TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Graph Convolution

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Graph Convolution

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Graph Convolution

A : Adjacency Matrix

U : Feature Matrix

W : Weight Matrix/ Filter

F_i : Number of features at the ith layer of network

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

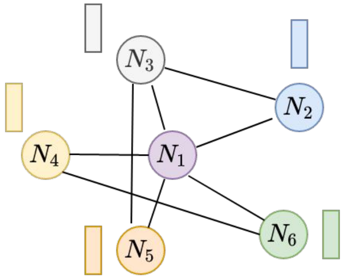
- Graph Convolution

A : Adjacency Matrix

U : Feature Matrix

W : Weight Matrix/ Filter

F_i : Number of features at the ith layer of network



	N_1	N_2	N_3	N_4	N_5	N_6
N_1	0	1	1	1	1	1
N_2	1	0	1	0	0	0
N_3	1	1	0	0	1	0
N_4	1	0	0	0	0	1
N_5	1	0	1	0	0	0
N_6	1	0	0	1	0	0

A

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

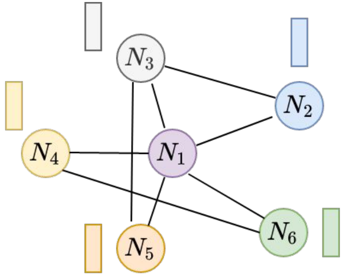
- Graph Convolution

A : Adjacency Matrix

U : Feature Matrix

W : Weight Matrix/ Filter

Fi : Number of features at the ith layer of network



	N_1	N_2	N_3	N_4	N_5	N_6
N_1	0	1	1	1	1	1
N_2	1	0	1	0	0	0
N_3	1	1	0	0	1	0
N_4	1	0	0	0	0	1
N_5	1	0	1	0	0	0
N_6	1	0	0	1	0	0

A

N_1	x_1	y_1	z_1
N_2	x_2	y_2	z_2
N_3	x_3	y_3	z_3
N_4	x_4	y_4	z_4
N_5	x_5	y_5	z_5
N_6	x_6	y_6	z_6

U

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

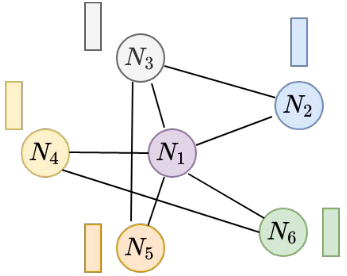
- Graph Convolution

A : Adjacency Matrix

U : Feature Matrix

W : Weight Matrix/ Filter

Fi : Number of features at the ith layer of network



	N_1	N_2	N_3	N_4	N_5	N_6
N_1	0	1	1	1	1	1
N_2	1	0	1	0	0	0
N_3	1	1	0	0	1	0
N_4	1	0	0	0	0	1
N_5	1	0	1	0	0	0
N_6	1	0	0	1	0	0

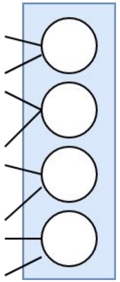
A

N_1	x_1	y_1	z_1
N_2	x_2	y_2	z_2
N_3	x_3	y_3	z_3
N_4	x_4	y_4	z_4
N_5	x_5	y_5	z_5
N_6	x_6	y_6	z_6

U

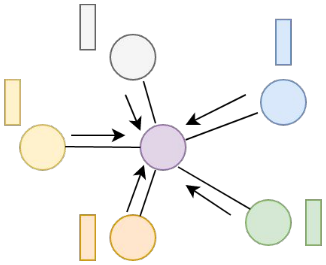
w_1	w_2	w_3	w_4
w_5	w_6	w_7	w_8
w_9	w_{10}	w_{11}	w_{12}

W



TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Graph Convolution

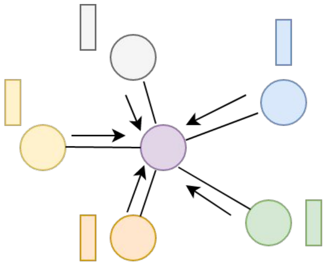


Feature Gathering

$$A_{N\times N} \times U_{N\times F^i}$$

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Graph Convolution



Feature Gathering

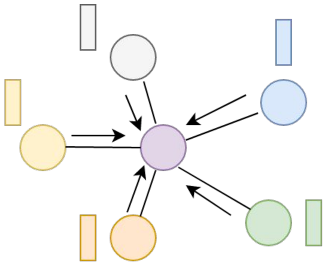
$$A_{N\times N} \times U_{N\times F^i}$$

	N_1	N_2	N_3	N_4	N_5	N_6
N_1	0	1	1	1	1	1
N_2	1	0	1	0	0	0
N_3	1	1	0	0	1	0
N_4	1	0	0	0	0	1
N_5	1	0	1	0	0	0
N_6	1	0	0	1	0	0

N_1	x_1	y_1	z_1
N_2	x_2	y_2	z_2
N_3	x_3	y_3	z_3
N_4	x_4	y_4	z_4
N_5	x_5	y_5	z_5
N_6	x_6	y_6	z_6

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Graph Convolution

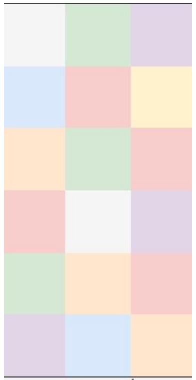


Feature Gathering

$$A_{N\times N} \times U_{N\times F^i}$$

	N_1	N_2	N_3	N_4	N_5	N_6
N_1	0	1	1	1	1	1
N_2	1	0	1	0	0	0
N_3	1	1	0	0	1	0
N_4	1	0	0	0	0	1
N_5	1	0	1	0	0	0
N_6	1	0	0	1	0	0

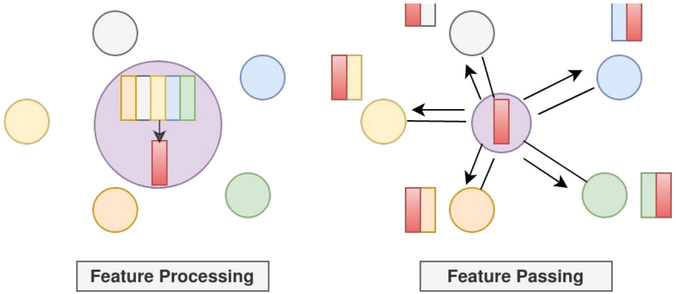
N_1	x_1	y_1	z_1
N_2	x_2	y_2	z_2
N_3	x_3	y_3	z_3
N_4	x_4	y_4	z_4
N_5	x_5	y_5	z_5
N_6	x_6	y_6	z_6



