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

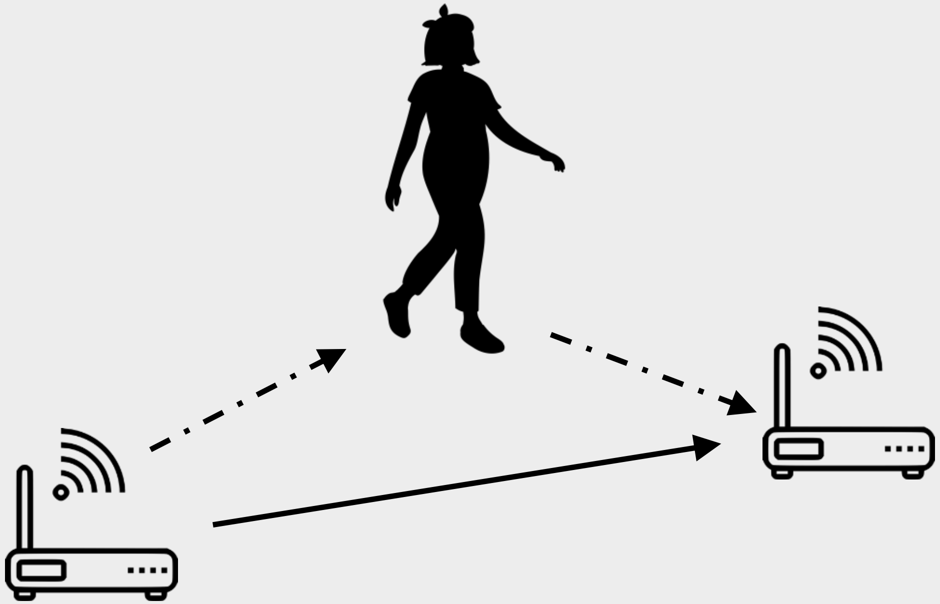
Affiliation *: Beyond 5G Team, Samsung R&D Institute India, Bengaluru.*

Problem Organizer*: NIST, USA.*

Motivation : Communication waves to create Digital world

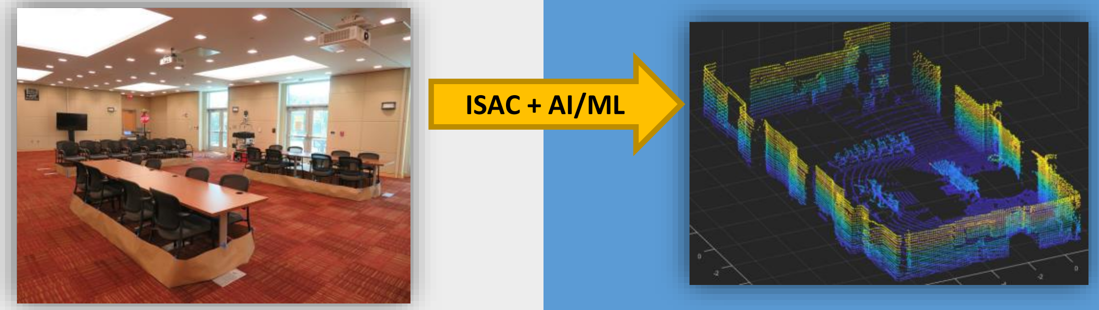
SAMSUNG

Integrated Sensing & communications



Inference using RF communication signal

Physical world | ***Digital world***



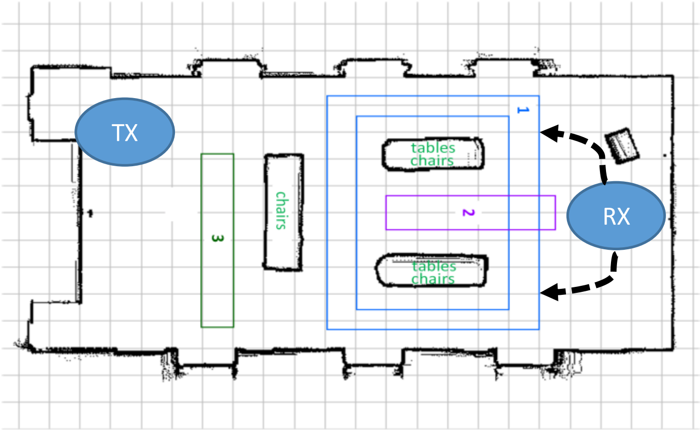
Reuse **spectrum, devices and protocols to perform both communication and sensing.**

Extra processing required to acquire multi-dimensional data

Problem statement

- **Setup** :

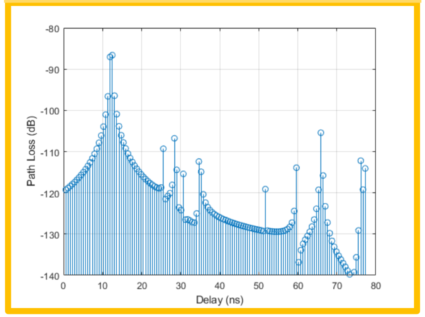
One fixed transmitter

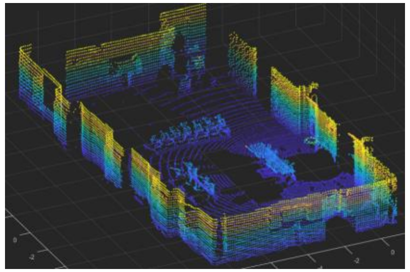


One moving receiver

- **ML Model** :

