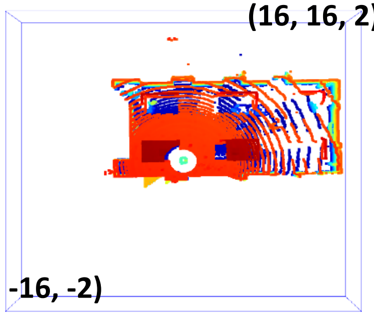
(-16, -16, -2)

(16, 16, 2)



(-16, -16, -2)

(16, 16, 2)



(-16, -16, -2)

Input for the ML model:
Power of reflections from :

- 64 directions : (8 Az x 8 El)
- Each direction 100 points*.

*Distance between consecutive points ~ 17cm

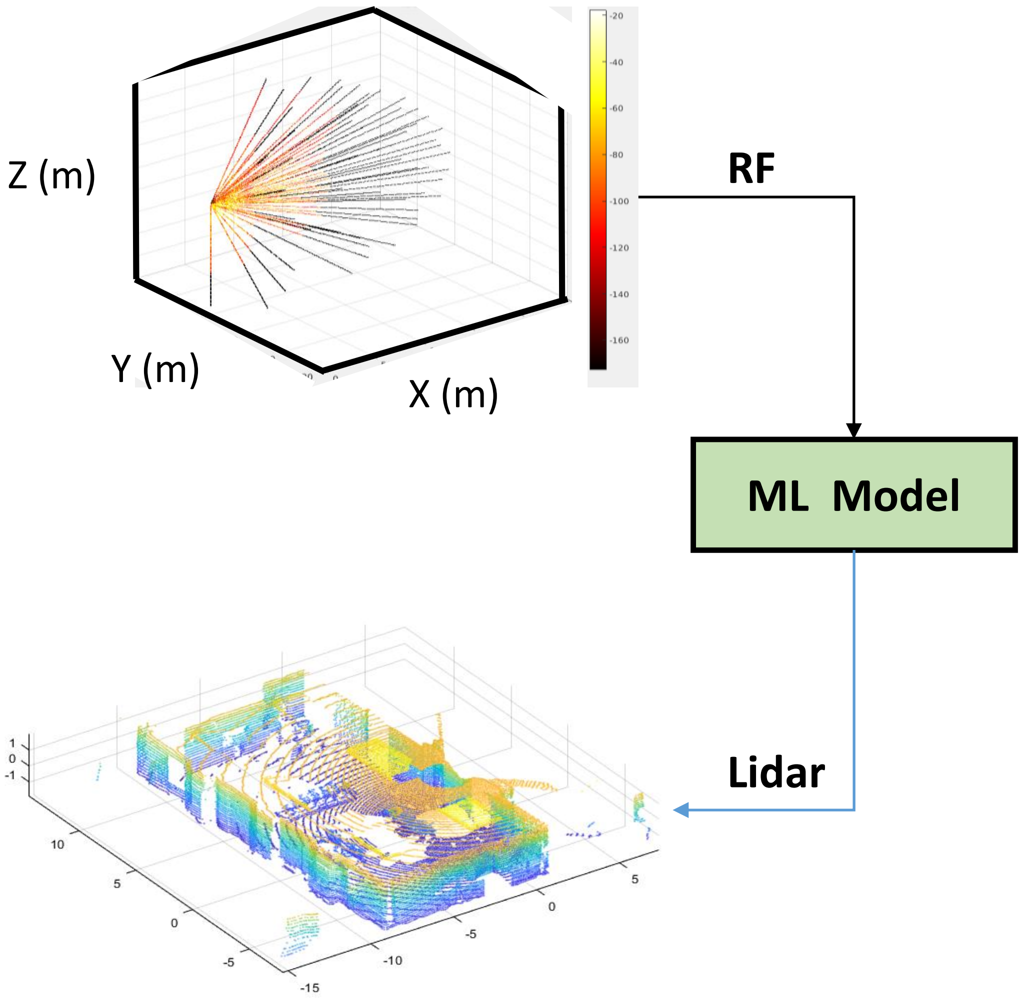
Output labels for ML model: Voxel Grid:

- Voxel size =**0.25** m -> Dim: **128 x 128 x 16**
- Voxel size=0.5 m -> Dim: 64x64x8

~ 13k voxels are 1 out of : 128 x 128 x 16 voxels

Solution : Why is it difficult ?

