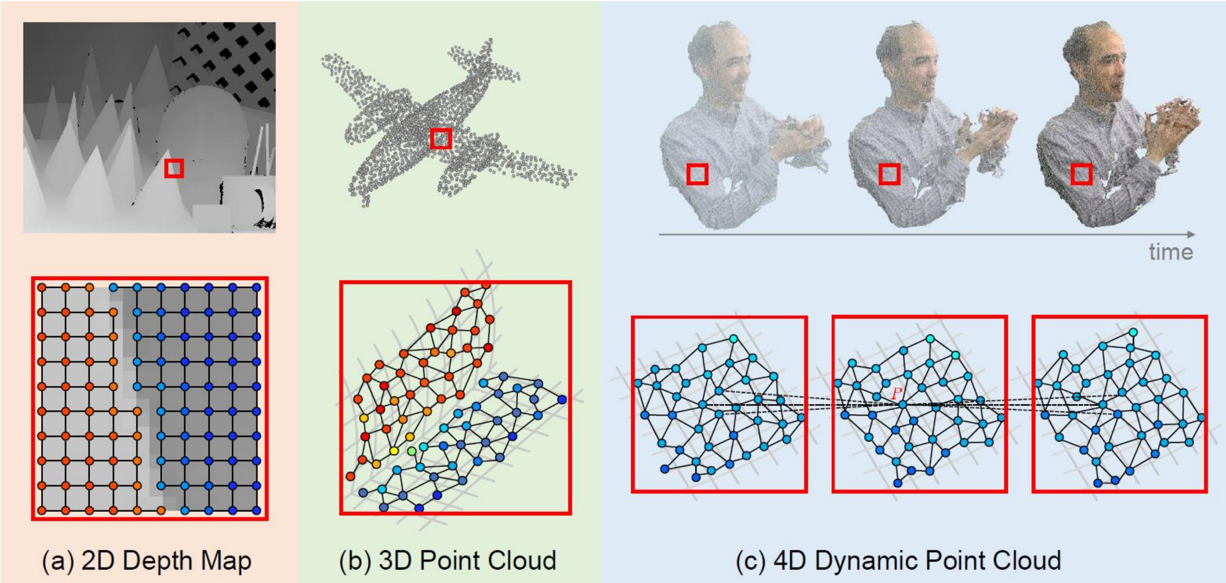
Graph Spectral Processing and Analysis for 3D Point Clouds and Beyond

Wei HU
Assistant Professor
Peking University

June, 2023



Outline

- Introduction to geometric data processing and analysis over graphs
- Basics in Graph Signal Processing and Graph-based Machine Learning
- Point cloud **representation** from feature graph learning
- Point cloud **reconstruction** from graph spectral prior
- Point cloud **analysis** in the graph spectral domain
- Summary and future works

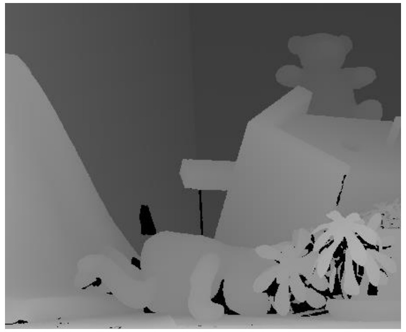
Outline

北京大學 PEKING UNIVERSITY

- Introduction to geometric data processing and analysis over graphs
- Basics in Graph Signal Processing and Graph-based Machine Learning
- Point cloud **representation** from feature graph learning
- Point cloud **reconstruction** from graph spectral prior
- Point cloud **analysis** in the graph spectral domain
- Summary and future works