Channel Impulse response

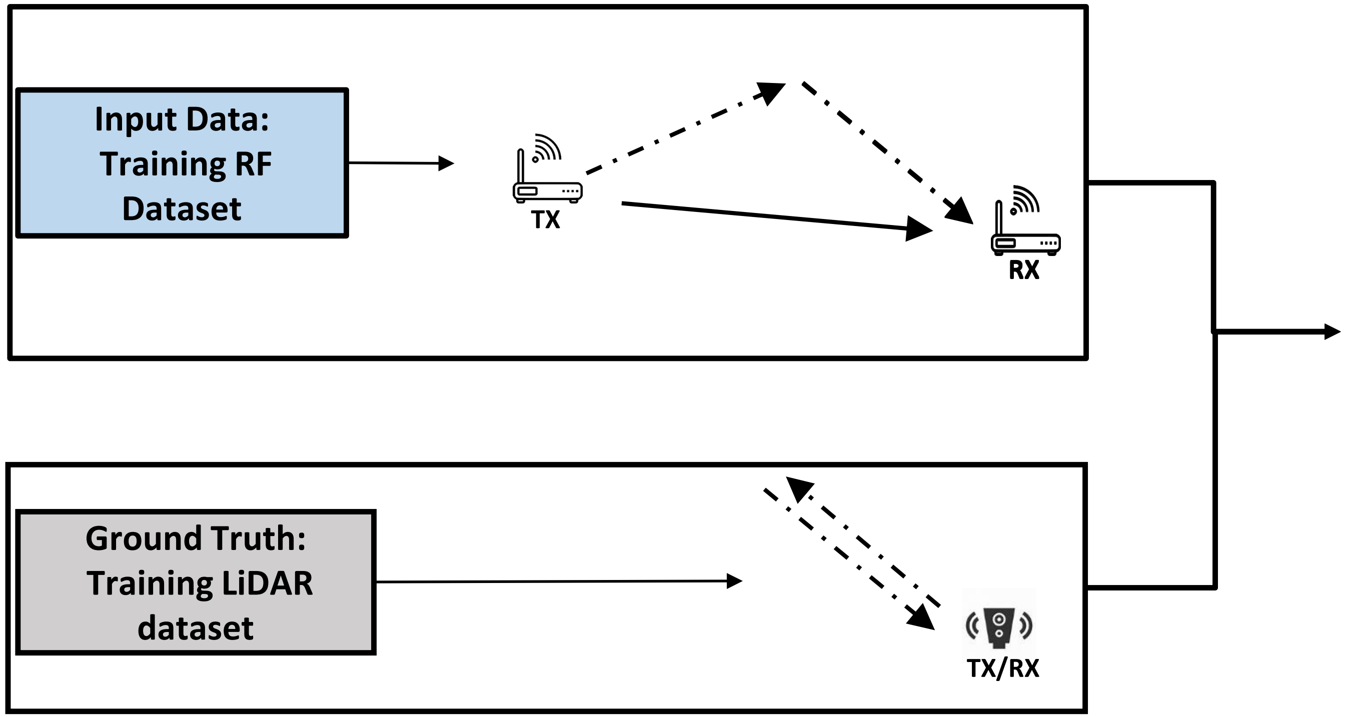




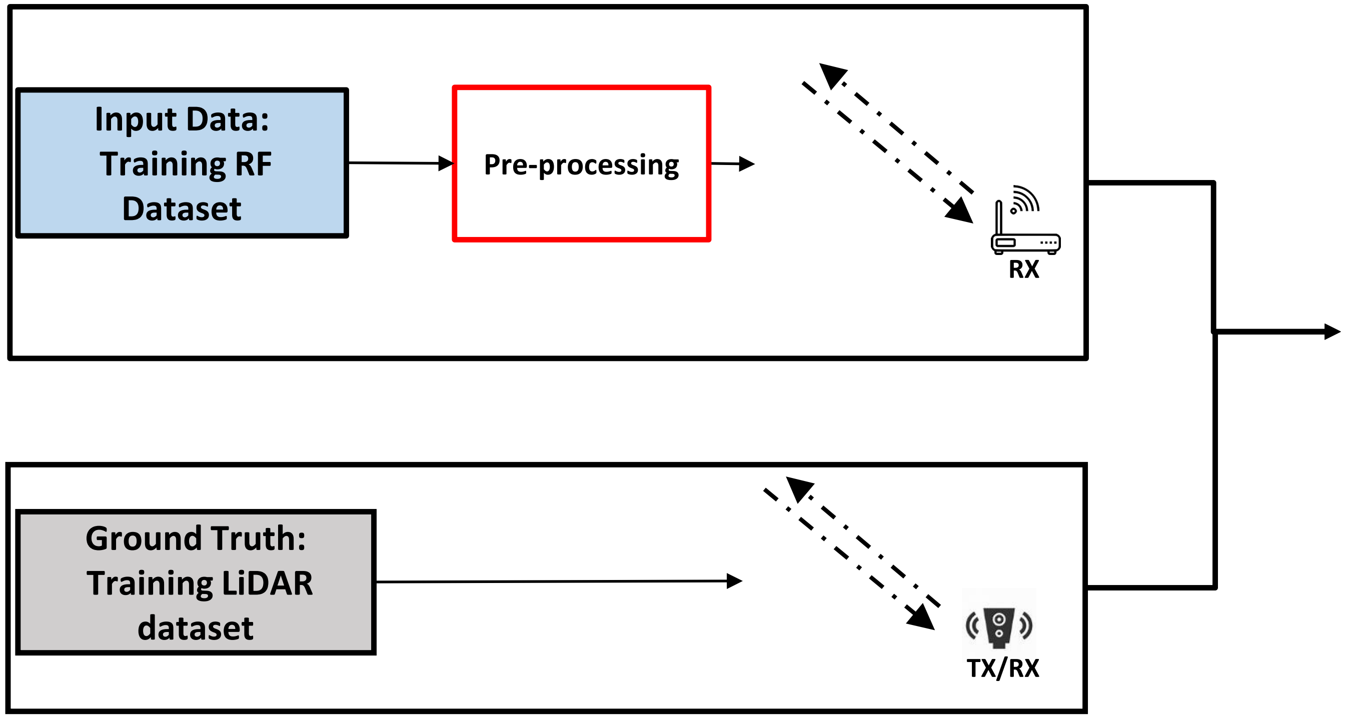
Input Data: Training RF Dataset → **Trained Model** → **Output: LiDAR Point Clouds**

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

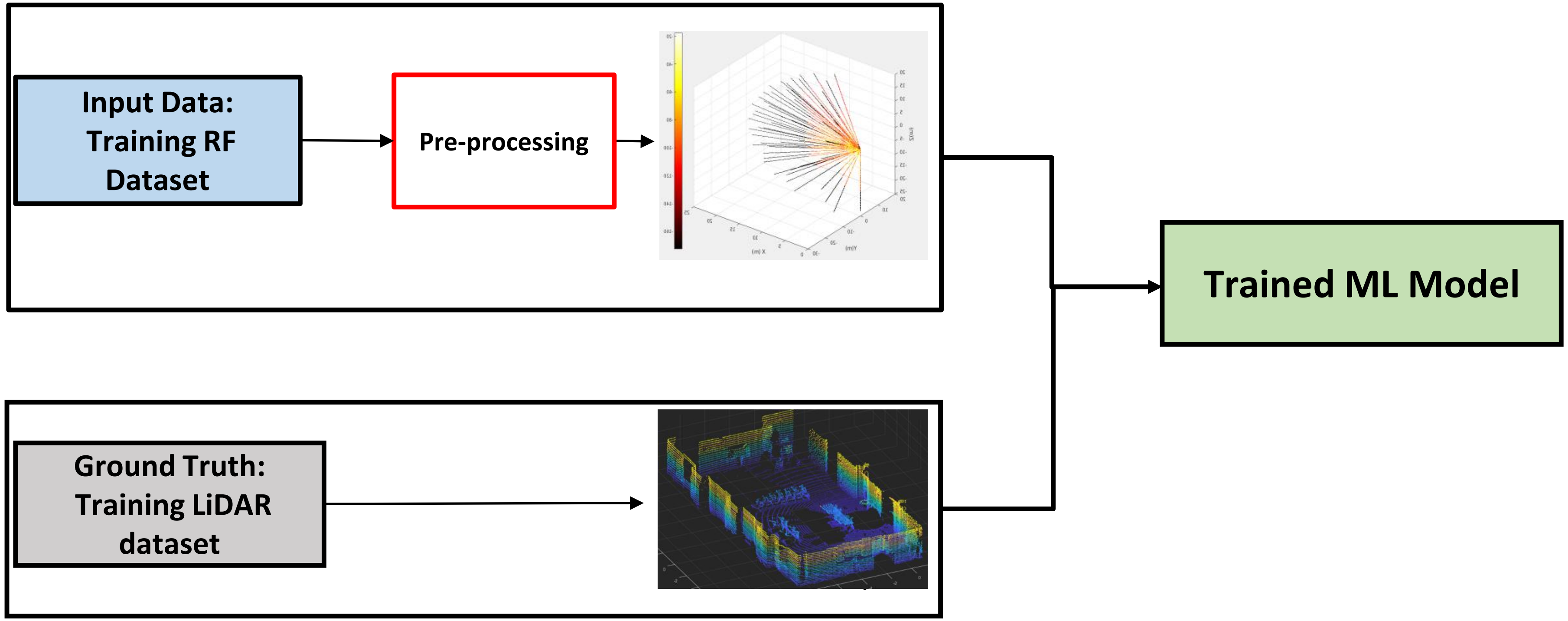
Solution outline



Solution outline



Solution outline

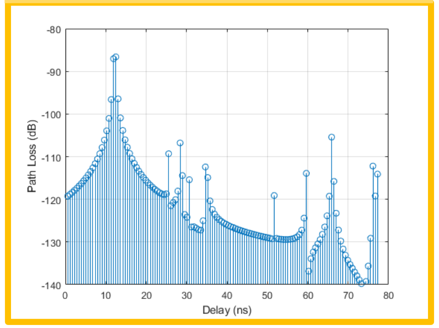


Solution : At each RX location

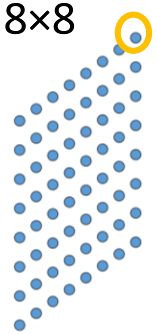
- Transform bi-static RF data to mono-static format.
- Train ML model by fitting it to similarly structured LiDAR ground truth.

