Point2Point : A Generative Neural Network for Spatio-Temporal Occupancy Prediction from Point Clouds

Athrva Pandhare

Goals

1. Develop a Simple, Efficient Neural Network for Point Cloud generation.
2. Should be able to learn geometric information and correlations between points.
3. Should be parameter efficient.
4. Should work on both, scene point clouds and object point clouds.

Learning on Point Clouds

Problems

1. **Un-orderedness :** Point clouds are just a set of unordered (XYZ) points, the order in which the points are stored does not change the representation of the scene.

2. **Irregularity :** meaning, that points are not evenly sampled across the different regions of an object/scene, so some regions could have dense points while others sparse points.

Solutions

1. **Impose Permutation Invariance in Neural Network or Impose a "Locality" preserving ordering on Point Clouds.**

2. **Implement a multiscale feature learning mechanism. Typically, requires a high receptive field.**

How to learn on Point Clouds?

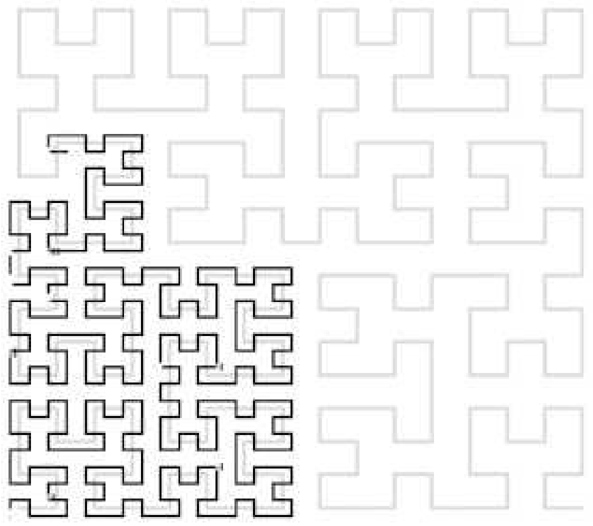
1. **Converting to a Structured Representation :** Most applications typically convert Point Clouds to structured representations like voxel grids or range images.
2. **Multi-view Methods :** Converting 3D point clouds to *N* 2D views. (Typically have better performance than their Voxel Grid based counterparts).
3. **Higher Dimensional Lattice based Representation :** SplatNet converts Point Clouds to a 6D “premutohedral” lattice.
4. **Direct Learning on Raw Point Clouds* :** These learning frameworks operate directly on raw Point cloud data, which is a (Nx3) matrix.

Eliminating the Permutation Invariance problem : Imposing an Ordering on Point Clouds