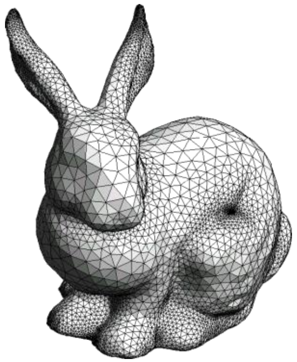
Geometric Data

- Describe the geometry of the 3D world



2D depth map

3D Point Cloud



3D Mesh

time

4D Dynamic Point Cloud

- Acquired by depth sensing, laser scanning or image processing

Microsoft Kinect

Intel RealSense

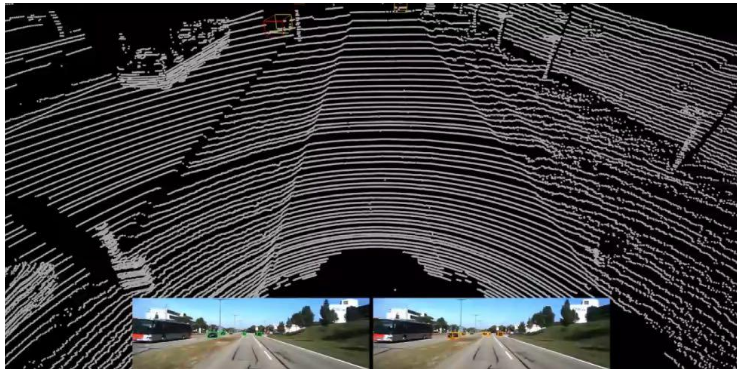
Velodyne LiDAR

LiDAR scanner of Apple iPad Pro

Introduction to geometric data processing and analysis

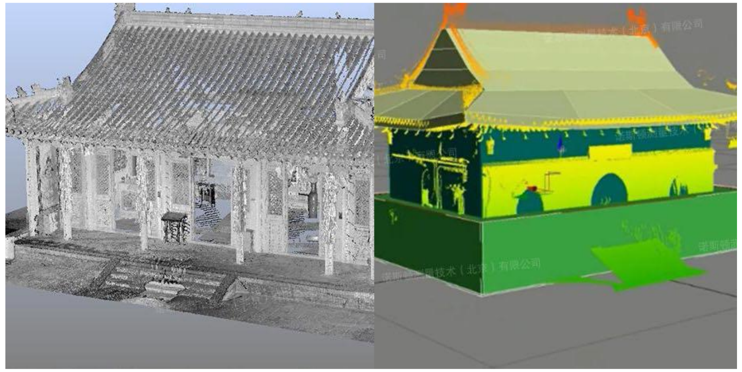
Geometric Data

- Central to a wide range of applications



Navigation in Autonomous Driving

Augmented/Virtual Reality



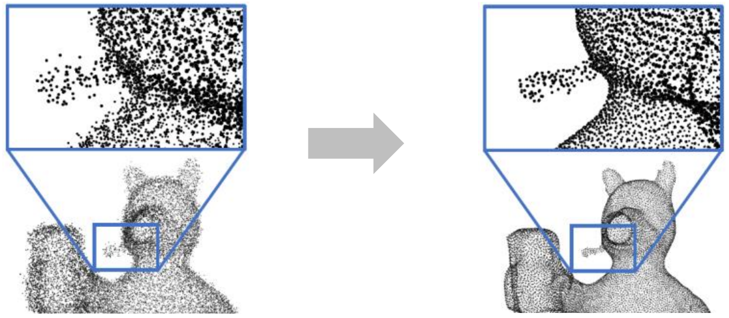
Heritage Protection

Free-viewpoint Video

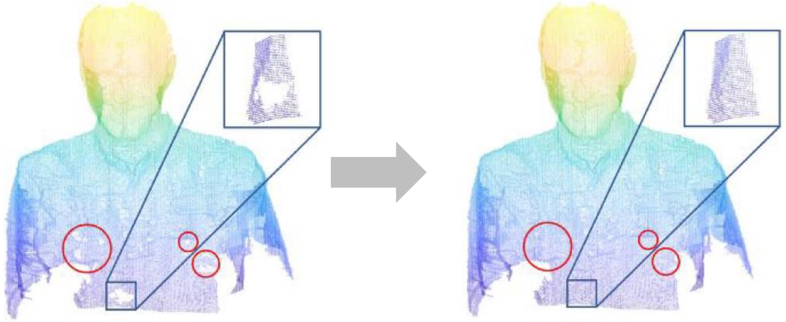
Introduction to geometric data processing and analysis

Tasks

- **Processing**: denoising, inpainting, super-resolution, resampling, etc.

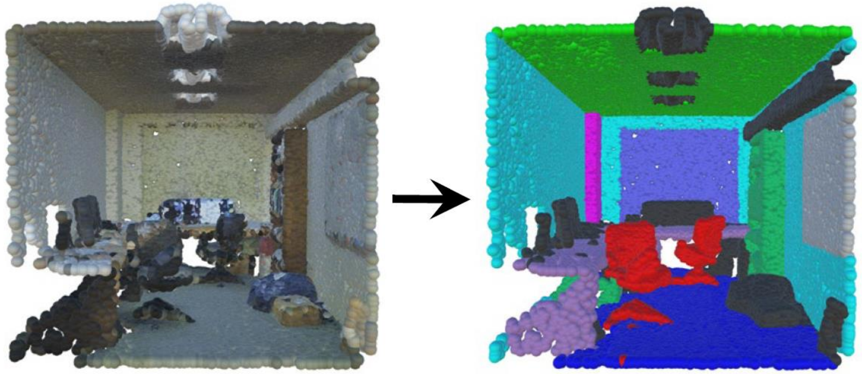


Point Cloud Denoising

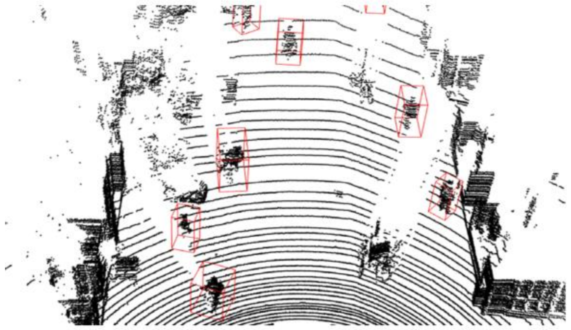


Point Cloud Inpainting

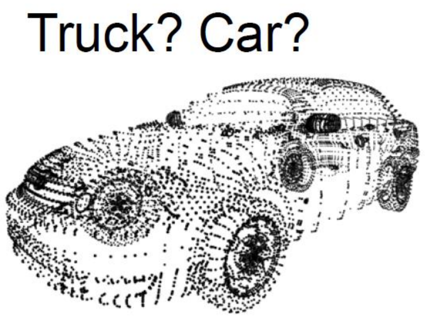
- **Analysis**: classification, segmentation, detection, etc.



Point Cloud Segmentation



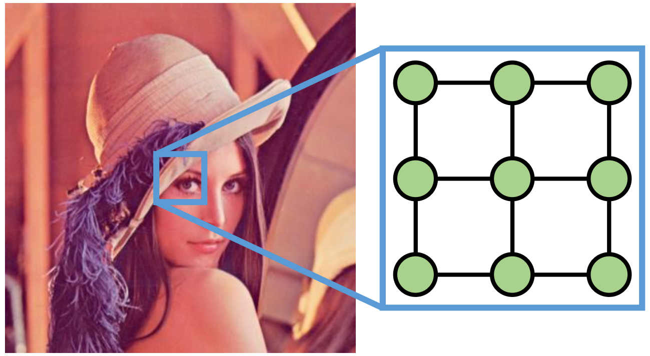
Point Cloud Detection

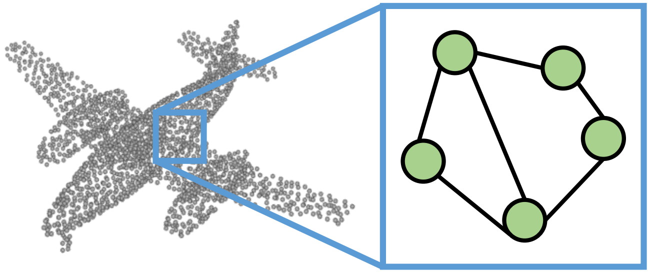


Point Cloud Classification

Challenges

① Unlike images, a wide range of geometric data have **irregular sampling patterns**

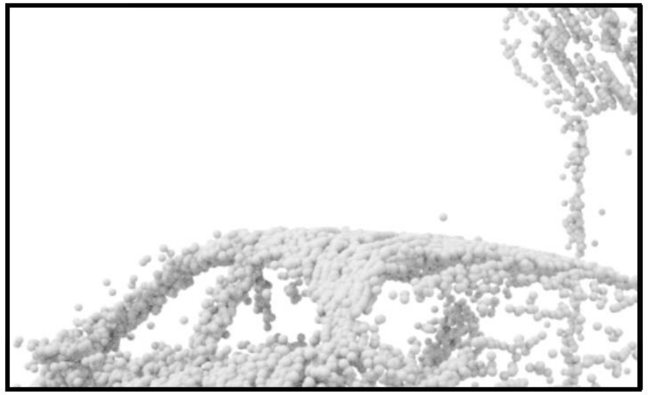




Traditional image/video processing/analysis methods: assume sampling patterns over ***regular*** grids

② Real-world geometric data often suffer from noise, missing data, ….

- Require **Robustness**



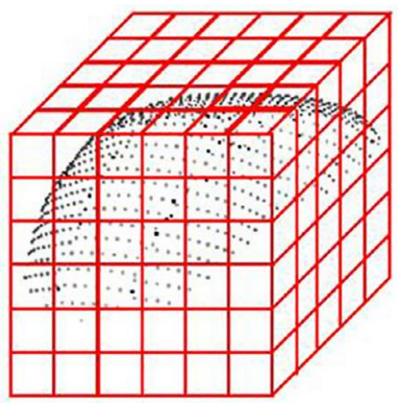
Paris-rue-Madame

③ Model **Interpretability** of geometric deep learning for analysis tasks

Why Graph Representation?