SAMSUNG

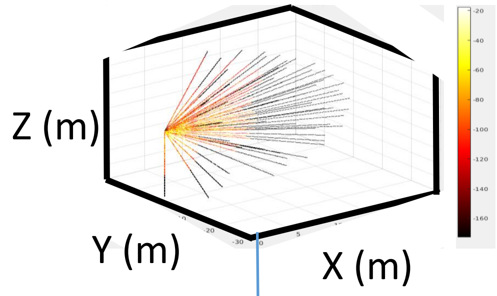


	Pre-processed RF data	LIDAR Data
Range resolution	0.17 m	10^-6 m
Angle resolution	Low (~ 22.5 º)	High
I/O Dimension	6400x1	2,62,000 voxels (voxel grid: 128x128x16)
Location of the Tx and Rx is unknown		

Solution is novel and first of its kind.

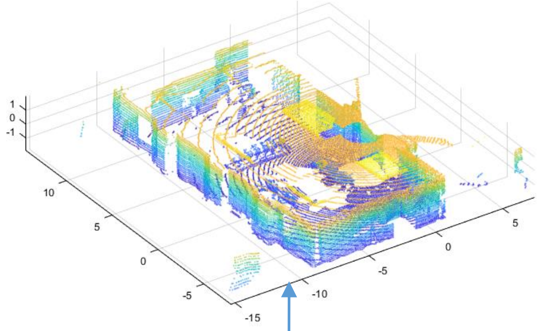
ML model architecture

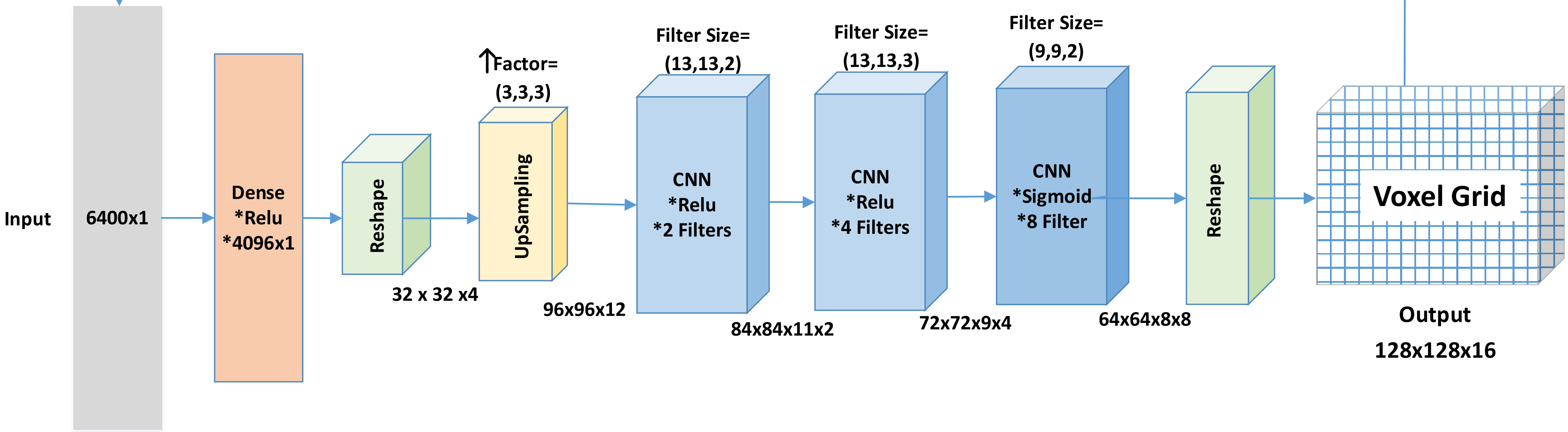
From Pre-processing step:



<u>Motivation behind used layers:</u>

- **Up-sampling** layer
  - **Ratio I/O** dimension=**3%**
- **CNN** layers:
  - **Correlation** in reflections exists in **close** neighbourhood





Training parameters

SAMSUNG

Custom Loss function: binary cross entropy with weights where $\boldsymbol{w_1} = \mathbf{10}$ **and** $\boldsymbol{w_0} = \mathbf{1}$.