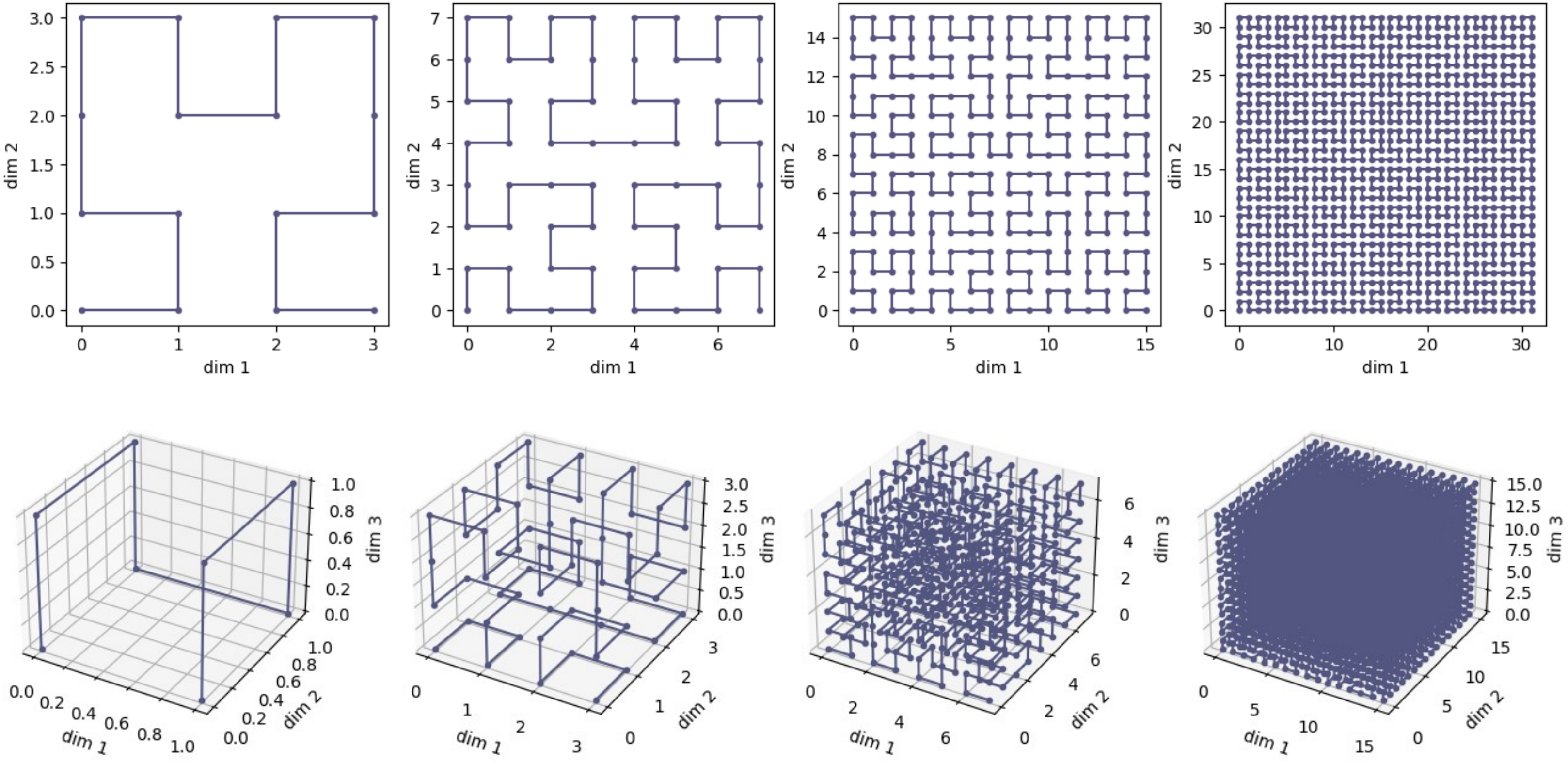
Space Filling Curves : Used in applications where a traversal/scan of a multi-dimensional grid is required.

Locality *: Traversal reflects proximity between points in $[N]^m$, meaning that points close in $[N]^m$ are also close in traversal order.*

Hilbert Curve (right) : *Construction of the HIlbert Curve upto 7th level. HIlbert Curves have excellent locality preserving properties.*



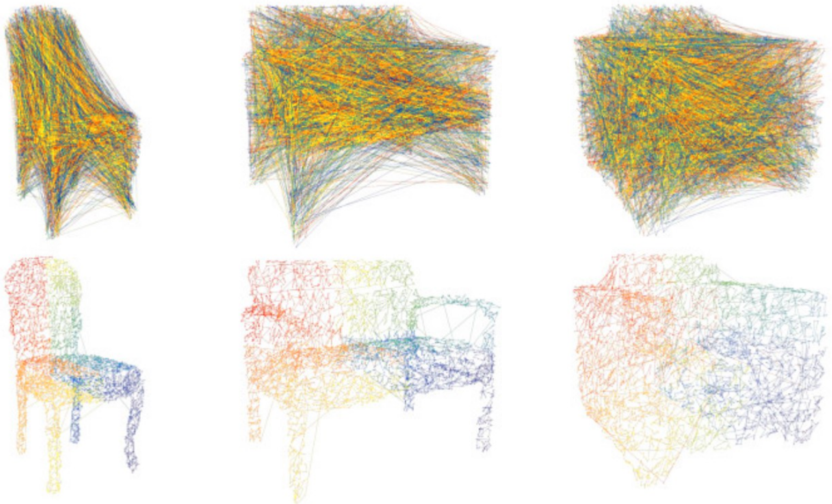
Hilbert Curves in 2d & 3D



An Example of Hilbert Sorted Point Cloud

Unordered Point Cloud

Sorted Point Cloud (Hilbert Order)



What Loss function is best?