Non-Graph representations of irregular geometric data

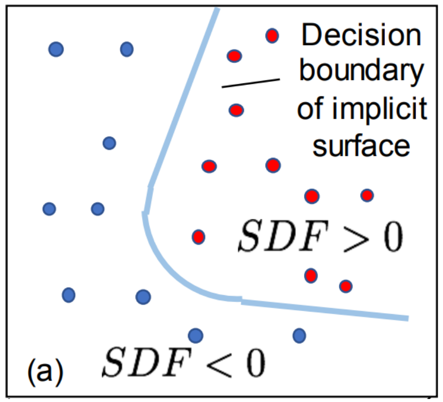
- Quantization-based representations



Quantize onto regular voxel grids

Project onto multiple viewpoints

- Implicit functions



Signed Distance Function

☺ Amenable to existing methods for Euclidean data

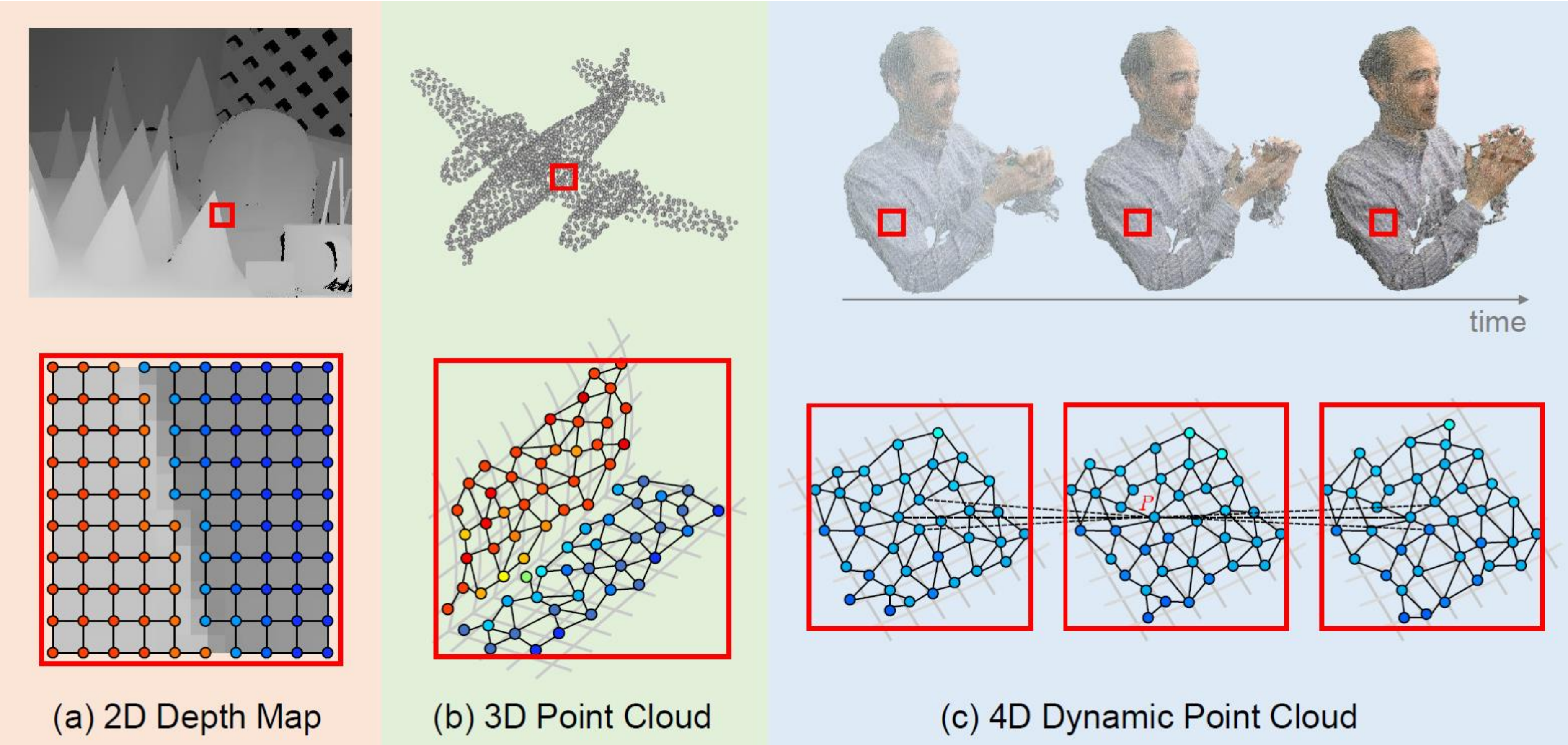
☹ Often deficient in capturing the geometric ***structure*** explicitly