- As the output voxel grid is sparse, training is skewed towards the label '1'.

$$L = -\frac{1}{N}\sum_{i=1}^{N} \left(\boldsymbol{w_1} * y_i \log(p(y_i)) + \boldsymbol{w_0}\, (1 - y_i\,) \log(1 - p(y_i)) \right)$$

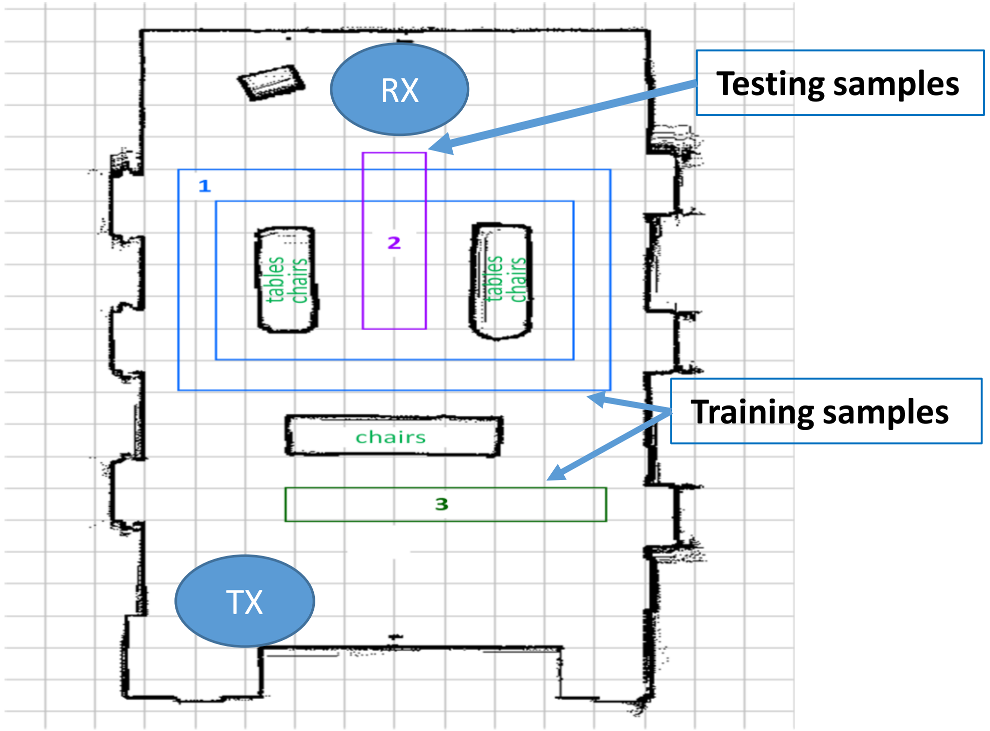
~ 13k voxels are 1 out of : 128 x 128 x 16 voxels

Dataset:

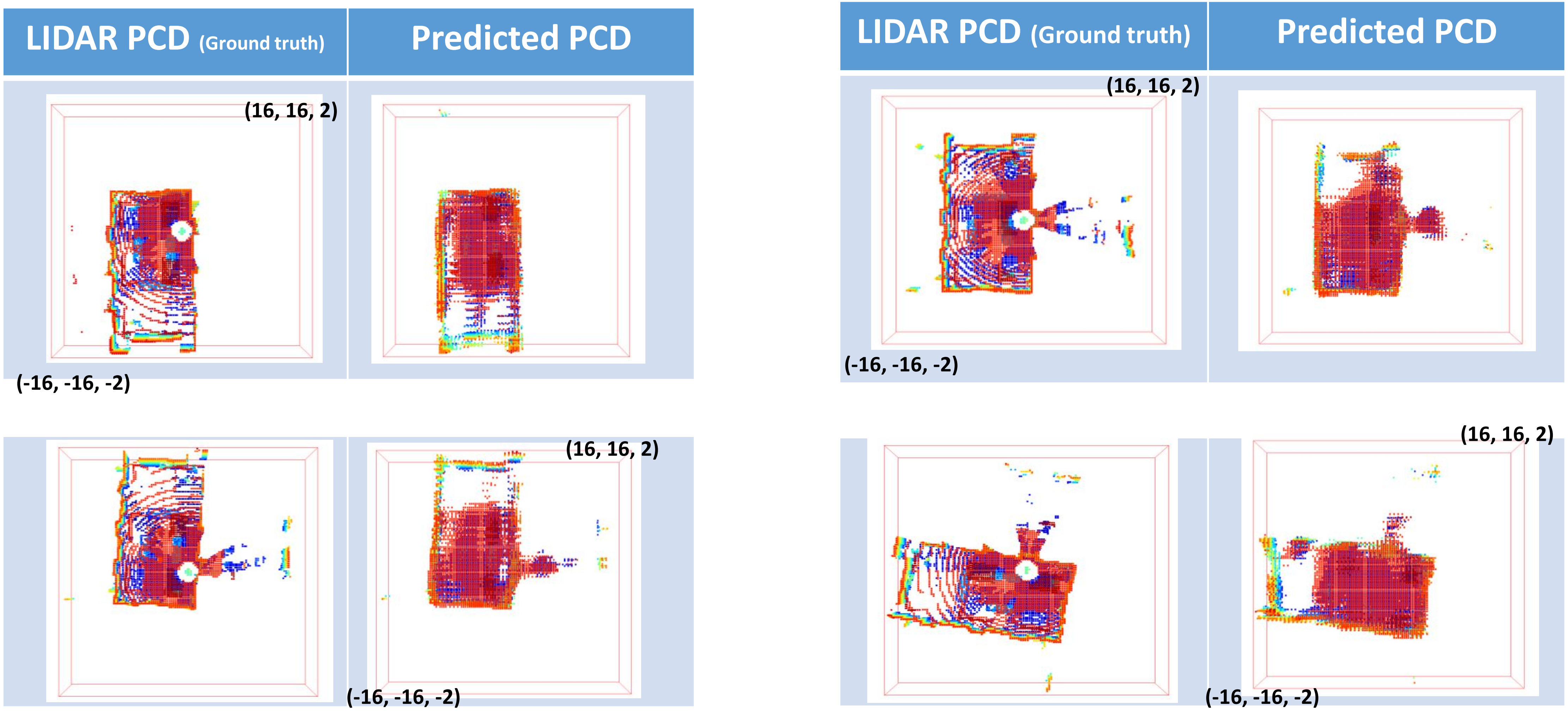
- Total training samples : 3400 samples
  - Area 1 : 2400 samples
  - Area 3: 1000 samples
- Validation samples : 350 (consisting of both the areas)
- Test data: 529 samples of Area 2

Optimizer:

- Adam with learning rate of 0.0005
- Decay rate: 0.9 every 10000 steps
- Epochs =100, batch size : 32.

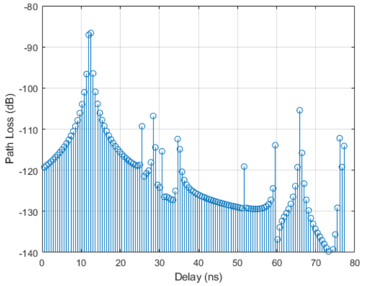


Prediction vs Actual LIDAR PCD

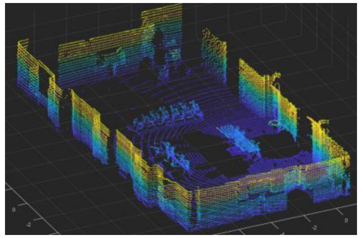


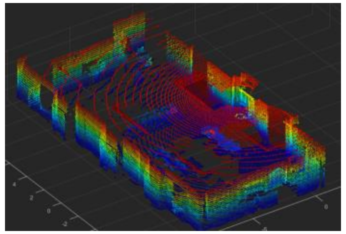
- The ML model **proposed** returns **LIDAR-like high resolution** point clouds
- **Change in perception**, is nicely **captured** even though the Tx and Rx locations are **unknown**.