$N \times F^i$

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

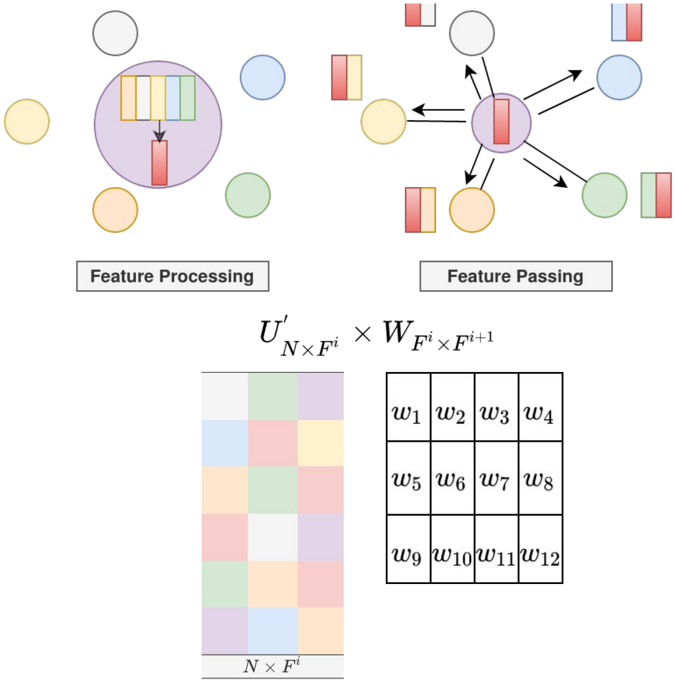
- Graph Convolution



$$U'_{N\times F^i} \times W_{F^i\times F^{i+1}}$$

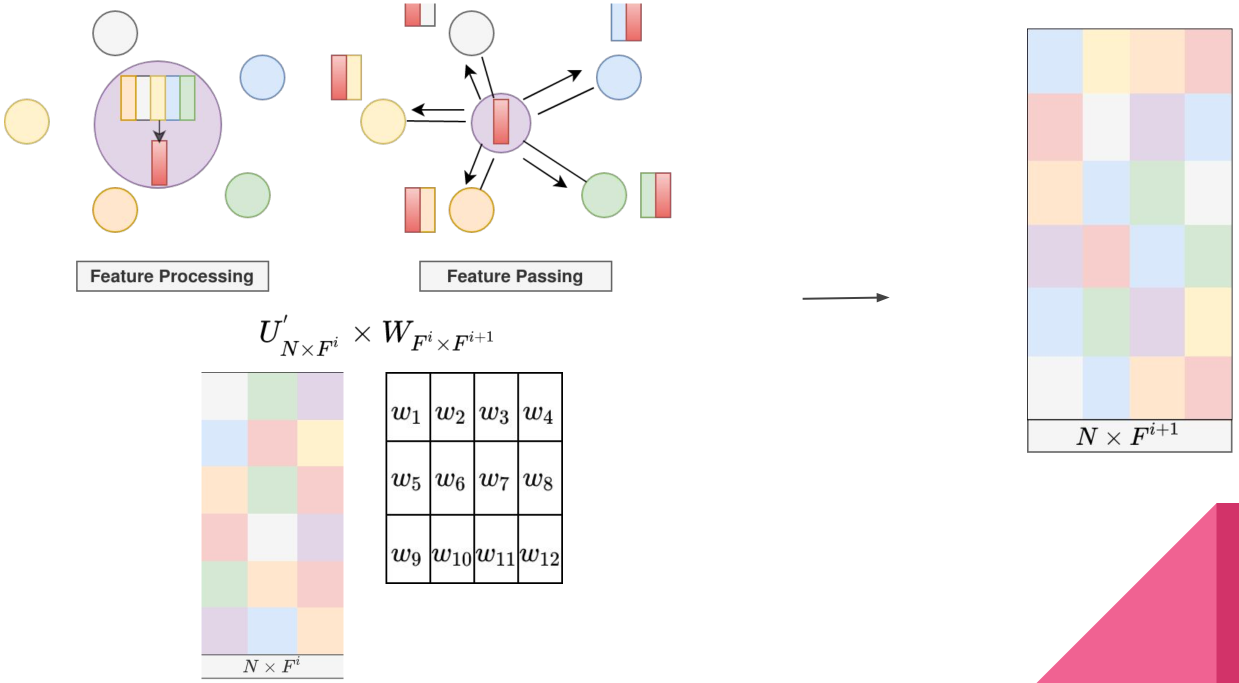
TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Graph Convolution



TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Graph Convolution



TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Graph Convolution

$$F^{i+1} = \sigma(A_{N \times N} U_{N \times F^i} W^i_{F^i \times F^{i+1}})$$

Here $\boldsymbol{\sigma}$ is non-linearity in the network. It can be relu, leaky-relu, sigmoid e.t.c. For our work we have considered leaky-relu function.

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Chamfer Distance
  - It measures the distance between two point set.
  - In our case point set contains distinct tuple values which represent (x, y, z) coordinates of 3D point cloud shape. Ex: S1 = {(xi, yi, zi)} and S2 = {(x'j, y'j, z'j)}

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Chamfer Distance
  - It measures the distance between two point set.
  - In our case point set contains distinct tuple values which represent (x, y, z) coordinates of 3D point cloud shape. Ex: S1 = {(xi, yi, zi)} and S2 = {(x’j, y’j, z’j)}

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Chamfer Distance
  - It measures the distance between two point set.
  - In our case point set contains distinct tuple values which represent (x, y, z) coordinates of 3D point cloud shape. Ex: S1 = {(xi, yi, zi)} and S2 = {(x'j, y'j, z'j)}

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Chamfer Distance
  - It measures the distance between two point set.
  - In our case point set contains distinct tuple values which represent (x, y, z) coordinates of 3D point cloud shape. Ex: S1 = {(xi, yi, zi)} and S2 = {(x'j, y'j, z'j)}

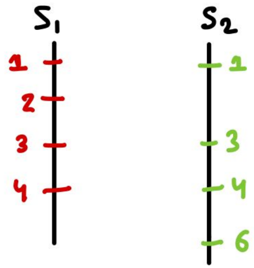
$$\mathcal{L}_{chamfer}(S_1, S_2) = \sum_{x \in S_1} min_{y \in S_2} ||x - y||_2^2 + \sum_{y \in S_2} min_{x \in S_1} ||x - y||_2^2$$