Pre-processing : Spatial Transform

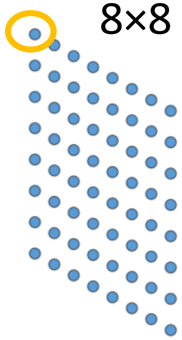
Channel Impulse response

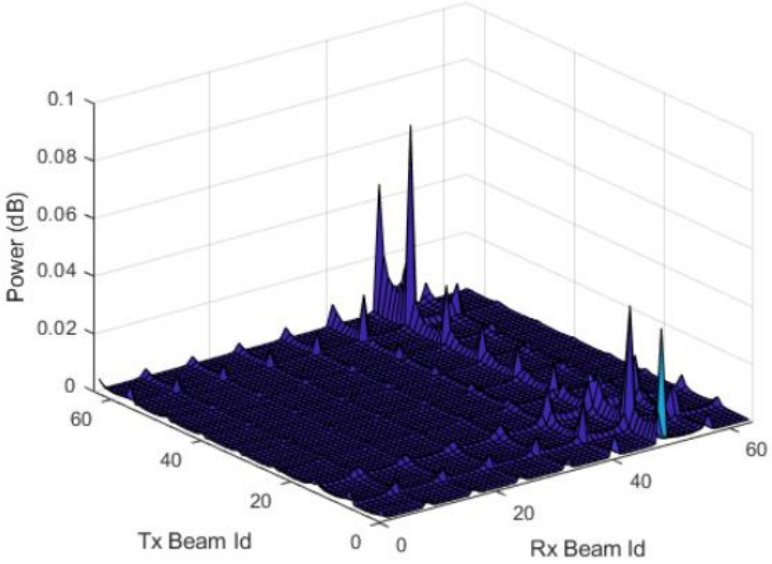


8×8



8×8





H_{CIR}

(TX Ant x RX Ant x Delay)

Spatial transform

H_{beam}

(AOD x AOA x Delay)

Delay $\mathrm{T}_s \frac{1}{1.76} ns$	⇔	**Distance resolution of** $\mathbf{0.17}\ m.$
Horz.Ant 8	⇔	**Azimuth resolution of 22.5 ° (Avg.)**
Vert.Ant 8	⇔	**Elevation resolution of 22.5 ° (Avg.)**

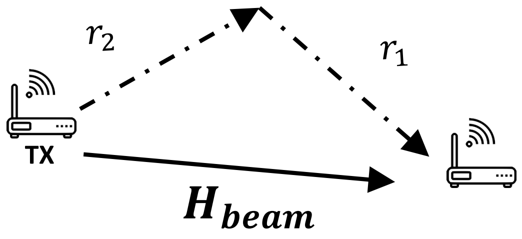
Transforms Communication data to Sensing framework

Pre-processing : Spatial Transform

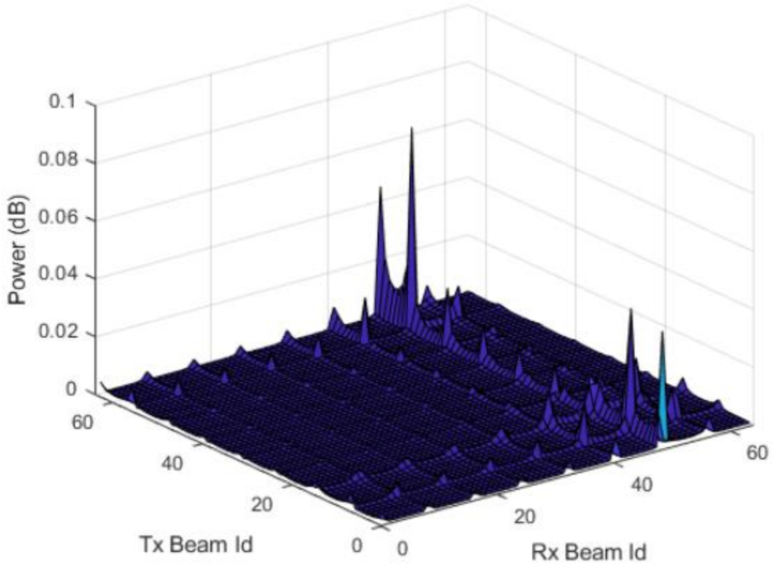
H_{CIR}

Spatial transform

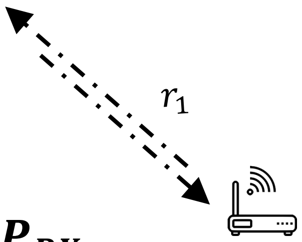
H_{beam}



(AOD x AOA x Delay)



P_{RX}



P_{RX}

Pre-processing : Find LOS

SAMSUNG

H_{CIR}

↓

Spatial transform

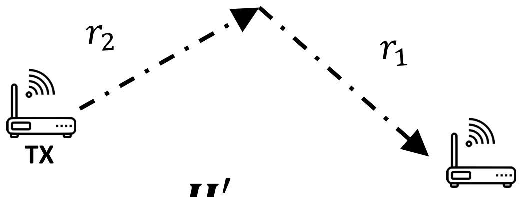
↓ H_{beam}

1 : Find LOS

↓ H'_{beam}

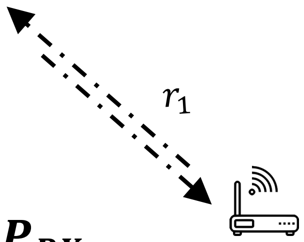
↓

P_{RX}

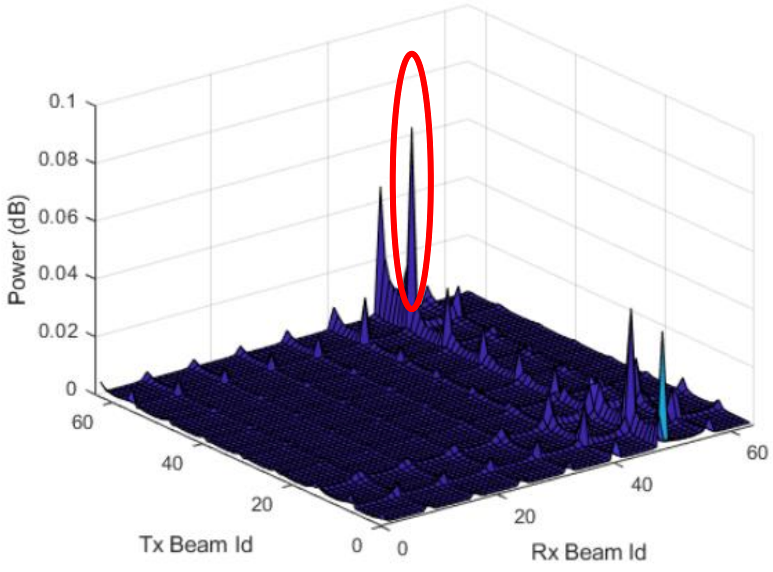


H'_{beam}

(AOD x AOA x Delay)