☹ Sometimes inaccurate

☹ Sometimes redundant

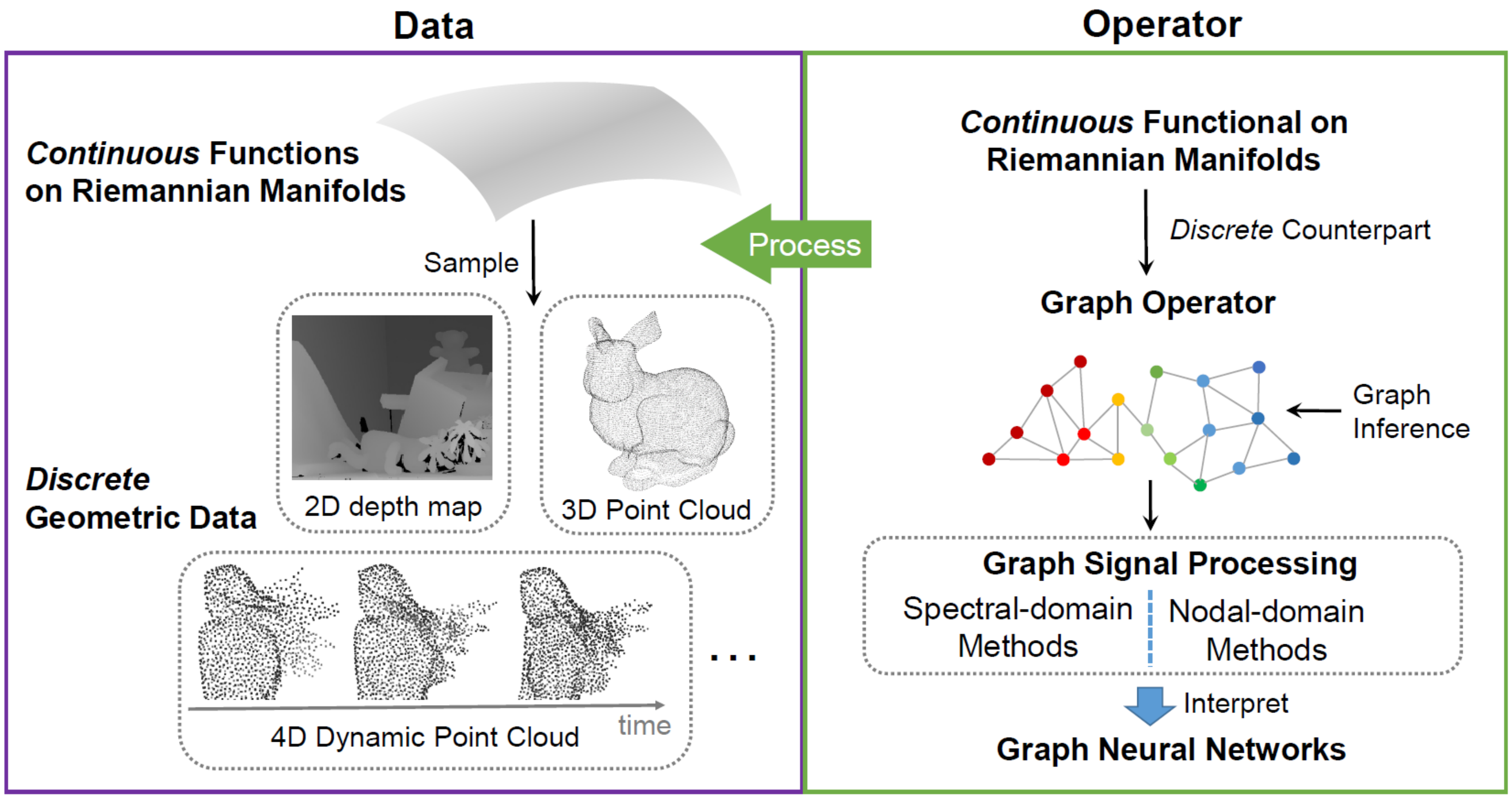
Why Graph Representation?

- Graphs provide ***structure-adaptive***, ***accurate***, and ***compact*** representations for geometric data



Wei Hu, Jiahao Pang, Xianming Liu, Dong Tian, Chia-Wen Lin, Anthony Vetro, "Graph Signal Processing for Geometric Data and Beyond: Theory and Applications," TMM 2021.

Why Graph Representation?



Wei Hu, Jiahao Pang, Xianming Liu, Dong Tian, Chia-Wen Lin, Anthony Vetro, "Graph Signal Processing for Geometric Data and Beyond: Theory and Applications," TMM 2021.

Background in GSP & GNN

Graph Signal Processing (GSP)

- Extend classical signal processing to the graph domain
- Principled mathematical models
- **Theoretical** guarantee

Tools: **Graph filter**, Graph Fourier Transform, graph wavelets, etc.

Graph Neural Network (GNN)

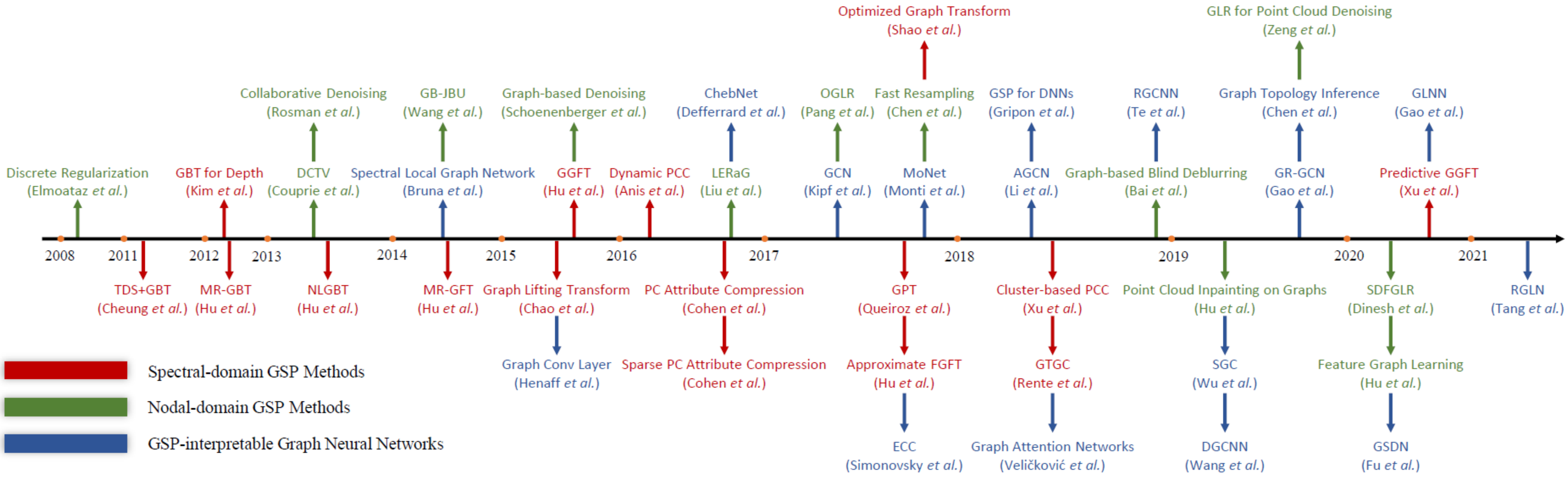
- Extend deep learning techniques to the graph domain
- Data-driven models
- **Empirical** performance

Tools: **Graph convolution**, graph attention, graph pooling, etc.

- **Interpretability** (e.g., interpretation of graph convolution)
- **Introduce GSP-based domain knowledge into GNNs**

Wei Hu, Jiahao Pang, Xianming Liu, Dong Tian, Chia-Wen Lin, Anthony Vetro, "Graph Signal Processing for Geometric Data and Beyond: Theory and Applications," TMM 2021.

Why Graph Representation?



Representative works leveraging GSP/GNNs to process or analyze geometric data.

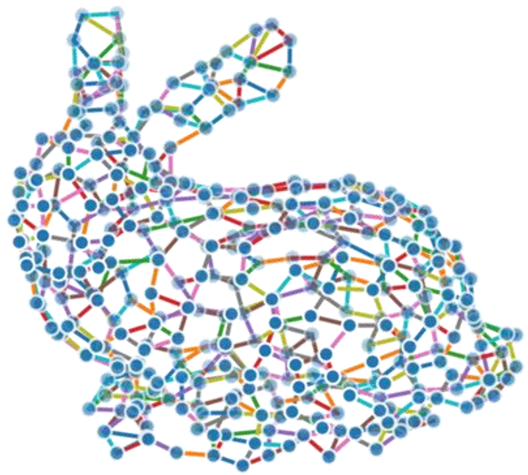
Wei Hu, Jiahao Pang, Xianming Liu, Dong Tian, Chia-Wen Lin, Anthony Vetro, “Graph Signal Processing for Geometric Data and Beyond: Theory and Applications,” TMM 2021.