Chamfer Distance

been adopted for many tasks. There are some variants of Chamfer discrepancy, which we provide here for completeness. For any two point clouds P, Q, a common formulation of the Chamfer discrepancy between P and Q is given by:

$$d_{CD}(P,Q) = \frac{1}{|P|}\sum_{x\in P}\min_{y\in Q}\|x-y\|_2^2 + \frac{1}{|Q|}\sum_{y\in Q}\min_{x\in P}\|x-y\|_2^2. \quad (1)$$

A slightly modified version of Chamfer divergence is also used by previous works [52, 10, 12, 3] that replaces the sum

Wasserstein Distance (EMD distance)

Q, throughout this paper, we denote its measure representation as follows: $P = \frac{1}{|P|}\sum_{x\in P}\delta_x$ and $Q = \frac{1}{|Q|}\sum_{y\in Q}\delta_y$ where δ_x denotes the Dirac delta distribution at point x in the point cloud P. When $|P| = |Q|$, the Earth Mover's distance [16, 1, 53] between P and Q is defined as

$$d_{EMD}(P,Q) = \min_{T:P\to Q}\sum_{x\in P}\|x - T(x)\|_2. \quad (4)$$

While earlier works [16, 1] showed that EMD is better than Chamfer in 3D point clouds reconstruction task, the computation of EMD can be very expensive compared to the

Chamfer Distance

Wasserstein Distance

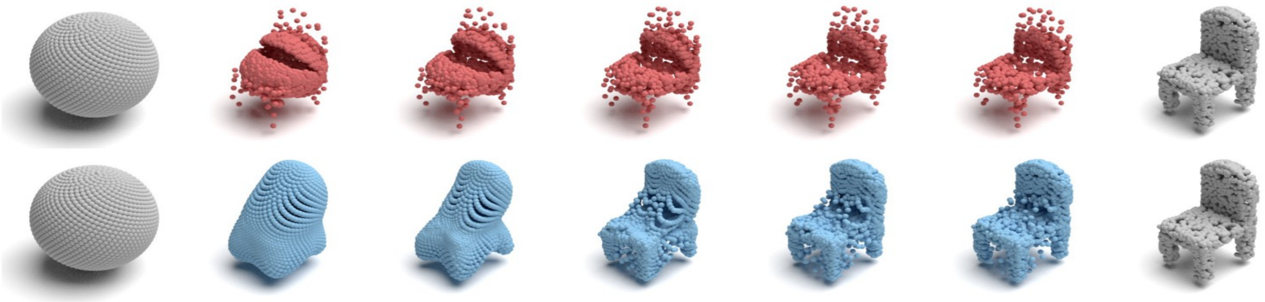


Figure 1: We advocate the use of sliced Wasserstein distance for training 3D point cloud autoencoders. In this example, we try to morph a sphere into a chair by optimizing two different loss functions: Chamfer discrepancy (top, red) and sliced

What Loss function is best?

Optimal Transport Problem with Entropic Regularization

riginal transport problem (2.11):

$$L_{\mathbf{C}}^{\varepsilon}(\mathbf{a},\mathbf{b}) \stackrel{\text{def.}}{=} \min_{\mathbf{P}\in\mathbf{U}(\mathbf{a},\mathbf{b})} \langle \mathbf{P},\ \mathbf{C}\rangle - \varepsilon\mathbf{H}(\mathbf{P}).$$

e is an ε-strongly convex function, Problem (4.2)

Where,

y of a coupling matrix is defined as

$$\mathbf{H}(\mathbf{P}) \stackrel{\text{def.}}{=} -\sum_{i,j} \mathbf{P}_{i,j}(\log(\mathbf{P}_{i,j}) - 1),$$

lefinition for vectors, with the convention tha

P = Coupling Matrix
C = Cost Matrix

Algorithm 1 Sinkhorn-Knopp Algorithm (SK).