Evaluation Metric for the challenge



Input Data: RF Data Evaluation Dataset → **Trained Model** → **Output: Predicted PCD** → **Evaluation** ← **Ground Truth: LiDAR Evaluation dataset**



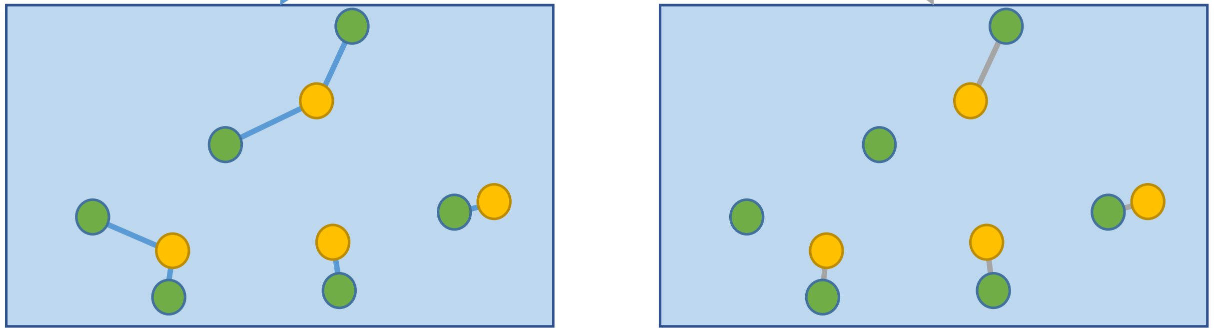


Chamfer Distance
measures discrepancies between point clouds.

$$d_{CD}(S_1, S_2) = \frac{1}{|S_1|}\sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \frac{1}{|S_2|}\sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2$$