Outline

- Introduction to geometric data processing and analysis over graphs
- Basics in Graph Signal Processing and Graph-based Machine Learning
- Point cloud **representation** from feature graph learning
- Point cloud **reconstruction** from graph spectral prior
- Point cloud **analysis** in the graph spectral domain
- Summary and future works

Basic Definition

- Graph: vertices (nodes) connected via some edges (links)
- Graph Signal: set of scalar/vector values defined on the vertices.



Graph $G = (\mathcal{V}, E, w)$

Vertex Set $\mathcal{V} = \{v_1, v_2, ...\}$

Edge Set $\mathbf{E} = \{(v_1, v_2), (v_1, v_3), ...\}$

Weighted edges w, sets of weights a_{ij}

Graph Signal $\mathbf{x} = \{x_1, x_2, ...\}$

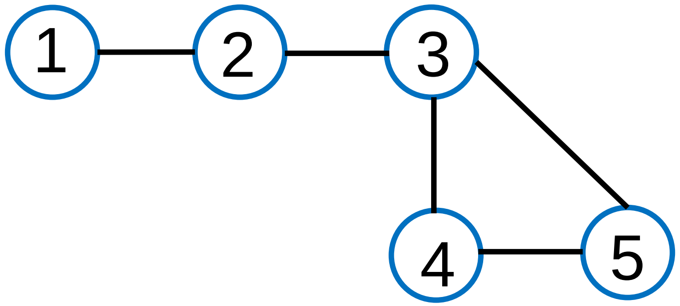
Neighborhood, h-hop

$\mathcal{N}_h(i) = \{j \in \mathcal{V} : hop\_dist(i, j) \leq h\}$

Allow to define locality on the graph

Basic Definition

- Adjacency matrix: $\mathbf{A}$
  - $a_{i,j}$: edge weight for the edge (v_i, v_j)
  - Describe the similarity / correlation between nodes
  - Undirected graph: $a_{i,j} = a_{j,i}$

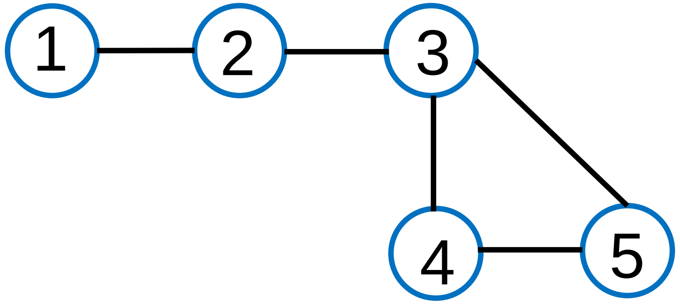


- Degree matrix: $\mathbf{D}$

$$d_{i,i} = \sum_{j=1}^{N} a_{i,j}$$

Basic Definition

- Combinatorial Graph Laplacian matrix



$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

  - $\mathbf{L}$ is symmetric
  - When operating $\mathbf{L}$ on a graph signal $\mathbf{x}$, it captures the variation in the signal

$$(\mathbf{Lx})(i) = \sum_{j \in \mathcal{N}_i} a_{i,j}(x_i - x_j)$$

  - Total variation

$$\mathbf{x}^T \mathbf{Lx} = \sum_{i \sim j} a_{i,j}(x_i - x_j)^2$$

\- Graph Laplacian Regularizer

Basic Definition

- The graph Laplacian $\mathbf{L} \in \mathcal{R}^{N\times N}$ is real and symmetric: $\mathbf{L}\psi_l = \lambda_l \psi_l$
  - a set of real eigenvalues $\{\lambda_l\}_{l=0}^{N-1}$ - graph frequency
  - a complete set of orthonormal eigenvectors $\{\psi_l\}_{l=0}^{N-1}$
- The eigenvectors $\{\psi_l\}_{l=0}^{N-1}$ define the GFT basis:

$$\mathbf{\Phi} = \begin{bmatrix} | & & | \\ \psi_0 & \cdots & \psi_{N-1} \\ | & & | \end{bmatrix}$$

- For any signal $\mathbf{x} \in \mathcal{R}^N$ residing on the nodes of $\mathcal{G}$, its GFT $\hat{\mathbf{x}} \in \mathcal{R}^N$ is defined as

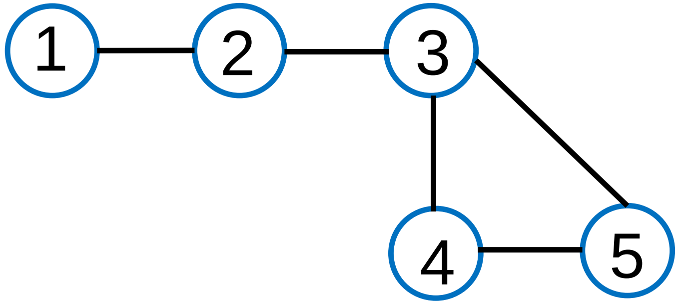
$$\hat{\mathbf{x}}(l) = < \psi_l, \mathbf{x} >, l = 0, 1, ..., N-1$$

$$(\hat{\mathbf{x}} = \mathbf{\Phi}^T \mathbf{x})$$

GFT coefficients → $\hat{\mathbf{x}}$; GFT basis → $\mathbf{\Phi}$; graph signal → $\mathbf{x}$

Basic Definition

- Combinatorial Graph Laplacian matrix



$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

  - $\mathbf{L}$ is symmetric
  - When operating $\mathbf{L}$ on a graph signal $\mathbf{x}$, it captures the variation in the signal