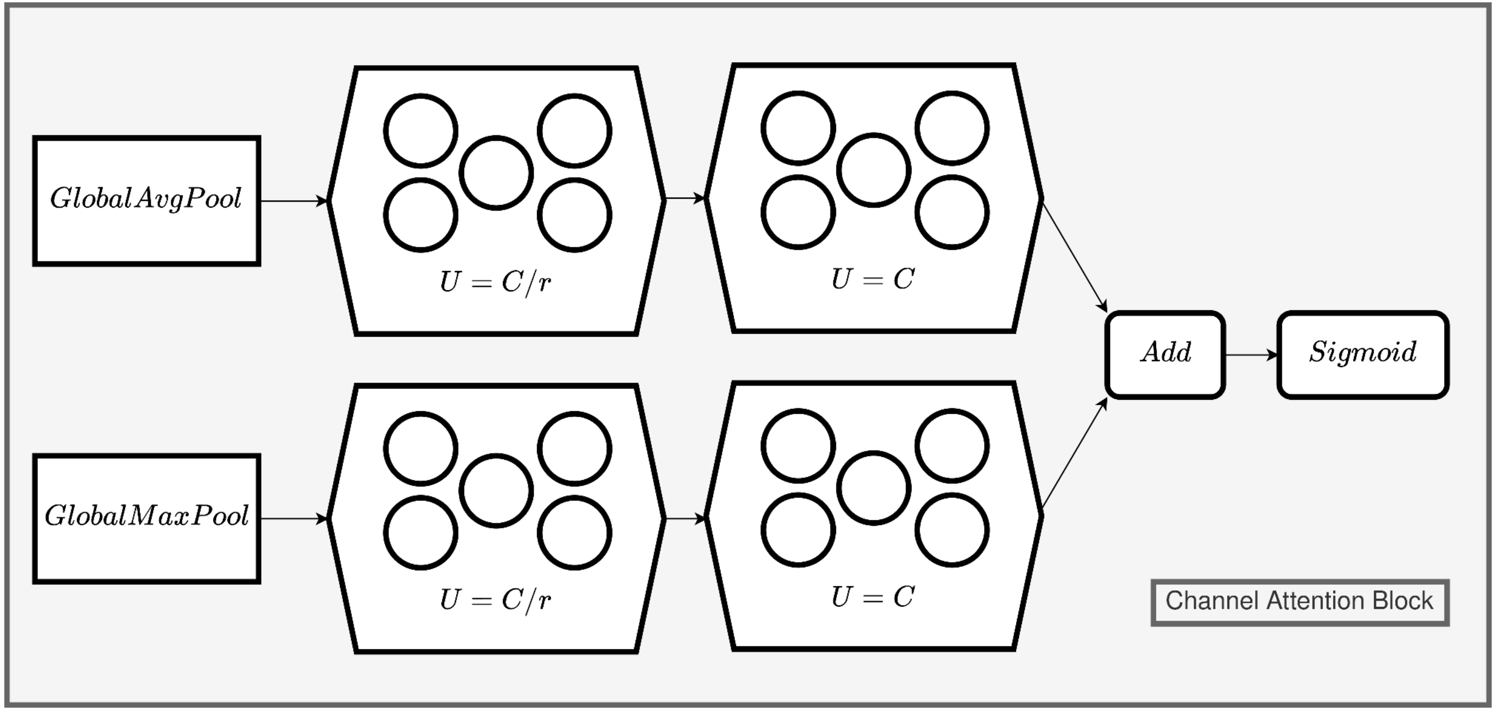
Require: $\mathbf{a}, \mathbf{b}, \mathbf{C}, \lambda$
$\mathbf{u}^{(0)} = \mathbf{1}, \mathbf{K} = \exp(-\mathbf{C}/\lambda)$
for i in $1, \ldots, n_{it}$ **do**
$\mathbf{v}^{(i)} = \mathbf{b} \oslash \mathbf{K}^{\top}\mathbf{u}^{(i-1)}$ // Update right scaling
$\mathbf{u}^{(i)} = \mathbf{a} \oslash \mathbf{K}\mathbf{v}^{(i)}$ // Update left scaling
end for
return $\mathbf{T} = \operatorname{diag}(\mathbf{u}^{(n_{it})})\mathbf{K}\operatorname{diag}(\mathbf{v}^{(n_{it})})$