P_{RX}

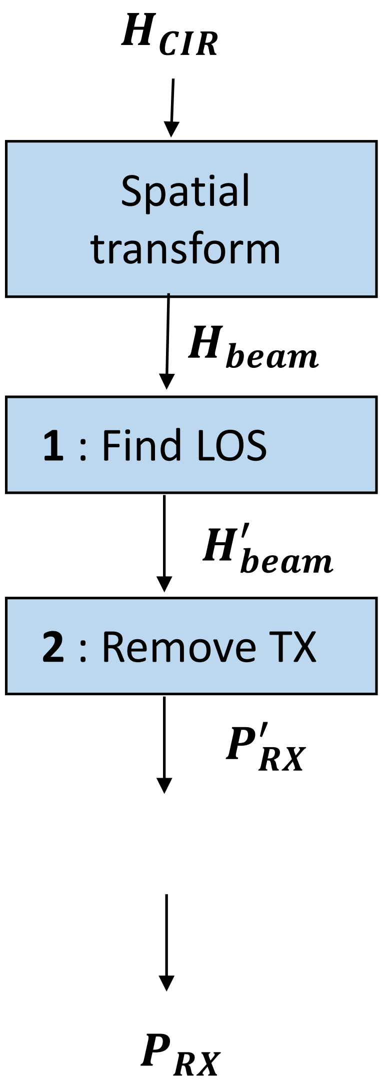


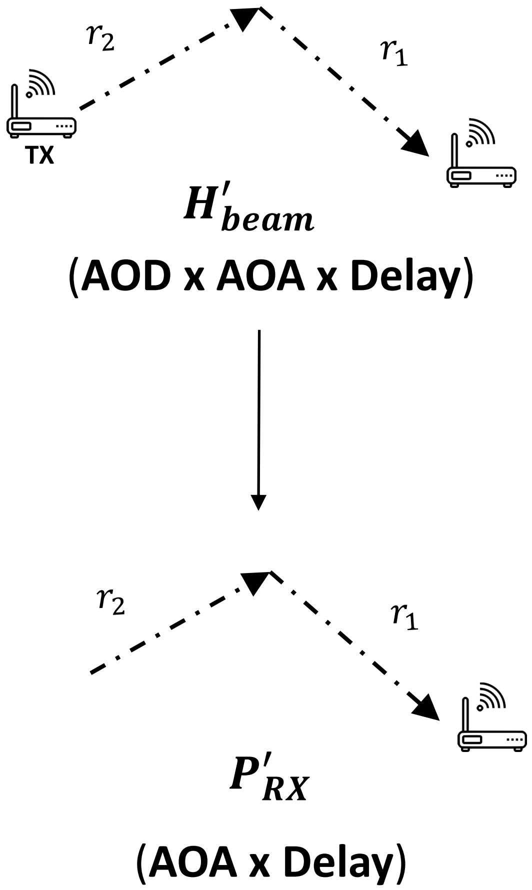
Select (TX,RX) Beam with

- Higher Power
- Lower Delay
- Tighter Beamwidth

Remove LOS from H_{beam} & Store LOS parameters

Pre-processing : Remove TX





Pre-processing : Remove TX

SAMSUNG

$\boldsymbol{H_{CIR}}$

↓

Spatial transform

↓ $\boldsymbol{H_{beam}}$

1 : Find LOS

↓ $\boldsymbol{H'_{beam}}$

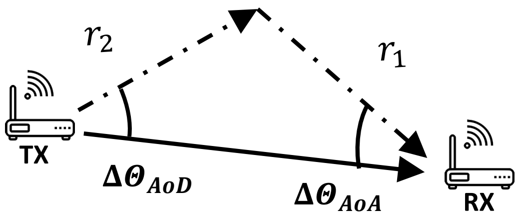
2 : Remove TX

↓ $\boldsymbol{P'_{RX}}$ **(AOA x Delay)**

↓

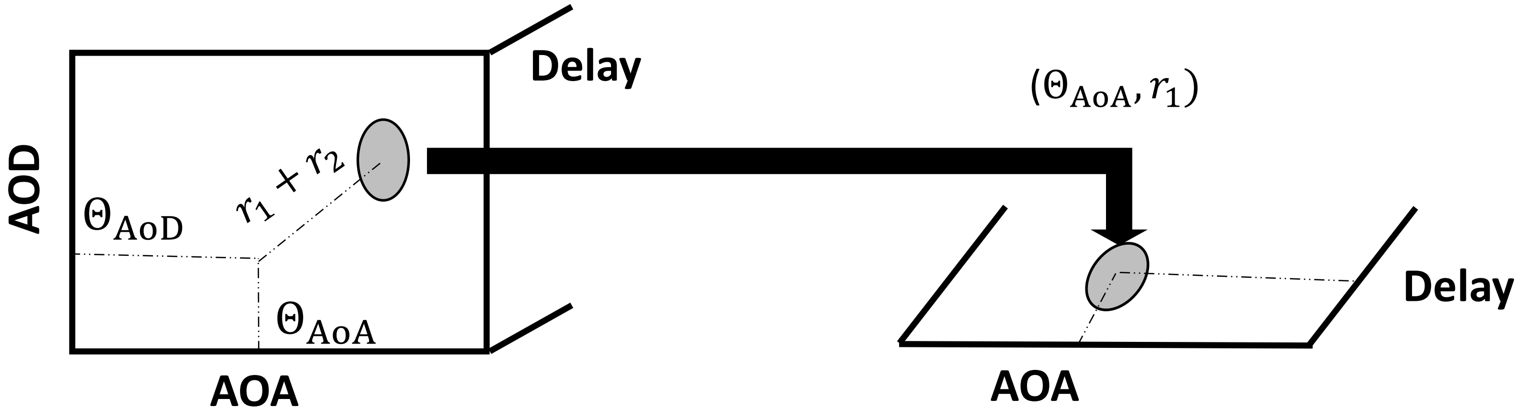
$\boldsymbol{P_{RX}}$

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



Project AOD to Delay

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



Pre-processing : Remove TX

$\boldsymbol{H_{CIR}}$

↓

Spatial transform

↓ $\boldsymbol{H_{beam}}$

1 : Find LOS

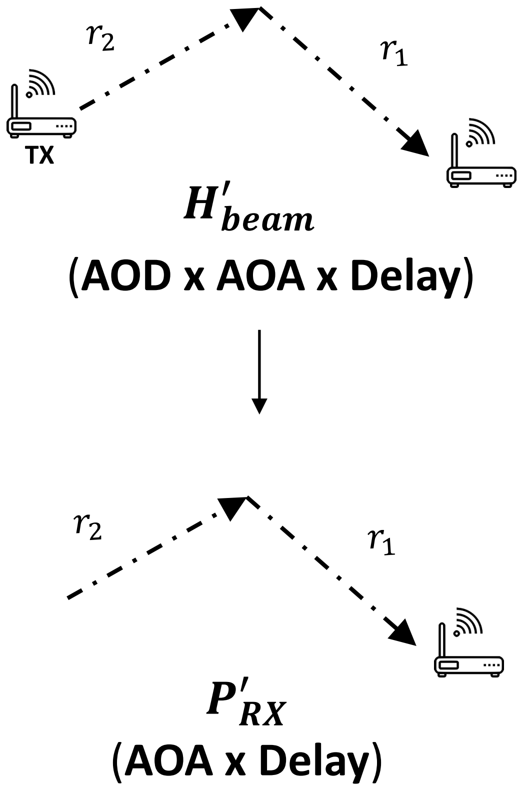
↓ $\boldsymbol{H'_{beam}}$

2 : Remove TX

↓ $\boldsymbol{P'_{RX}}$

↓

$\boldsymbol{P_{RX}}$



Remove effect of TX by projecting AOD to Delay

Pre-processing : Rescale Power

H_{CIR}

↓

Spatial transform

↓ H_{beam}

1 : Find LOS

↓ H'_{beam}

2 : Remove TX

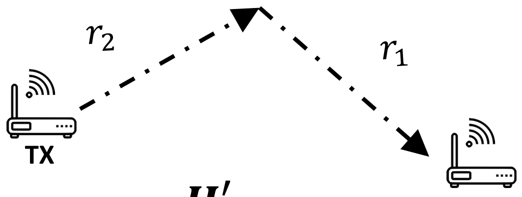
↓ P'_{RX}

3 : Rescale power

↓

P_{RX}

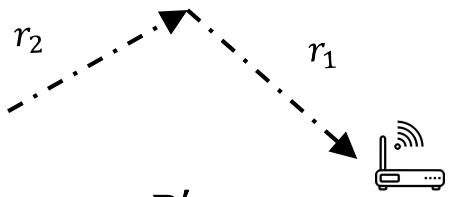
(AOA x Delay)