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Chamfer Distance
  - It measures the distance between two point set.
  - In our case point set contains distinct tuple values which represent (x, y, z) coordinates of 3D point cloud shape. Ex: S1 = {(xi, yi, zi)} and S2 = {(x'j, y'j, z'j)}

$$\mathcal{L}_{chamfer}(S_1, S_2) = \sum_{x \in S_1} min_{y \in S_2} ||x - y||_2^2 + \sum_{y \in S_2} min_{x \in S_1} ||x - y||_2^2$$

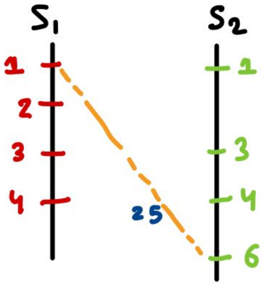
Ex: S1 = {1, 2, 3, 4} and S2 = {3, 4, 1, 6}



TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Chamfer Distance

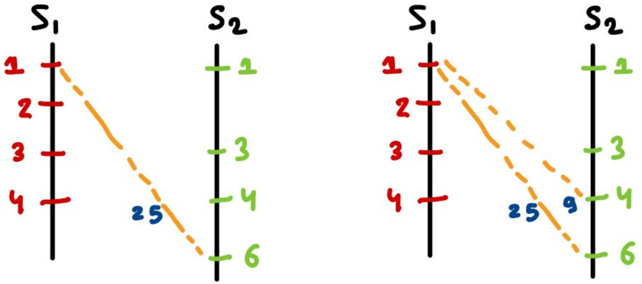
$$\mathcal{L}_{chamfer}(S_1, S_2) = \sum_{x \in S_1} min_{y \in S_2} ||x - y||_2^2 + \sum_{y \in S_2} min_{x \in S_1} ||x - y||_2^2$$



TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Chamfer Distance

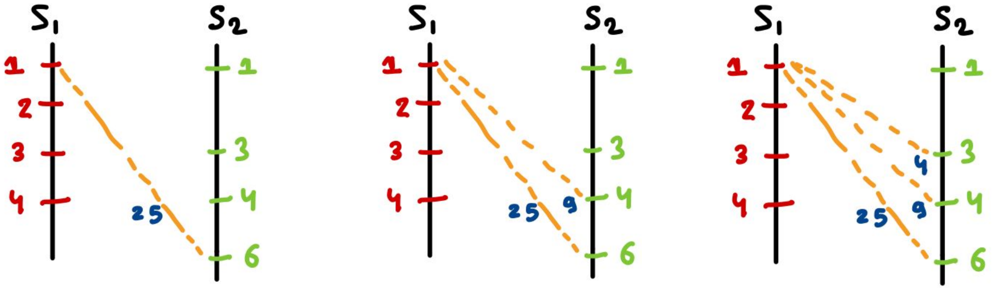
$$\mathcal{L}_{chamfer}(S_1, S_2) = \sum_{x \in S_1} min_{y \in S_2} ||x - y||_2^2 + \sum_{y \in S_2} min_{x \in S_1} ||x - y||_2^2$$



TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Chamfer Distance

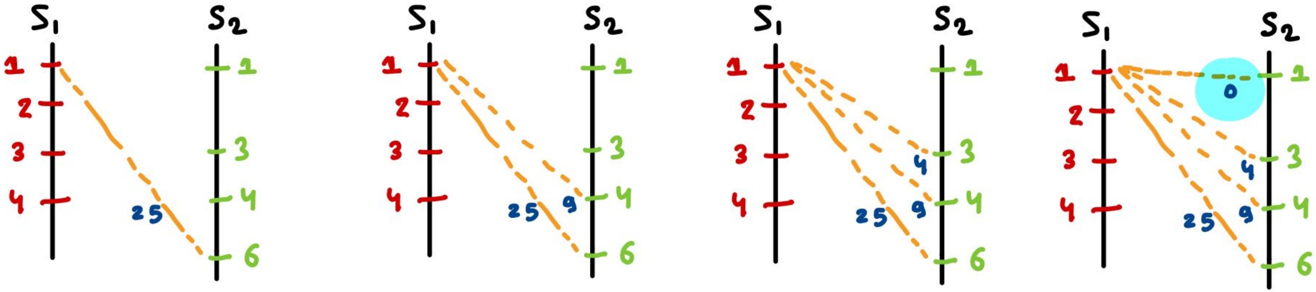
$$\mathcal{L}_{chamfer}(S_1, S_2) = \sum_{x \in S_1} min_{y \in S_2} ||x - y||_2^2 + \sum_{y \in S_2} min_{x \in S_1} ||x - y||_2^2$$



TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Chamfer Distance

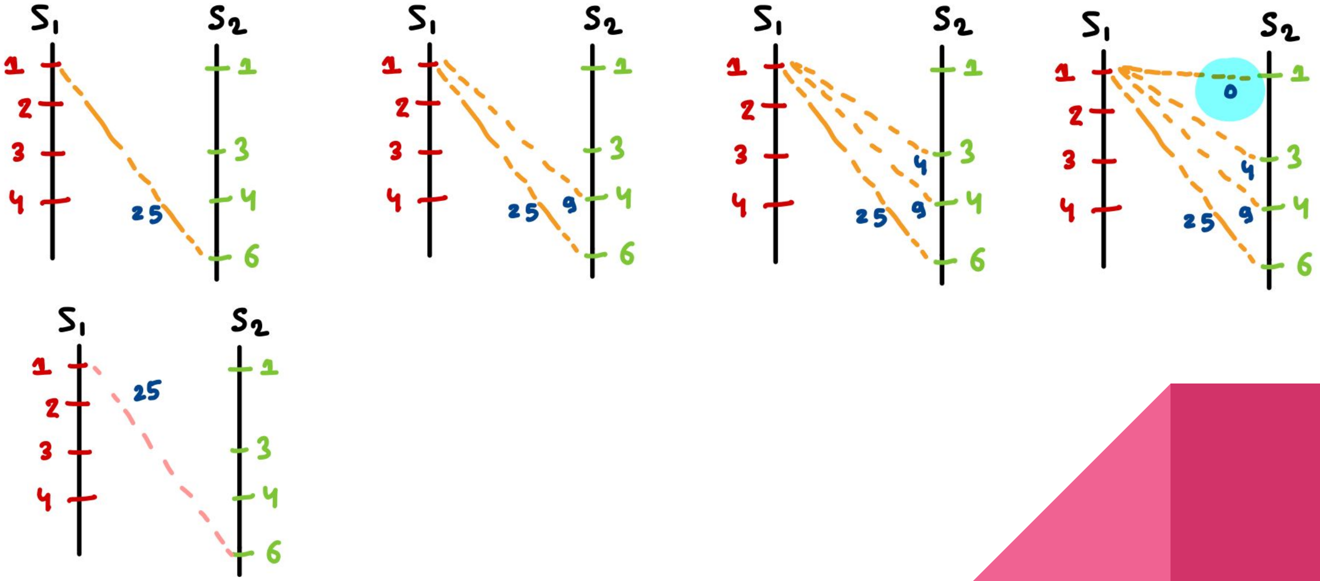
$$\mathcal{L}_{chamfer}(S_1, S_2) = \sum_{x \in S_1} min_{y \in S_2} ||x - y||_2^2 + \sum_{y \in S_2} min_{x \in S_1} ||x - y||_2^2$$



TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Chamfer Distance

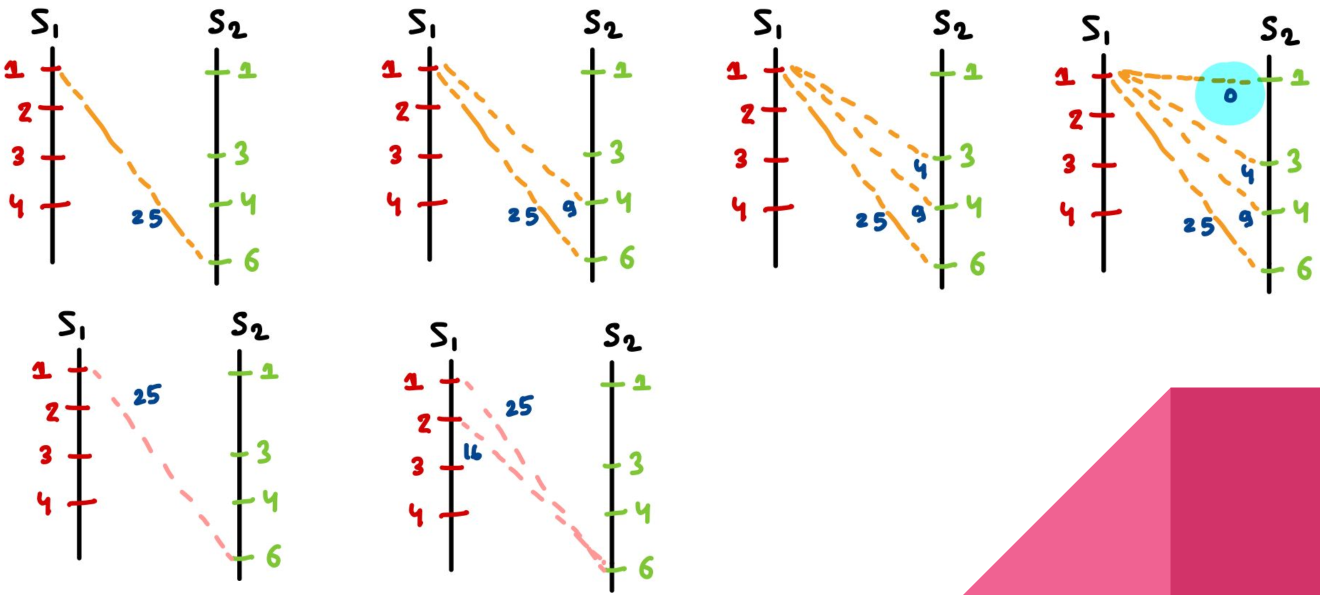
$$\mathcal{L}_{chamfer}(S_1, S_2) = \sum_{x \in S_1} min_{y \in S_2} ||x - y||_2^2 + \sum_{y \in S_2} min_{x \in S_1} ||x - y||_2^2$$



TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Chamfer Distance

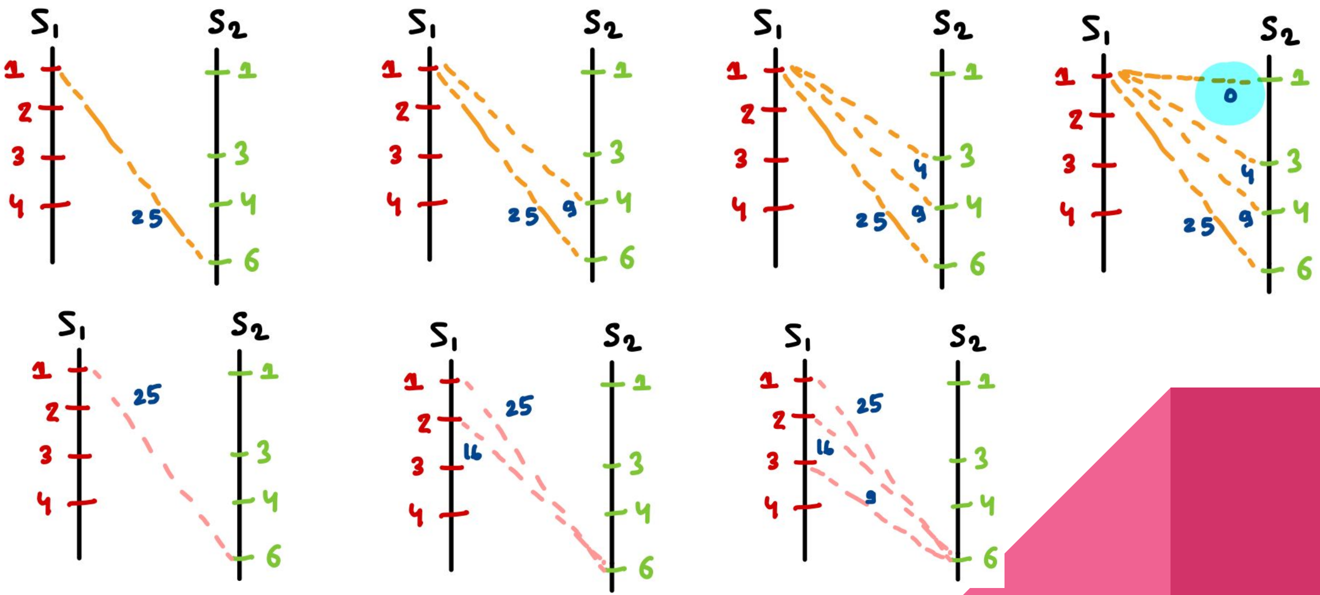
$$\mathcal{L}_{chamfer}(S_1, S_2) = \sum_{x \in S_1} min_{y \in S_2} ||x - y||_2^2 + \sum_{y \in S_2} min_{x \in S_1} ||x - y||_2^2$$



TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Chamfer Distance

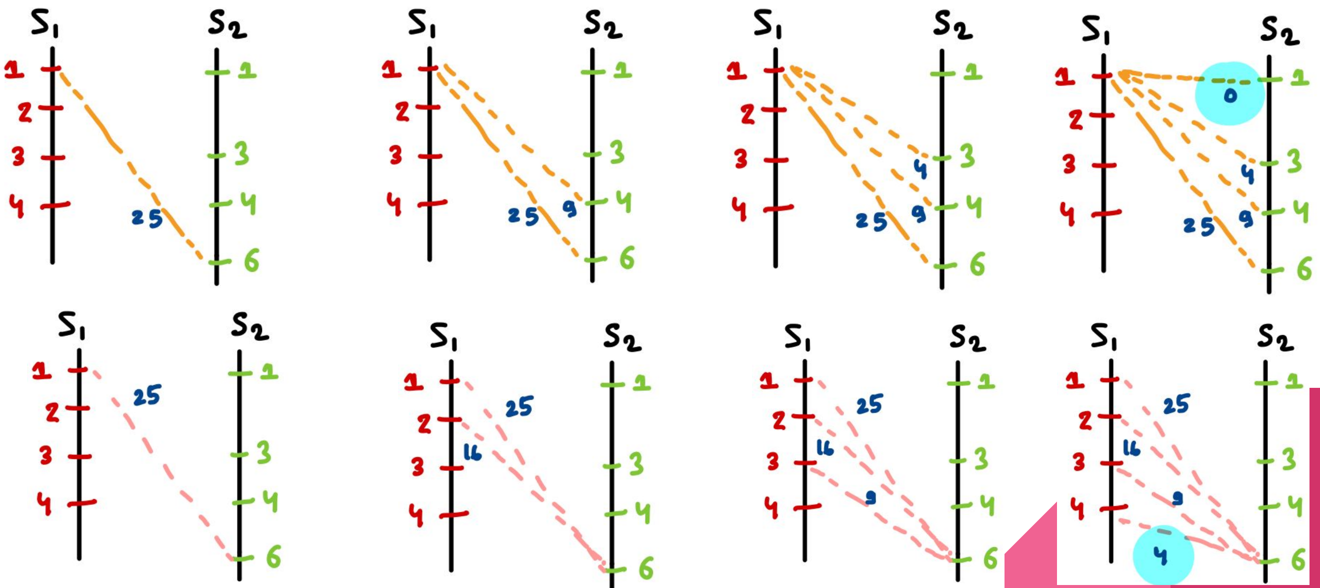
$$\mathcal{L}_{chamfer}(S_1, S_2) = \sum_{x \in S_1} min_{y \in S_2} ||x - y||_2^2 + \sum_{y \in S_2} min_{x \in S_1} ||x - y||_2^2$$



TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Chamfer Distance

$$\mathcal{L}_{chamfer}(S_1, S_2) = \sum_{x \in S_1} min_{y \in S_2} ||x - y||_2^2 + \sum_{y \in S_2} min_{x \in S_1} ||x - y||_2^2$$



TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Chamfer Distance

$$\mathcal{L}_{chamfer}(S_1, S_2) = \sum_{x \in S_1} min_{y \in S_2} ||x - y||_2^2 + \sum_{y \in S_2} min_{x \in S_1} ||x - y||_2^2$$

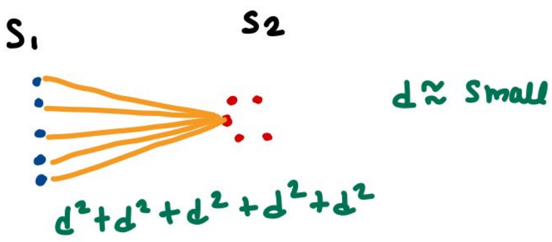
  - In case of 3D coordinates

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Chamfer Distance

$$\mathcal{L}_{chamfer}(S_1, S_2) = \sum_{x \in S_1} min_{y \in S_2} ||x - y||_2^2 + \sum_{y \in S_2} min_{x \in S_1} ||x - y||_2^2$$

  - In case of 3D coordinates

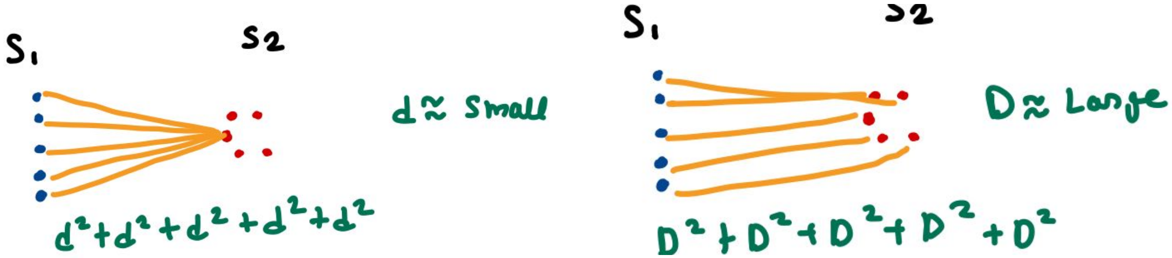


TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Chamfer Distance

$$\mathcal{L}_{chamfer}(S_1, S_2) = \sum_{x \in S_1} min_{y \in S_2} ||x-y||_2^2 + \sum_{y \in S_2} min_{x \in S_1} ||x-y||_2^2$$

  - In case of 3D coordinates



TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Frechet Inception Distance [3] and Frechet Point Cloud [1]
  - FID is use to evaluate the quality of target image with respect to reference image.
  - FPD is use to evaluate the quality of target point cloud with respect to reference point cloud.
  - These both are the metrics for evaluation and comparison of the deep learning models.
  - We use FID in image generation GAN to evaluate the quality of image generated by the GAN model.
  - Idea of FPD is first proposed is TreeGCN-GAN [1] paper.

[1] TreeGCN-GAN [https://arxiv.org/abs/1905.06292] [3] GAN Training Local Nash Equilibrium [https://arxiv.org/abs/1706.08500]

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Frechet Inception Distance [3] and Frechet Point Cloud [1]
  - FID is use to evaluate the quality of target image with respect to reference image.
  - FPD is use to evaluate the quality of target point cloud with respect to reference point cloud.
  - These both are the metrics for evaluation and comparison of the deep learning models.
  - We use FID in image generation GAN to evaluate the quality of image generated by the GAN model.
  - Idea of FPD is first proposed is TreeGCN-GAN [1] paper.