Note : P := T and λ := ε

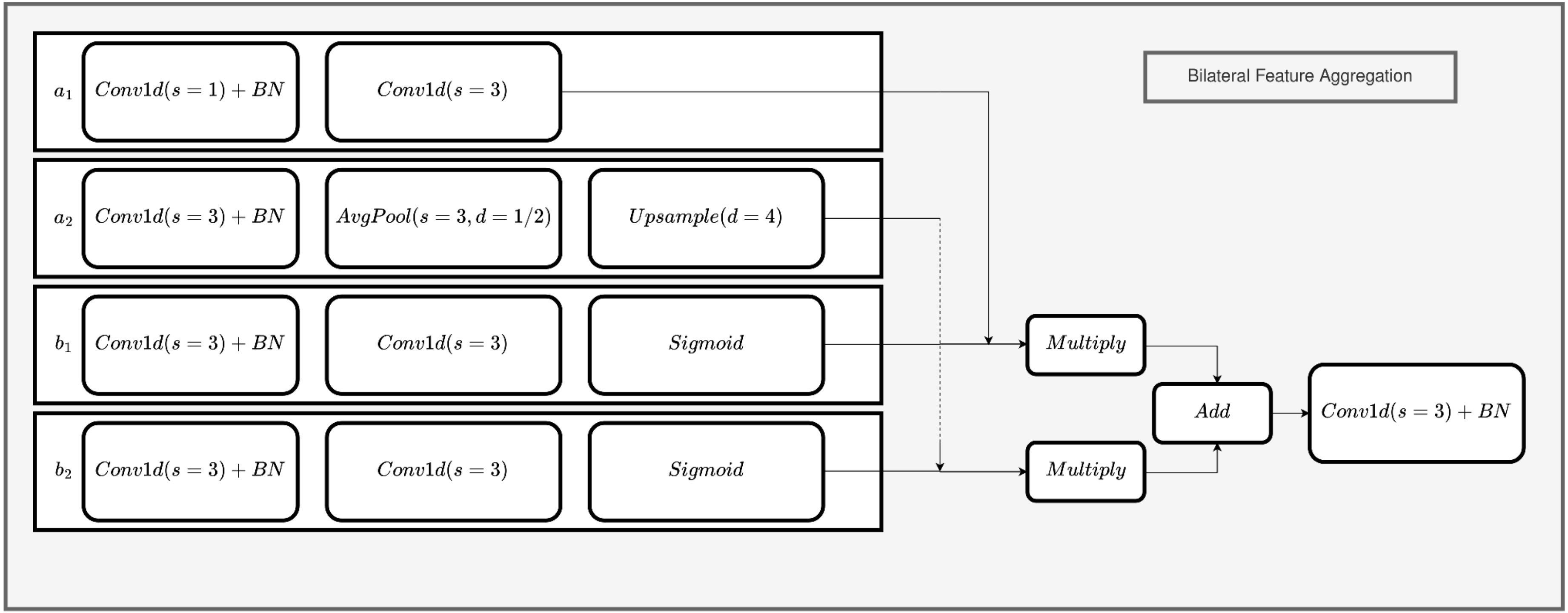
Proposed Model : Important Blocks

1. Channel Attention
2. Bilateral Feature Aggregation
3. Multiscale Feature Aggregation
4. Attentive Rechecking

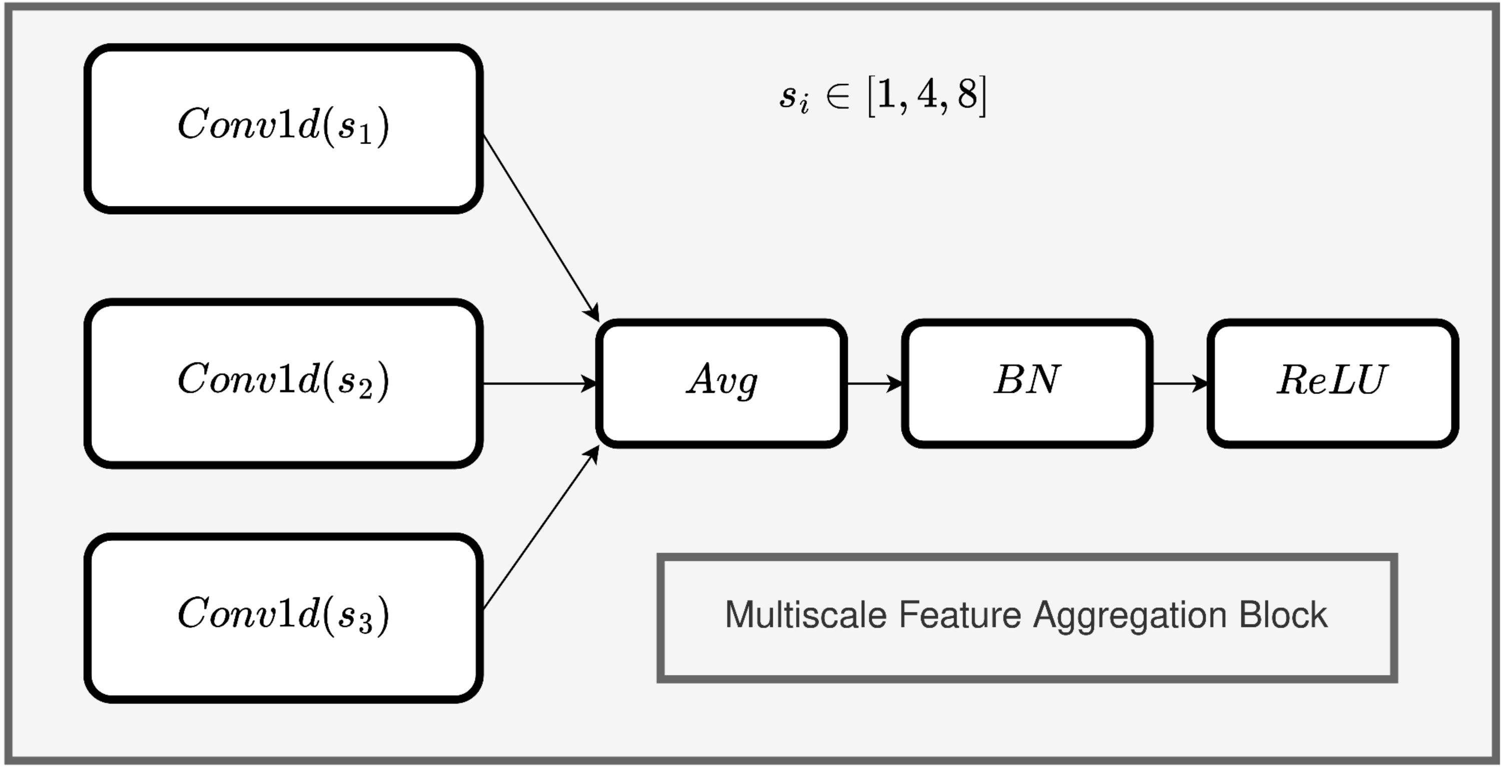
Channel Attention