H'_{beam}

(AOD x AOA x Delay)

↓

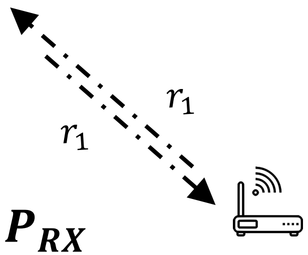


P'_{RX}

(AOA x Delay)

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

→



P_{RX}

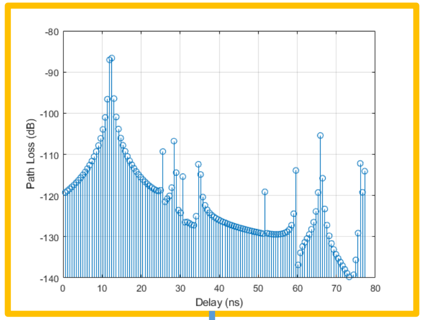
(AOA x Delay)

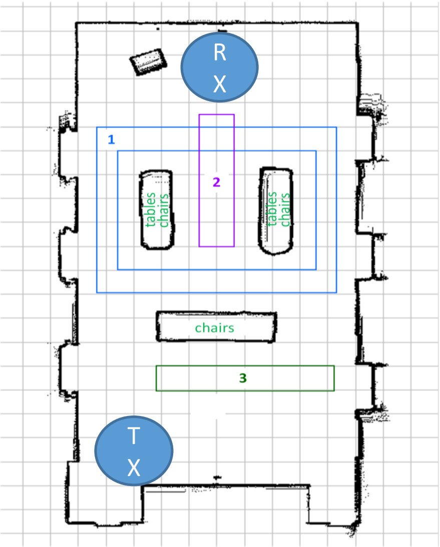
Rescale to get monostatic power

Input and Output for ML Model Training

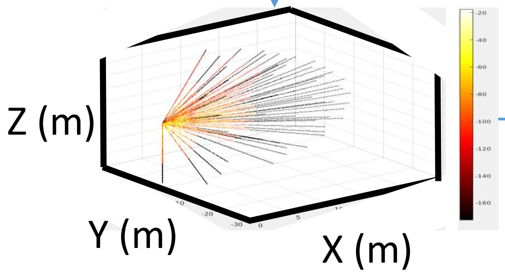
SAMSUNG

Given Input: CIR for 64x64 MIMO



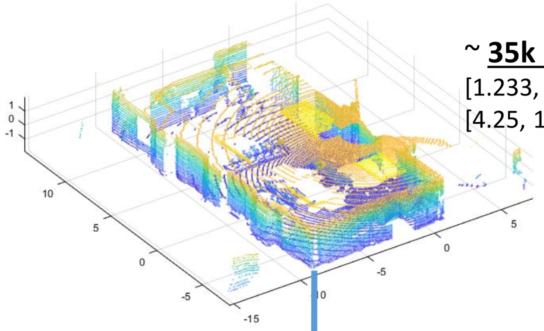


Pre-processing



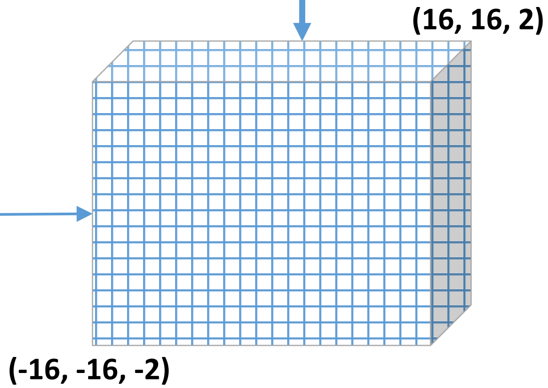
ML Model

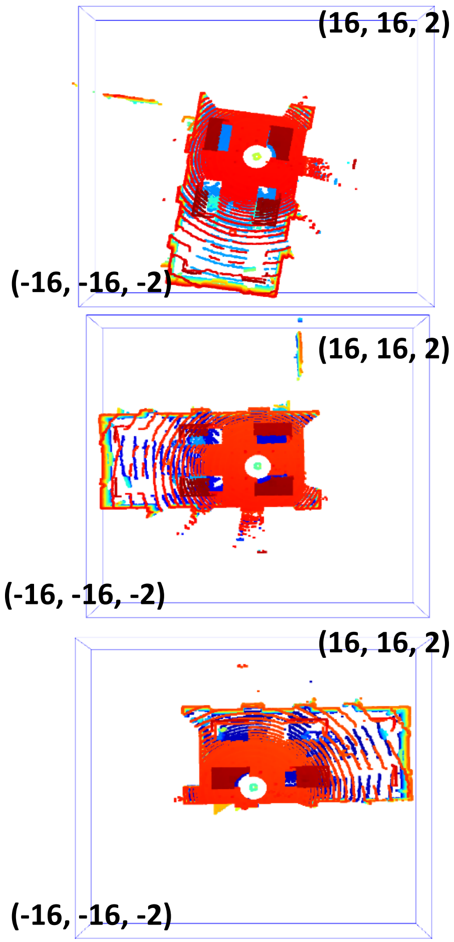
Given Output (ground truth): LIDAR PCD



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

If a point is in voxel:
Voxel value =1





Input for the ML model:
Power of reflections from:
- 64 directions : (8 Az x 8 El)
- Each direction 100 points*.

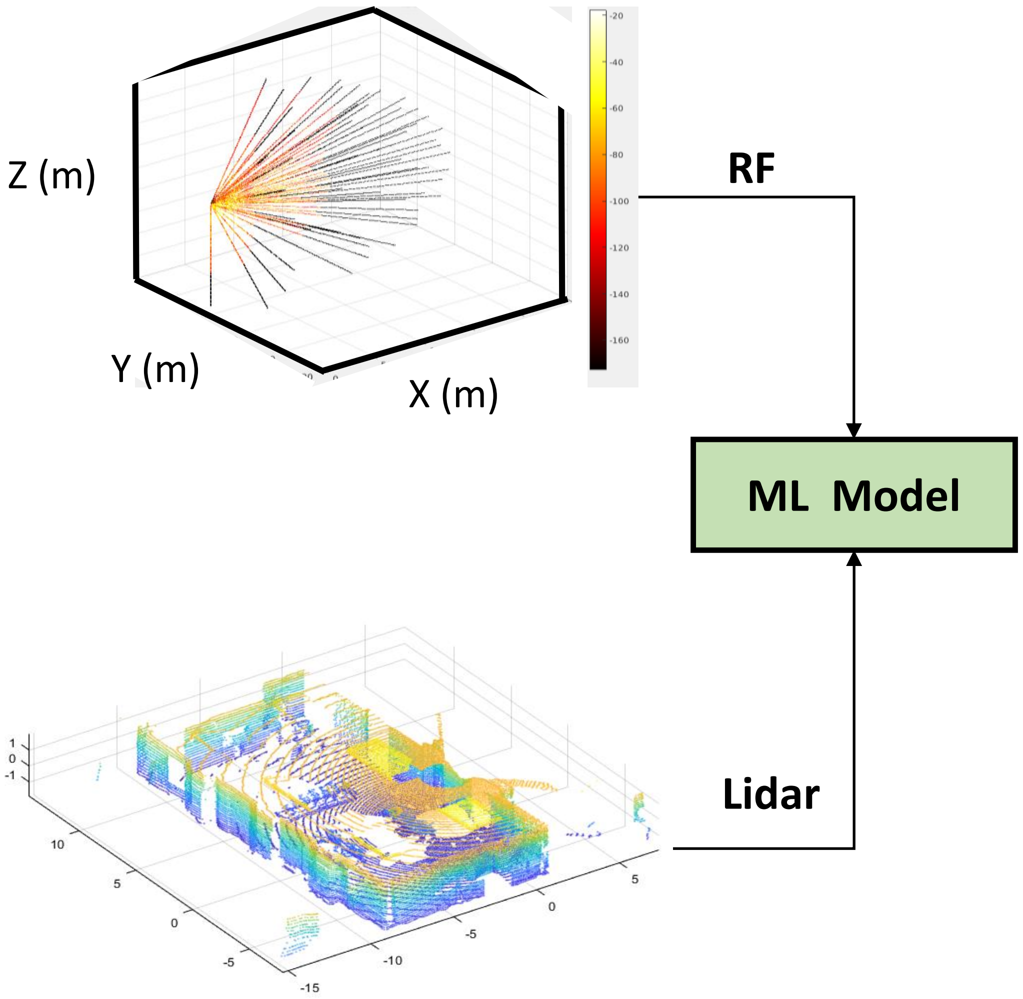
*Distance between consecutive points ~ 17cm