[1] TreeGCN-GAN [https://arxiv.org/abs/1905.06292] [3] GAN Training Local Nash Equilibrium [https://arxiv.org/abs/1706.08500]

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Frechet Inception Distance [3] and Frechet Point Cloud [1]
  - FID is use to evaluate the quality of target image with respect to reference image.
  - FPD is use to evaluate the quality of target point cloud with respect to reference point cloud.
  - These both are the metrics for evaluation and comparison of the deep learning models.
  - We use FID in image generation GAN to evaluate the quality of image generated by the GAN model.
  - Idea of FPD is first proposed is TreeGCN-GAN [1] paper.

[1] TreeGCN-GAN [https://arxiv.org/abs/1905.06292] [3] GAN Training Local Nash Equilibrium [https://arxiv.org/abs/1706.08500]

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Frechet Inception Distance [3] and Frechet Point Cloud [1]
  - FID is use to evaluate the quality of target image with respect to reference image.
  - FPD is use to evaluate the quality of target point cloud with respect to reference point cloud.
  - These both are the metrics for evaluation and comparison of the deep learning models.
  - We use FID in image generation GAN to evaluate the quality of image generated by the GAN model.
  - Idea of FPD is first proposed is TreeGCN-GAN [1] paper.

[1] TreeGCN-GAN [https://arxiv.org/abs/1905.06292] [3] GAN Training Local Nash Equilibrium [https://arxiv.org/abs/1706.08500]

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Frechet Inception Distance [3] and Frechet Point Cloud [1]
  - FID is use to evaluate the quality of target image with respect to reference image.
  - FPD is use to evaluate the quality of target point cloud with respect to reference point cloud.
  - These both are the metrics for evaluation and comparison of the deep learning models.
  - We use FID in image generation GAN to evaluate the quality of image generated by the GAN model.
  - Idea of FPD is first proposed is TreeGCN-GAN [1] paper.

[1] TreeGCN-GAN [https://arxiv.org/abs/1905.06292] [3] GAN Training Local Nash Equilibrium [https://arxiv.org/abs/1706.08500]

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Frechet Inception Distance [3] and Frechet Point Cloud [1]
  - FID is use to evaluate the quality of target image with respect to reference image.
  - FPD is use to evaluate the quality of target point cloud with respect to reference point cloud.
  - These both are the metrics for evaluation and comparison of the deep learning models.
  - We use FID in image generation GAN to evaluate the quality of image generated by the GAN model.
  - Idea of FPD is first proposed is TreeGCN-GAN [1] paper.

[1] TreeGCN-GAN [https://arxiv.org/abs/1905.06292] [3] GAN Training Local Nash Equilibrium [https://arxiv.org/abs/1706.08500]

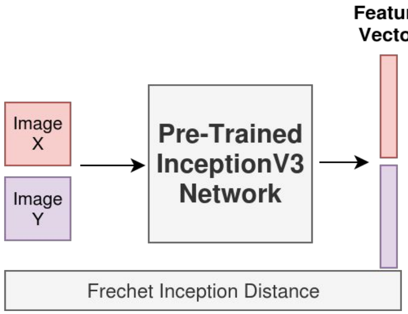
TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Frechet Inception Distance [3] and Frechet Point Cloud [1]
  - General Idea is

[1] TreeGCN-GAN [https://arxiv.org/abs/1905.06292] [3] GAN Training Local Nash Equilibrium [https://arxiv.org/abs/1706.08500]

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

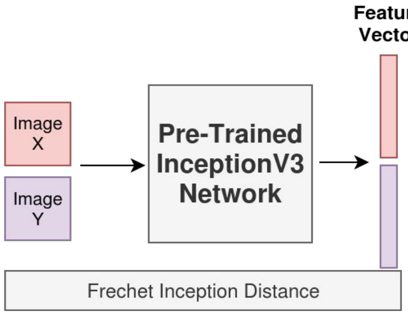
- Frechet Inception Distance [3] and Frechet Point Cloud [1]
  - General Idea is

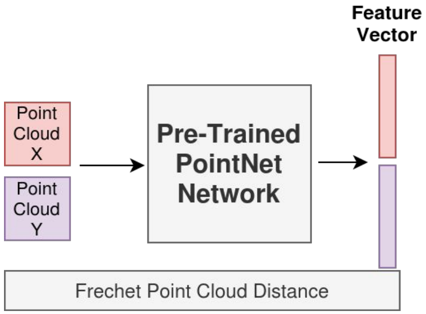


[1] TreeGCN-GAN [https://arxiv.org/abs/1905.06292] [3] GAN Training Local Nash Equilibrium [https://arxiv.org/abs/1706.08500]

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Frechet Inception Distance [3] and Frechet Point Cloud [1]
  - General Idea is

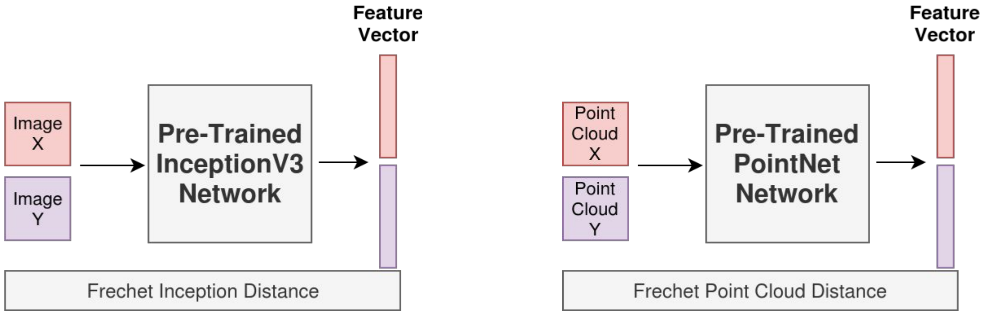




[1] TreeGCN-GAN [https://arxiv.org/abs/1905.06292] [3] GAN Training Local Nash Equilibrium [https://arxiv.org/abs/1706.08500]

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Frechet Inception Distance [3] and Frechet Point Cloud [1]
  - General Idea is

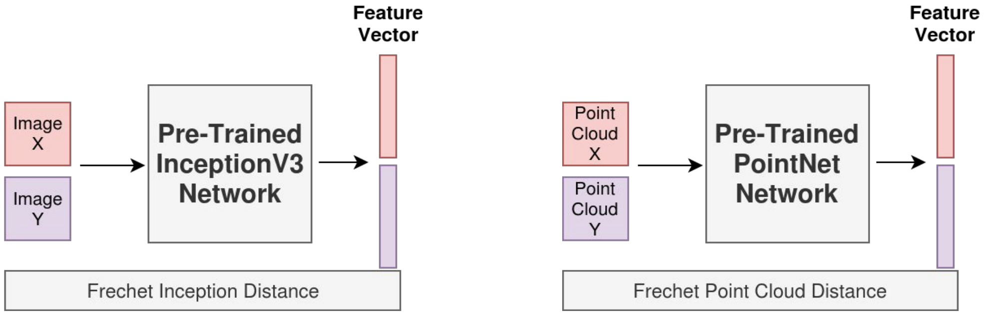


$$FD = ||\mu_X - \mu_Y||^2 - Tr(\Sigma_X + \Sigma_Y - 2\sqrt{\Sigma_X \Sigma_Y})$$

  - Lower the Frechet Distance better the generated result is.