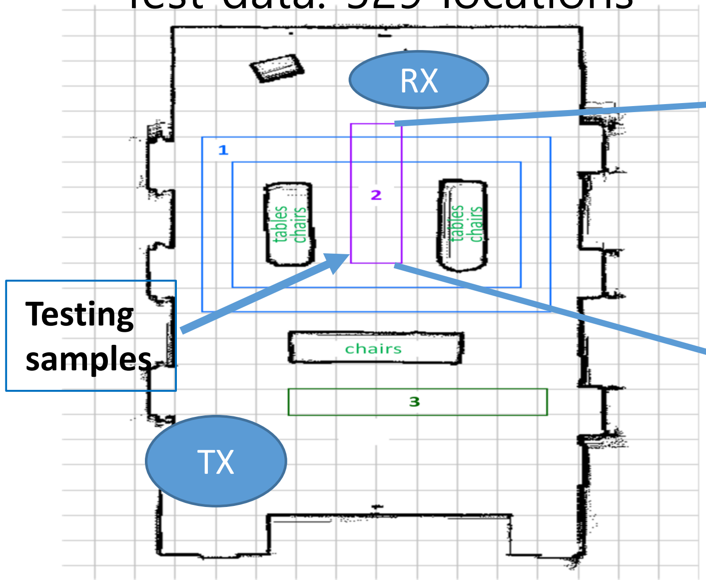
LIDAR PCD: S_1

Predicted PCD: S_2

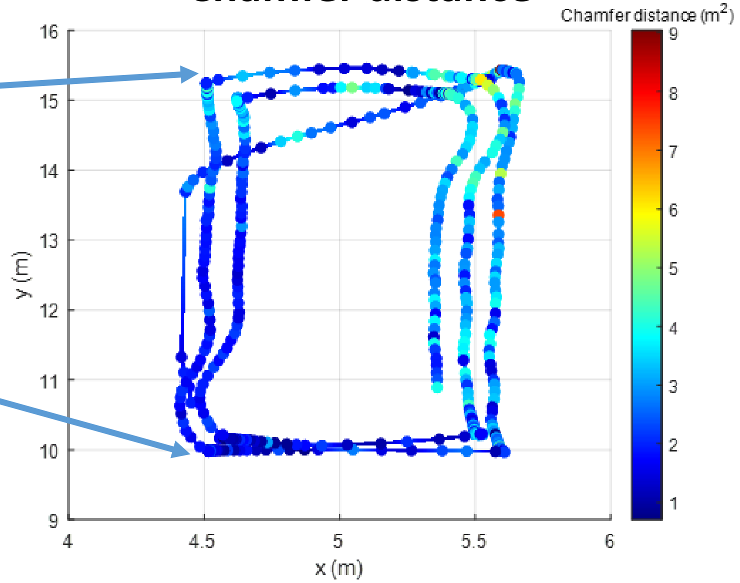


Evaluation : Error Analysis on Testing Data

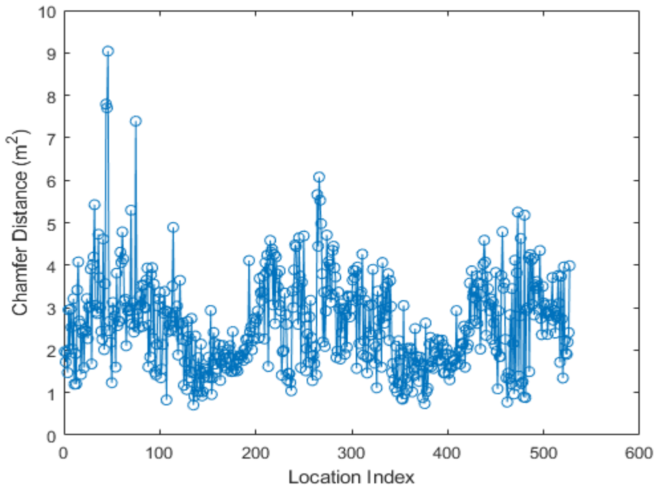
Test data: 529 locations



Chamfer distance



Chamfer distance



Test data: 529 locations

Chamfer distance for Testing data

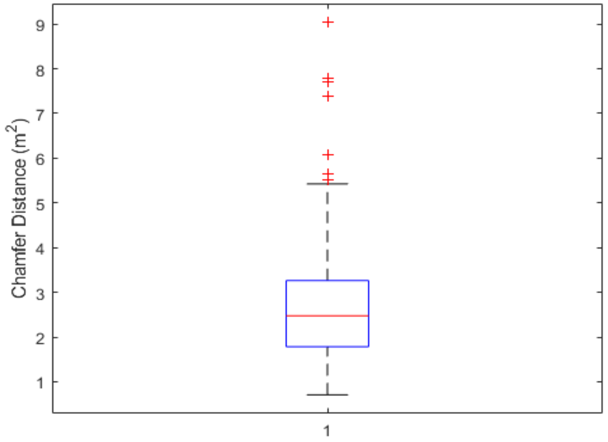
- Average **Chamfer distance = 2 m^2** which is **good** for the room of size ~16 m x 16 m x 4 m

- Average **Chamfer distance = 1 m^2** for the samples which are closer to the Tx.

Chamfer distance for Training data

- Average **Chamfer distance = 1.5 m^2** for voxel-size=0.25 m
- Average **Chamfer distance = 2.2 m^2** for voxel-size=0.5 m

Chamfer distance



Conclusion and Future work..

SAMSUNG

Observations:

- The ML model **proposed** returns **LIDAR-like high resolution** point clouds.
- **Change in perception of room** across locations, is nicely **captured in the predictions**.
- Average **Chamfer distance = 2 m^2** which is **good** for the room of size ~16 m x 16 m x 4 m
- Average **Chamfer distance = 1 m^2** for the samples which are closer to the Tx.

Future Work :

- Train the network with higher resolution input data.
- Experiment with the Voxel grid size
- Global training inclusive of all the locations
- Train using Tx perspective along with Rx perspective

Thank you.

Questions ?

We thank the NIST, USA and ITU challenge team for providing this exciting problem.