$$(\mathbf{Lx})(i) = \sum_{j \in \mathcal{N}_i} a_{i,j}(x_i - x_j)$$

  - Total variation

$$\mathbf{x}^T\mathbf{Lx} = \sum_{i \sim j} a_{i,j}(x_i - x_j)^2$$

- Graph Laplacian Regularizer

$$= \sum \lambda_k \hat{\mathbf{x}}_k^2$$

- Spectral interpretation

Why Graph Fourier Transform

- Offer compact transform domain representation

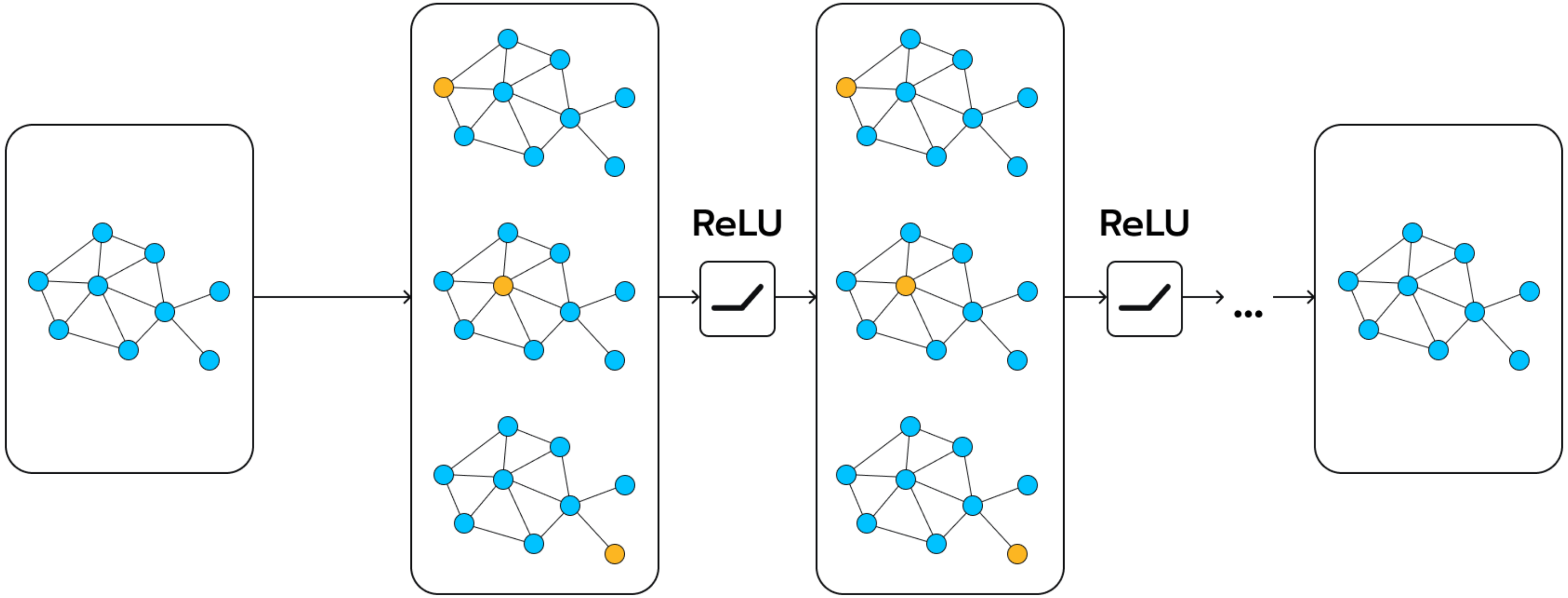
(a) Original. (b) 10 frequencies. (c) 100 frequencies. (d) 400 frequencies. (e) Graph spectral distribution.

- Reason: the graph adaptively captures the correlation in the graph signal

- $\approx$ KLT for a family of statistical models

Wei Hu, Gene Cheung, Antonio Ortega, Oscar C. Au, “Multi-resolution Graph Fourier Transform for Compression of Piecewise Smooth Images,” *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 419-433, January 2015.

Graph Neural Networks



Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P., “Geometric deep learning: going beyond Euclidean Data,” *IEEE Signal Processing Magazine*. 2017 Jul 11;34(4):18-42.

Graph Convolution

Euclidean	Non-Euclidean
Spatial domain $$(f \star g)(x) = \int_{-\pi}^{\pi} f(x')g(x-x')dx'$$	?
Spectral domain $$\widehat{(f \star g)}(\omega) = \hat{f}(\omega) \cdot \hat{g}(\omega)$$ 'Convolution Theorem'	?

Graph Convolution

Euclidean	Non-Euclidean
Spatial domain $$(f \star g)(x) = \int_{-\pi}^{\pi} f(x')g(x-x')dx'$$	$$f_i' = \square_{i':(i,i')\in\mathcal{E}} h_\Theta(f_i, f_{i'})$$
Spectral domain $$\widehat{(f \star g)}(\omega) = \hat{f}(\omega) \cdot \hat{g}(\omega)$$ 'Convolution Theorem'	$$\widehat{\mathbf{f} \star \mathbf{g}} = (\mathbf{\Phi}^\top \mathbf{g}) \circ (\mathbf{\Phi}^\top \mathbf{f})$$

Graph Convolution

Euclidean

Spatial domain

$$(f \star g)(x) = \int_{-\pi}^{\pi} f(x')g(x-x')dx'$$

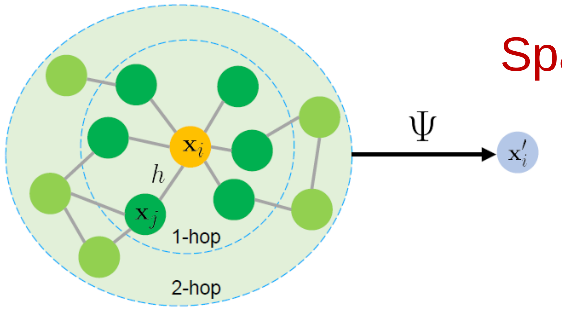
Spectral domain

$$\widehat{(f \star g)}(\omega) = \hat{f}(\omega) \cdot \hat{g}(\omega)$$

'Convolution Theorem'

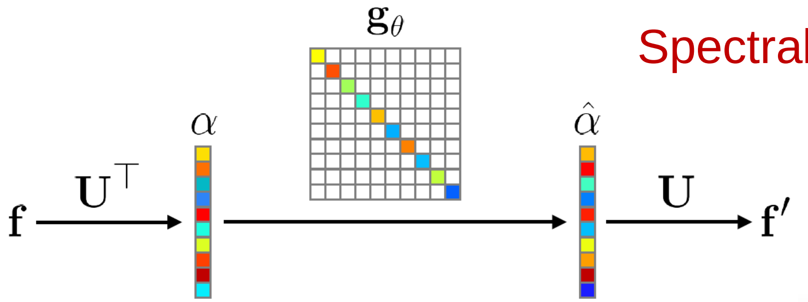
Non-Euclidean

$$f_i' = \square_{i':(i,i')\in\mathcal{E}} h_\Theta(f_i, f_{i'})$$



Spatial graph filtering

$$\widehat{\mathbf{f} \star \mathbf{g}} = (\mathbf{\Phi}^\top \mathbf{g}) \circ (\mathbf{\Phi}^\top \mathbf{f})$$