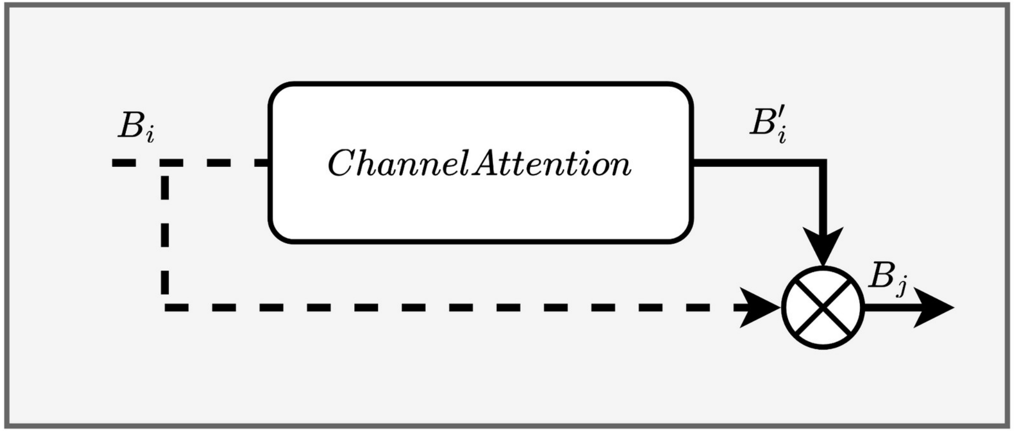
Bilateral Feature Aggregation



Multiscale Feature Aggregation



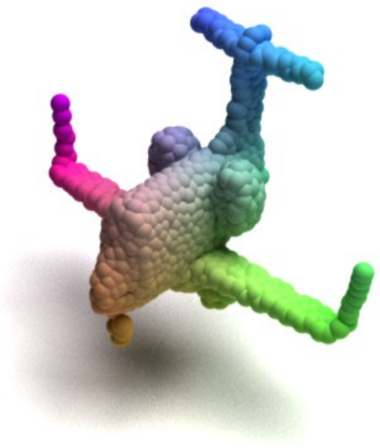
Attentive Rechecking



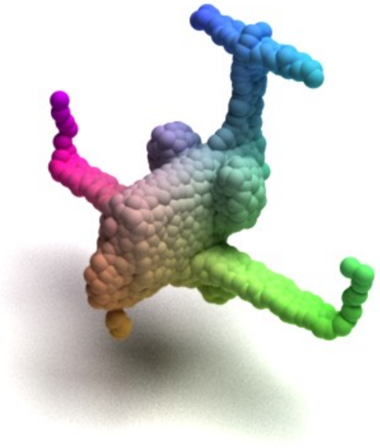
Comparison with Baseline (Task : Auto-encoding Point Clouds)