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 ?

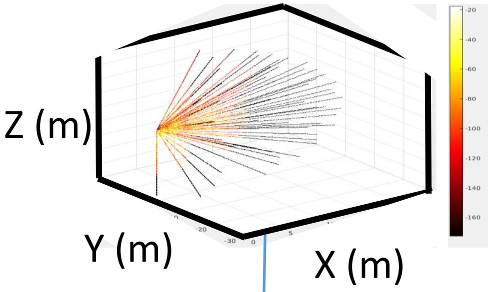
SAMSUNG



	Pre-processed RF data	LIDAR Data
Data	Power values @ 100 points x 64 directions (8 Az x 8 El)	Co-ordinates @ 35k pts of reflections
Range resolution	0.17 m	10^-6 m
Angle resolution	Low (~ 22.5 ^{o})	High
I/O Dimension	6400x1	262k (voxel grid: 128x128x16)

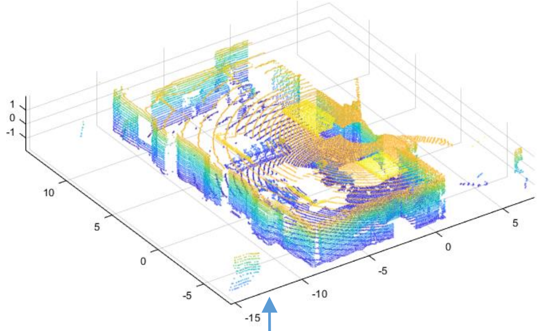
ML model architecture

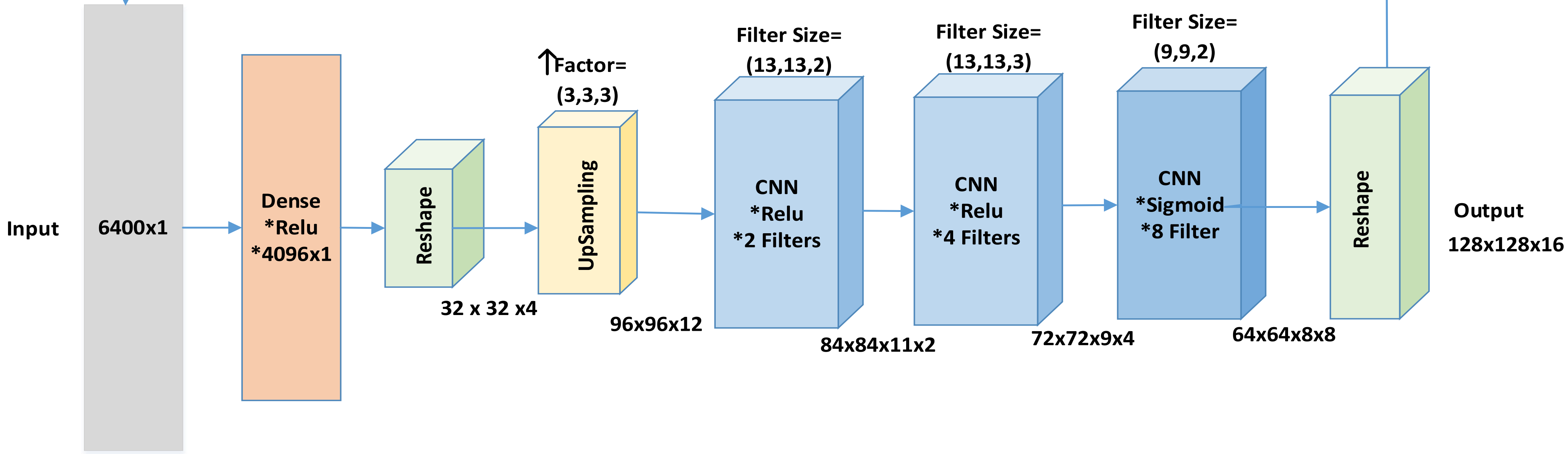
From Pre-processing step:



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

- Input dim =6400, Output dim=128x128x16=2,62,000
  Ratio I/O=3% -> Up-sampling layer
- **Correlation** in reflections exists in **close** neighbourhood
  -> CNN is used





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'.

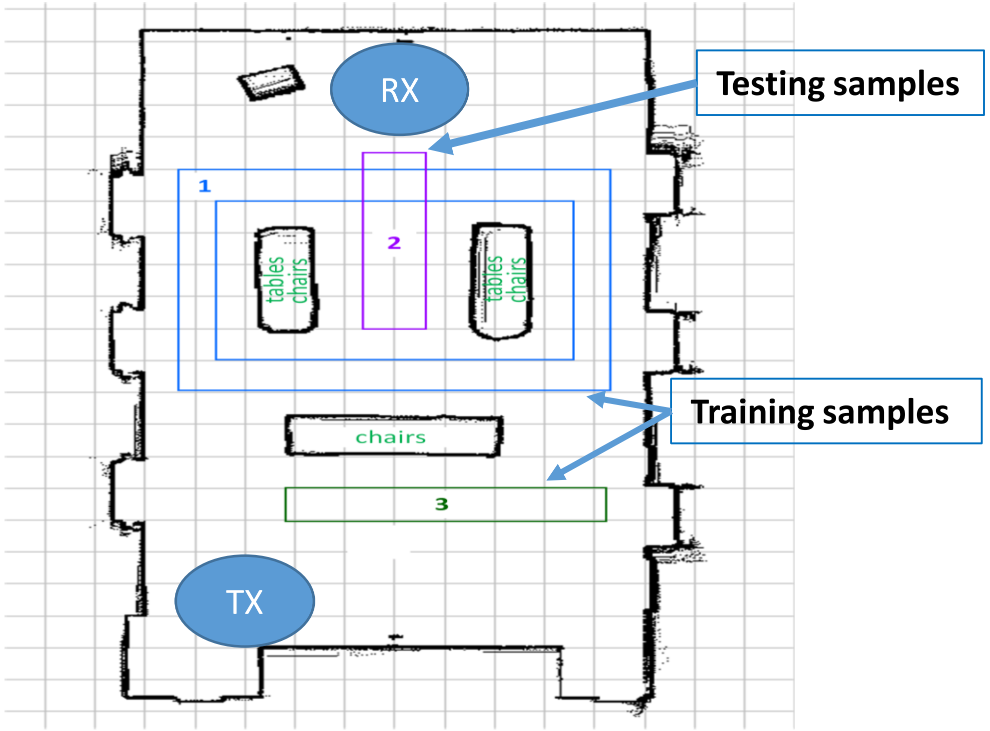
$$L = -\frac{1}{N}\sum_{i=1}^{N} \left(\boldsymbol{w_1} * y_i \log\big(p(y_i)\big) + \boldsymbol{w_0}\ (1 - y_i)\log\big(1 - p(y_i)\big) \right)$$