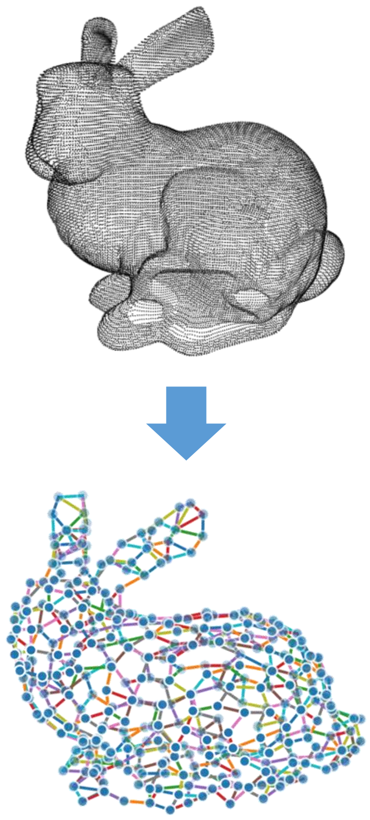


Spectral graph filtering

Outline

- Introduction to geometric data processing and analysis over graphs
- Basics in Graph Signal Processing and Graph-based Machine Learning
- Point cloud **representation** from feature graph learning
- Point cloud **reconstruction** from graph spectral prior
- Point cloud **analysis** in the graph spectral domain
- Summary and future works

Problem Statement - Feature graph learning



- **Problem:** The graph is often unavailable over geometric data
- **Previous works:**
  - ➢ Previous graph learning methods often require *multiple observations*
- **Contributions:**
  - ➢ Given **feature vector** per node, we propose feature graph learning from only **a single or even partial signal observation**
  - ➢ Develop a fast algorithm (eigen-decomposition-free)

Wei Hu, Xiang Gao, Gene Cheung, Zongming Guo, "Feature Graph Learning for 3D Point Cloud Denoising," *IEEE Transactions on Signal Processing (TSP)*, vol. 68, pp.2841-2856, March 2020.

Cheng Yang, Gene Cheung, Wei Hu, "Signed Graph Metric Learning via Gershgorin Disc Alignment," accepted to *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

Key Idea - Feature graph learning

PEKING UNIVERSITY

- The key idea: learn a good distance metric matrix
- Given **a single or partial observation** with relevant **feature vector** $\mathbf{f}_i$, compute the *Mahalanobis distance*: $\delta_{i,j} = (\mathbf{f}_i - \mathbf{f}_j)^\top \mathbf{M}(\mathbf{f}_i - \mathbf{f}_j)$

  ↖ PD **metric matrix**

- **Edge weight** of **feature graph** is $w_{i,j} = \exp\{-\delta_{i,j}\}$ ⟵ feature distance
- Minimize **Graph Laplacian Regularizer** (GLR):

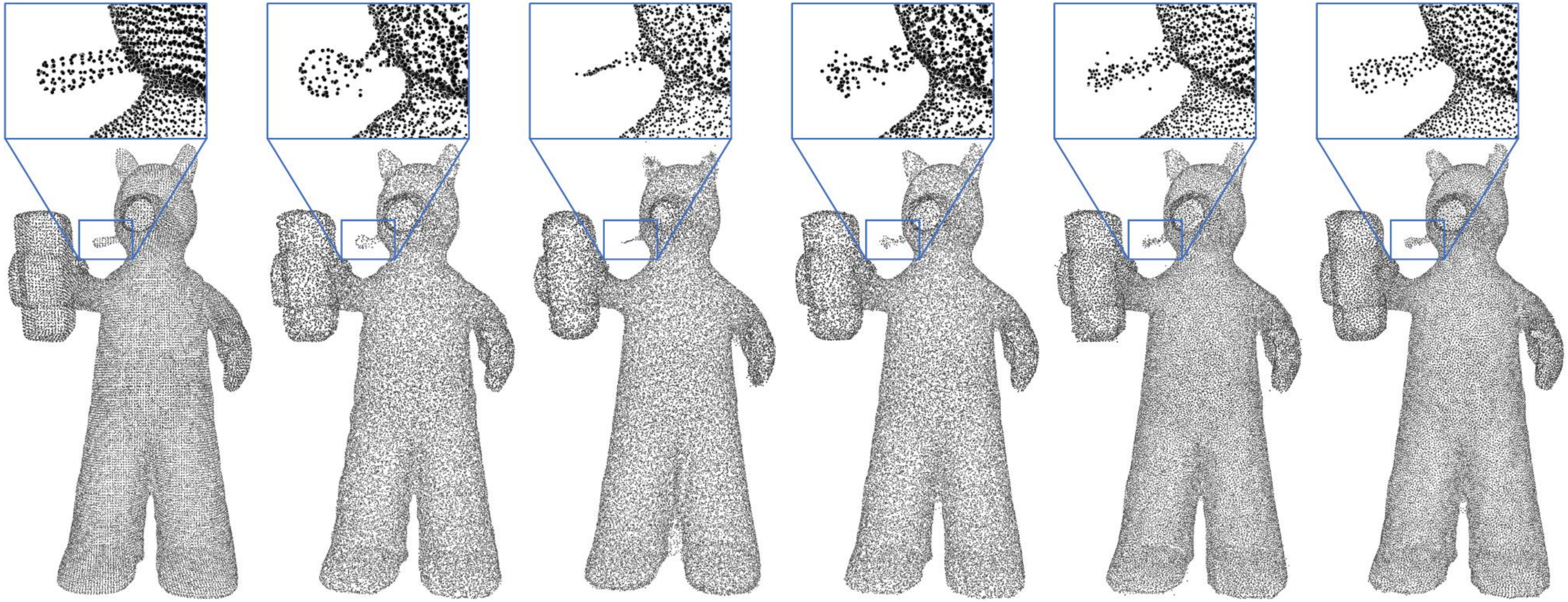
$$\min_{\mathbf{M}} \boxed{\mathbf{x}^\top \mathbf{L}\mathbf{x} = \sum_{i,j} w_{i,j}(x_i - x_j)^2} = \sum_{i,j} \exp\{-(\mathbf{f}_i - \mathbf{f}_j)^\top \mathbf{M}(\mathbf{f}_i - \mathbf{f}_j)\} d_{i,j}$$

$$\text{s.t.} \quad \mathbf{M} \succ 0; \quad \mathrm{tr}(\mathbf{M}) \leq C.$$

Minimizing GLR makes the graph adapt to the signal structure

➢ Solved via our proposed **eigen-decomposition-free** block-coordinate descent algorithm