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Frechet Inception Distance [3] and Frechet Point Cloud [1]
  - General Idea is

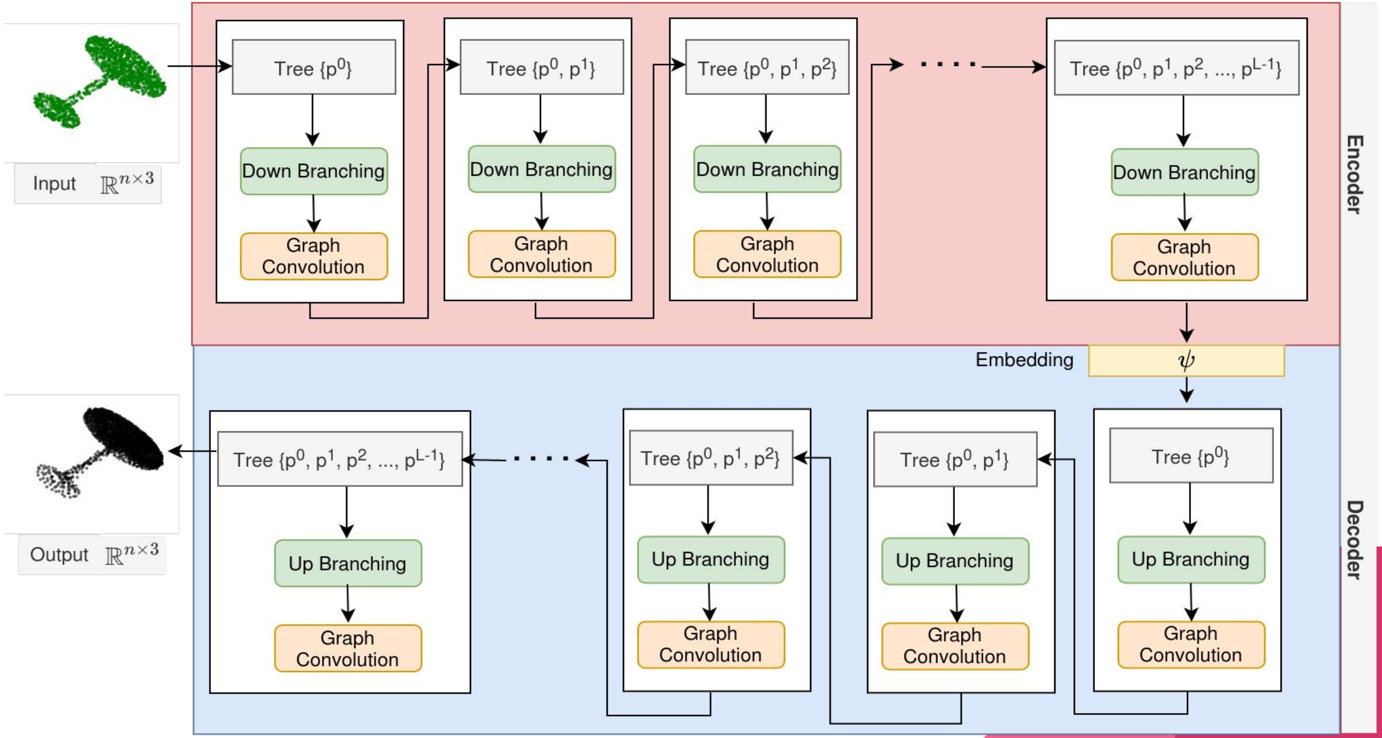


$$FD = ||\mu_X - \mu_Y||^2 - Tr(\Sigma_X + \Sigma_Y - 2\sqrt{\Sigma_X \Sigma_Y})$$

  - Lower the Frechet Distance better the generated result is.

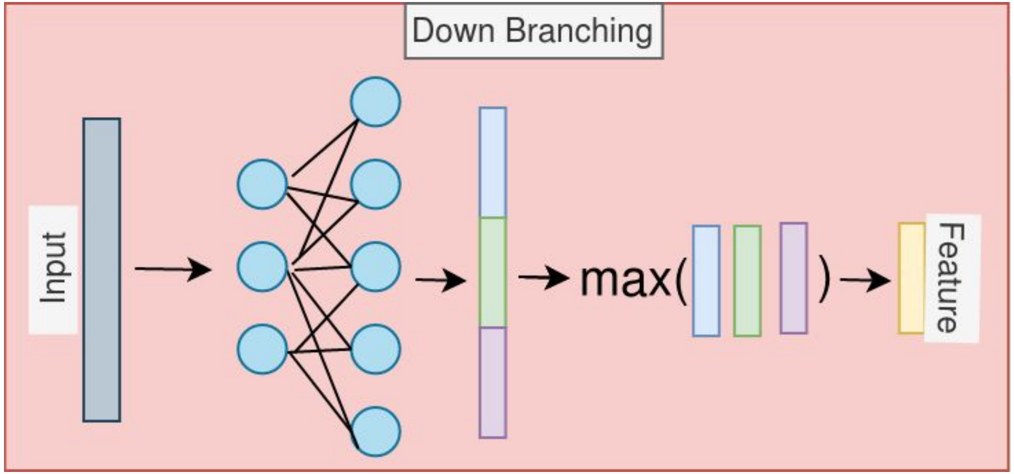
TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Proposed TreeGCN-ED Model