$Model$	$\#Parameters(M)$	CD	EMD
$Latent-GAN$	1.77	7.12	7.95
$AtlasNet$	44.9	5.13	5.97
$PointFlow$	1.30	7.54	5.18
$Point2Point(ours)$	1.28	4.75	5.05

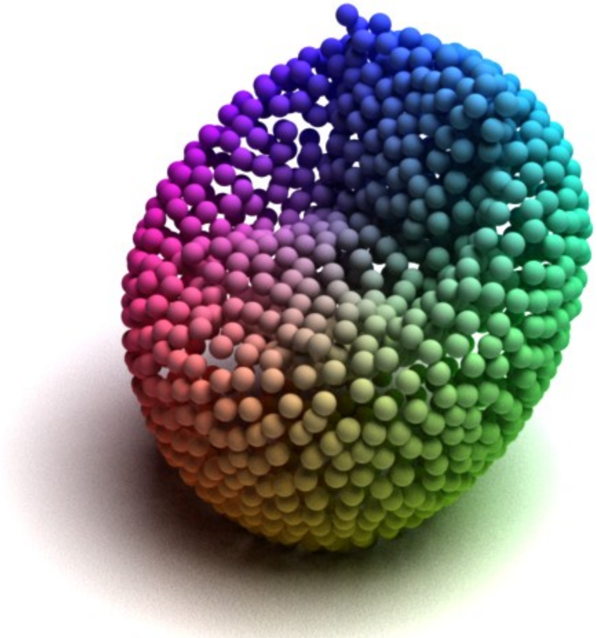
Some Examples



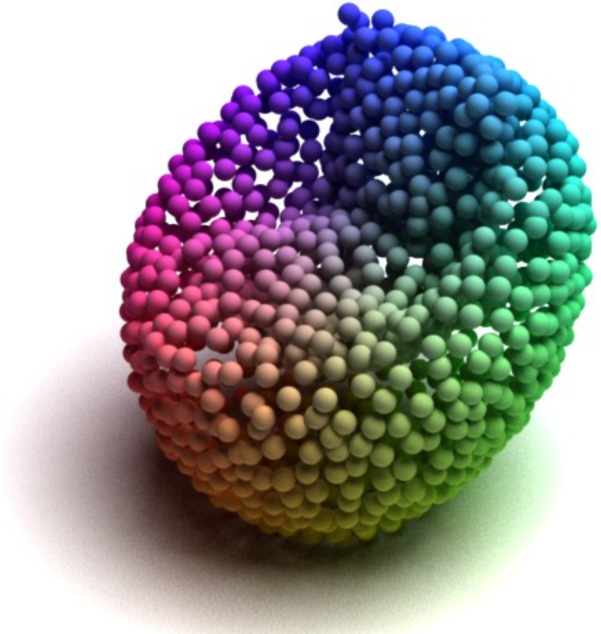
Input



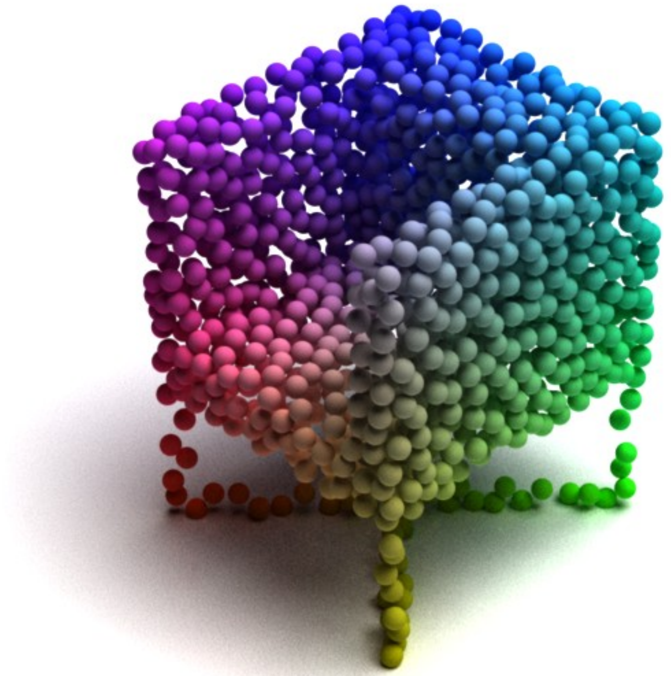
Predicted



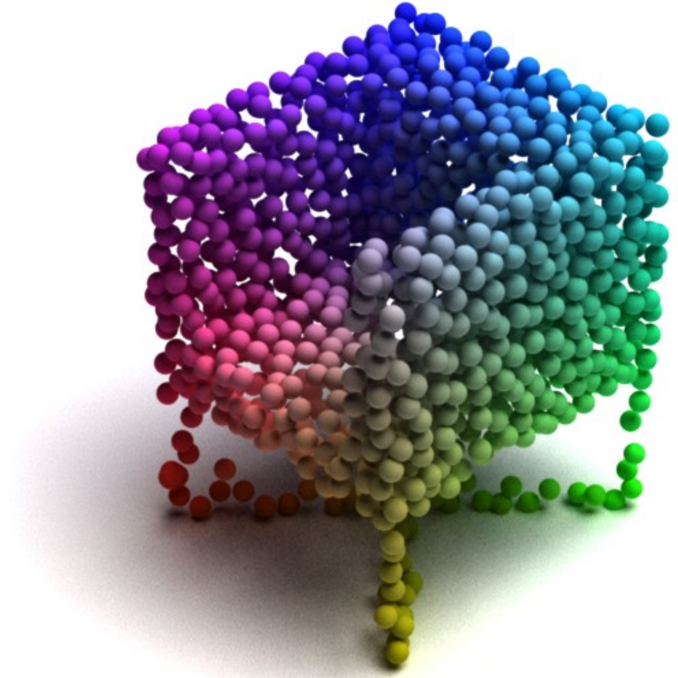
Input



Predicted



Input



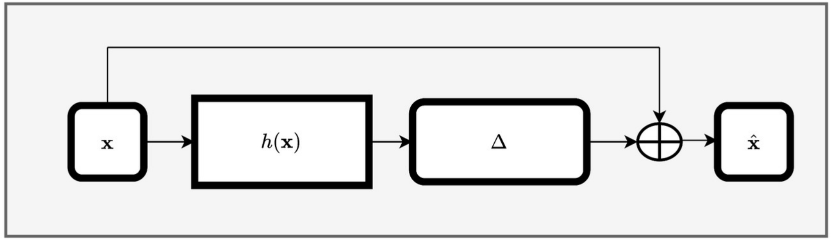
Predicted

Application : Single Step Occupancy Prediction

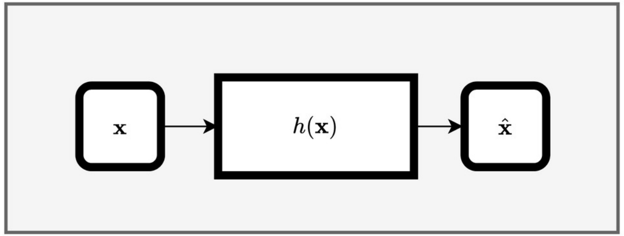
Problem Statement : Given occupancy information at t=k, predict occupancy information at t=k+1.

Learning Methodologies :

Generative Difference Learning :



Generative Learning :



THANK YOU FOR YOUR TIME

References

1. C-Flow: Conditional Generative Flow Models for Images and 3D Point Clouds
2. BiSeNet V2: Bilateral Network with Guided Aggregation for Real-time Semantic Segmentation
3. Space-Filling Curve Based Point Clouds Index
4. Hilbert Space Filling Curve Based Scan-Order for Point Cloud Attribute Compression
5. LOCALITY-PRESERVING PROPERTIES OF SPACE-FILLING CURVES
6. Deep Diffeomorphic Transformer Networks
7. Squeeze-and-Excitation Networks
8. The general theory of permutation equivariant neural networks and higher order graph variational encoders

References (contd.)

9. INTRODUCTION TO DIFFERENTIAL GEOMETRY

10. Multi-Step Prediction of Occupancy Grid Maps with Recurrent Neural Networks

11. Computational Optimal Transport

12. A comparative analysis of some two-dimensional orderings

13. On the Locality Properties of Space-Filling Curves

14. Self-supervised Point Cloud Prediction Using 3D Spatio-temporal Convolutional Networks