<u>Dataset:</u>

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

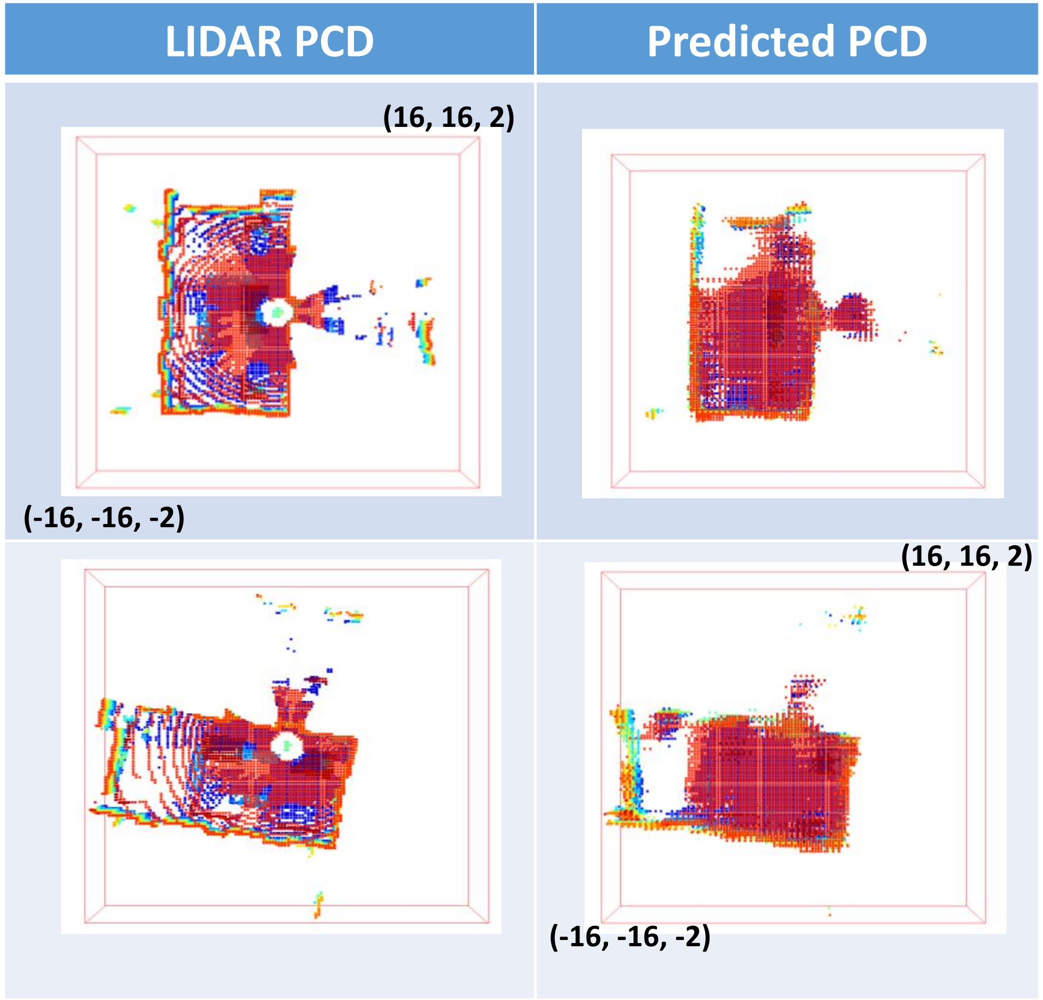
<u>Optimizer</u>:

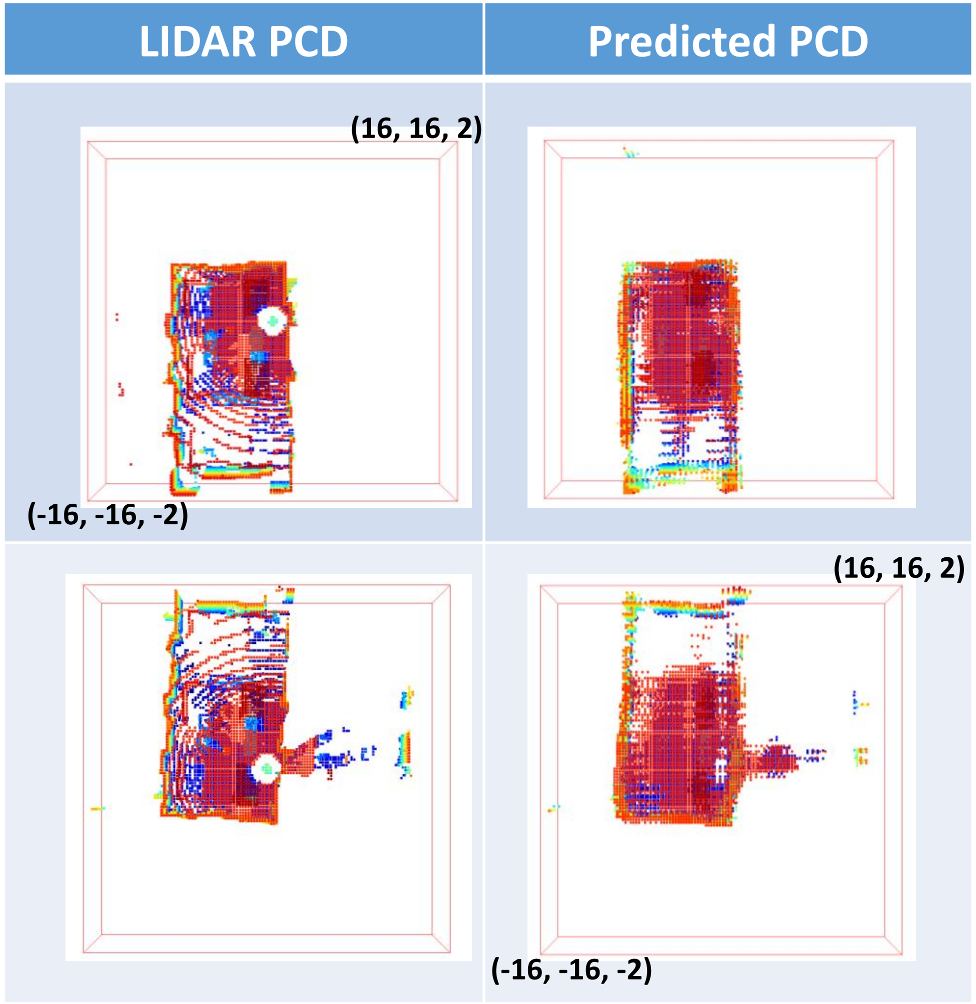
- Adam with learning rate of 0.0005
- Decay rate: 0.9 every 10000 steps
- Epochs =100, batch size : 32.