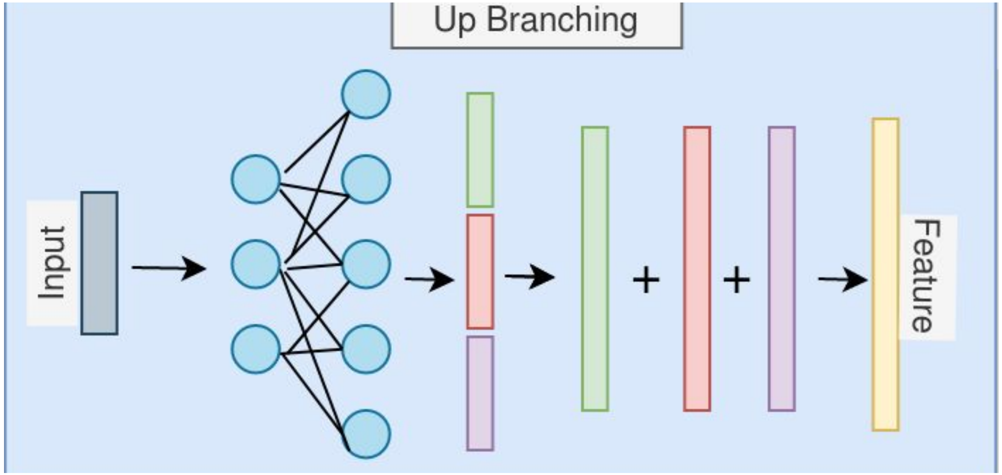
TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Proposed TreeGCN-ED Model
  - Down Branching



TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Proposed TreeGCN-ED Model
  - Up Branching



TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Loss Function
    - Chamfer distance function is used to train complete TreeGCN-ED model.
- Dataset
    - ShapenetBenchmarkV0 [5]
    - Specifically we trained our model on 2 classes [chair and table]

[5] Shapnet Dataset [https://arxiv.org/abs/1512.03012]

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Loss Function
  - Chamfer distance function is used to train complete TreeGCN-ED model.

$$\mathcal{L}_{chamfer}(S_1, S_2) = \sum_{x \in S_1} min_{y \in S_2} ||x - y||_2^2 + \sum_{y \in S_2} min_{x \in S_1} ||x - y||_2^2$$

- Dataset
  - ShapenetBenchmarkV0 [5]
  - Specifically we trained our model on 2 classes [chair and table]

[5] Shapnet Dataset [https://arxiv.org/abs/1512.03012]

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Loss Function
  - Chamfer distance function is used to train complete TreeGCN-ED model.

$$\mathcal{L}_{chamfer}(S_1, S_2) = \sum_{x \in S_1} min_{y \in S_2} ||x - y||_2^2 + \sum_{y \in S_2} min_{x \in S_1} ||x - y||_2^2$$

- Dataset
  - ShapenetBenchmarkV0 [5]
  - Specifically we trained our model on 2 classes [chair and table]

[5] Shapnet Dataset [https://arxiv.org/abs/1512.03012]

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Loss Function
  - Chamfer distance function is used to train complete TreeGCN-ED model.

$$\mathcal{L}_{chamfer}(S_1, S_2) = \sum_{x \in S_1} min_{y \in S_2} ||x - y||_2^2 + \sum_{y \in S_2} min_{x \in S_1} ||x - y||_2^2$$

- Dataset
  - ShapenetBenchmarkV0 [5]
  - Specifically we trained our model on 2 classes [chair and table]

[5] Shapnet Dataset [https://arxiv.org/abs/1512.03012]

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

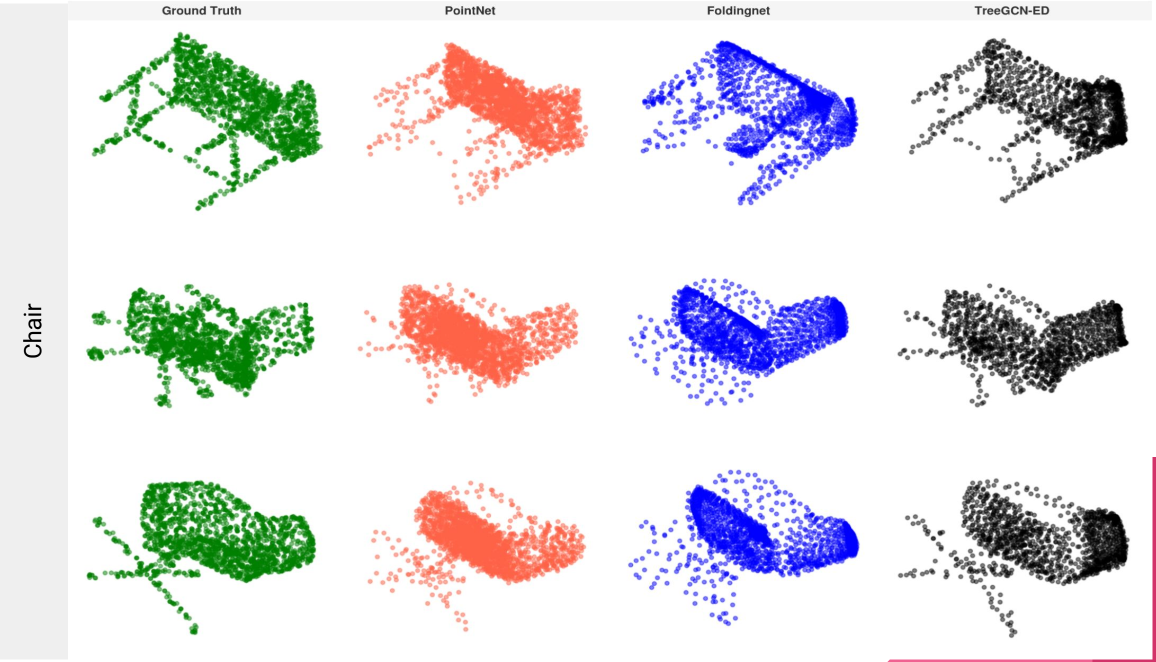
- Result and Comparison
  - Quantitative Results

Class	Models	CD	FPD
Table	PointNet-ED [6]	20.89	11849.31
	FoldingNet [7]	0.93	26.04
	TreeGCN-ED	**0.70**	**11.77**
Chair	PointNet-ED [6]	12.25	3810.69
	FoldingNet [7]	0.99	17.64
	TreeGCN-ED	**0.76**	**7.60**

[6] Pointnet [https://arxiv.org/abs/1612.00593] [7] FoldingNet [https://arxiv.org/abs/1712.07262]

TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Result and Comparison
  - Qualitative Results



TreeGCN-ED : A Point Cloud Encoder-Decoder Network

- Result and Comparison
  - Qualitative Results