Prediction vs Actual PCD

SAMSUNG

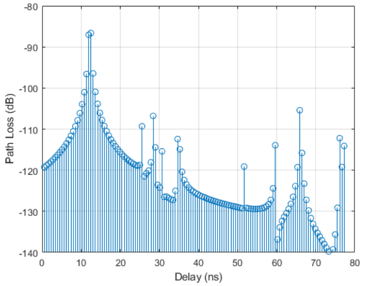


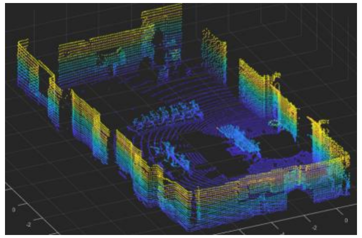


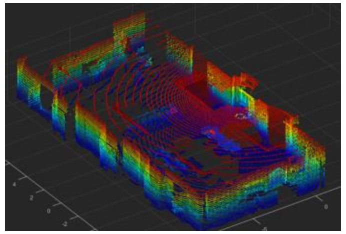
- Estimated **Depth Map similar to LIDAR** PCD.
- **Change in perception** across locations, is nicely **captured**.

Evaluation Metric for the challenge

SAMSUNG







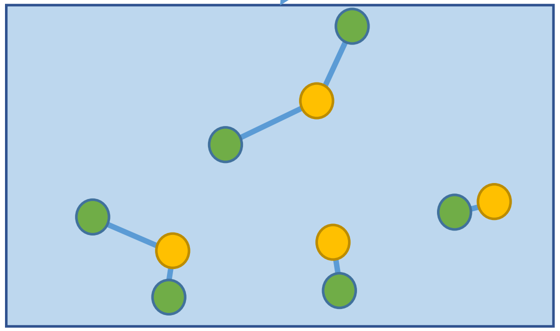
Input Data: RF Data Evaluation Dataset → Trained Model → Output: LiDAR Point Clouds → Evaluation ← Ground Truth: LiDAR Evaluation dataset

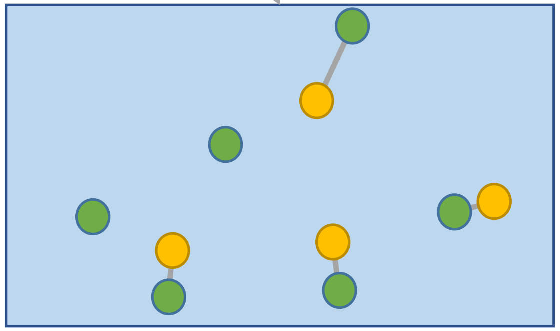
<u>Chamfer Distance</u>
measures discrepancies between point clouds.

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

Predicted PCD: S_1

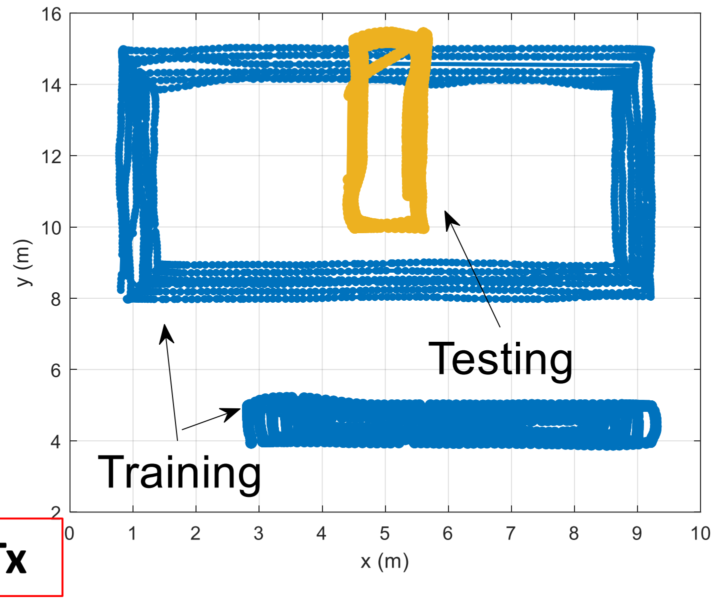
LIDAR PCD: S_2

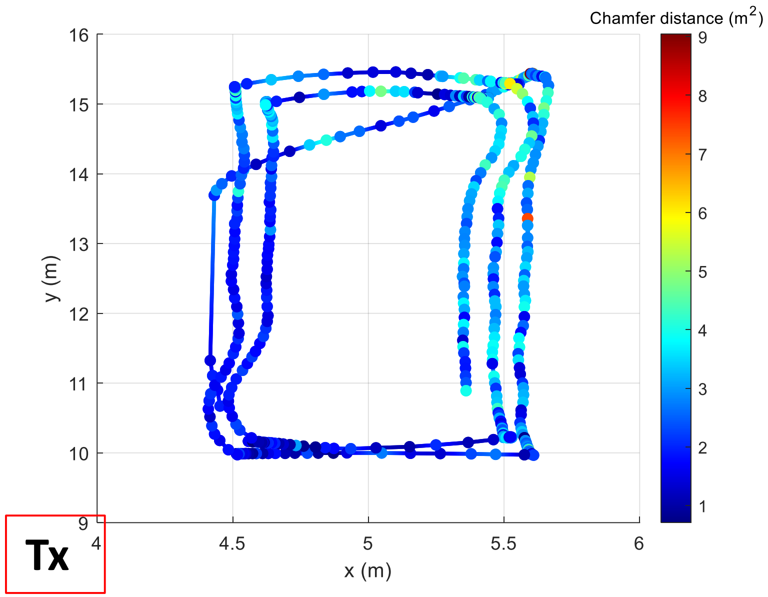


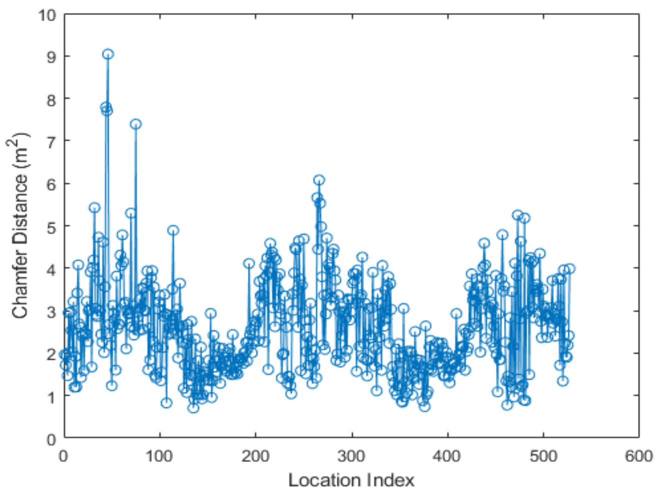


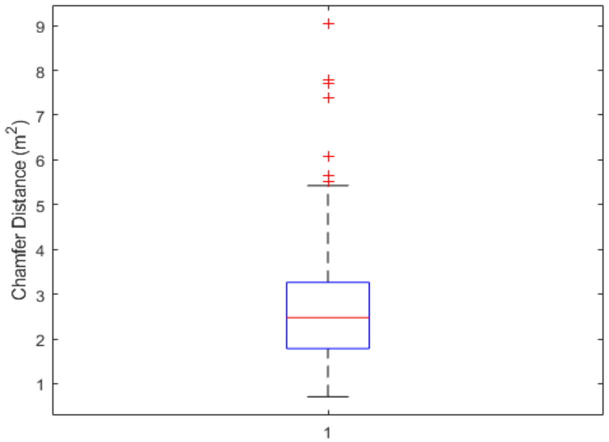
Evaluation : Error Analysis on Testing Data

SAMSUNG









<u>Chamfer distance for Testing data</u>

- 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.

<u>Chamfer distance for Training data</u>

- 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

Conclusion and Future work..

SAMSUNG

Observations:

- Estimated **Depth Map similar to LIDAR** PCD.
- **Change in perception** across locations, is nicely **captured**.
- 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 :

- Using Multi-path components instead of CIR as input.
- Global training inclusive of all the locations
- Experiment with the Voxel grid size
- Train using Tx perspective along with Rx perspective

Thank you.

Questions ?

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