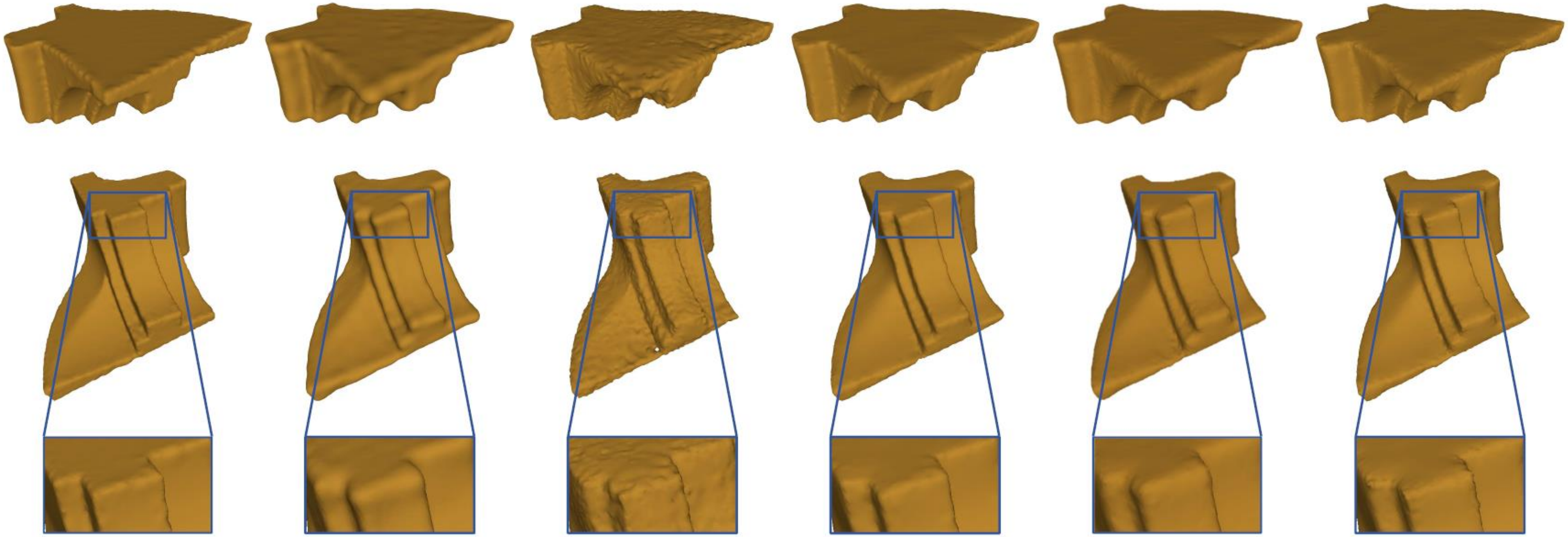
Results: 3D Point Cloud Denoising



(a) Ground-truth (b) APSS (c) AWLOP (d) MRPCA (e) GLR (f) Ours

Wei Hu, Xiang Gao, Gene Cheung, Zongming Guo, "Feature Graph Learning for 3D Point Cloud Denoising," *IEEE Transactions on Signal Processing (TSP)*, vol. 68, pp.2841-2856, March 2020.

Results: 3D Point Cloud Denoising



(a) Ground-truth (b) APSS (c) AWLOP (d) MRPCA (e) GLR (f) Ours

Wei Hu, Xiang Gao, Gene Cheung, Zongming Guo, “Feature Graph Learning for 3D Point Cloud Denoising,” *IEEE Transactions on Signal Processing (TSP)*, vol. 68, pp.2841-2856, March 2020.

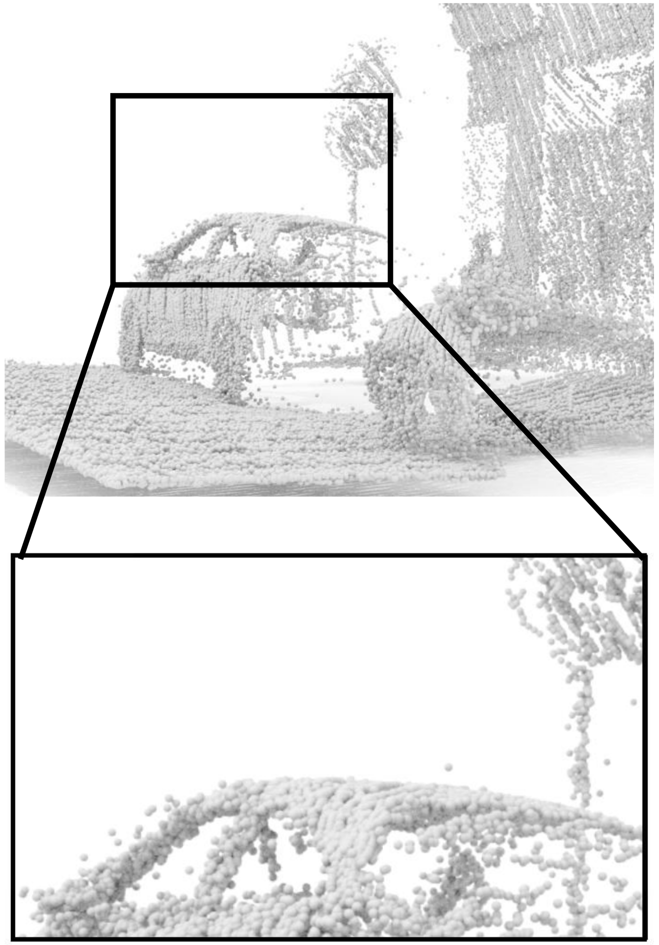
Outline

北京大學 PEKING UNIVERSITY

- Introduction to geometric data processing and analysis over graphs
- Basics in Graph Signal Processing and Graph-based Machine Learning
- Point cloud **representation** from feature graph learning
- Point cloud **reconstruction** from graph spectral prior
- Point cloud **analysis** in the graph spectral domain
- Summary and future works

Problem Statement – Deep Point Set Resampling

- **Problem:** Real-word data often suffer from **noise**, **low density**…
- **Previous Works:**
  - Optimization-based approaches rely heavily on geometric priors
  - Deep learning methods often suffer from over-estimation or under-estimation of the displacement
- **Contributions:**
  - propose deep point set resampling for point cloud restoration, which models the distribution of degraded point clouds via **gradient fields** and converges points towards the underlying surface for restoration.

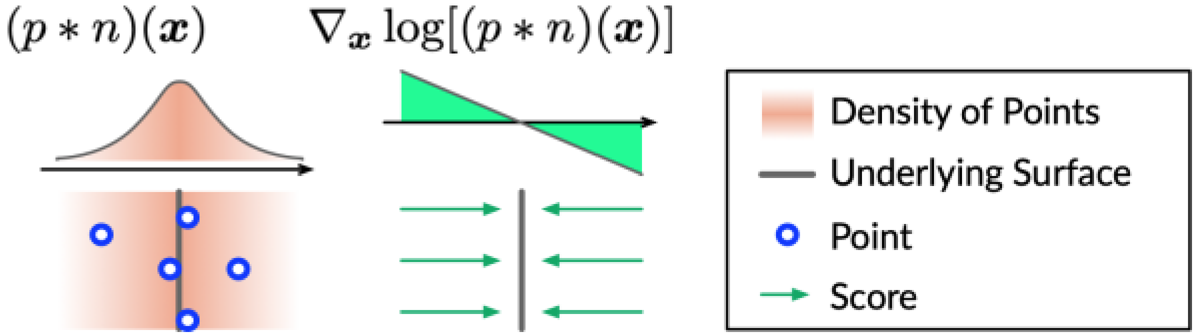


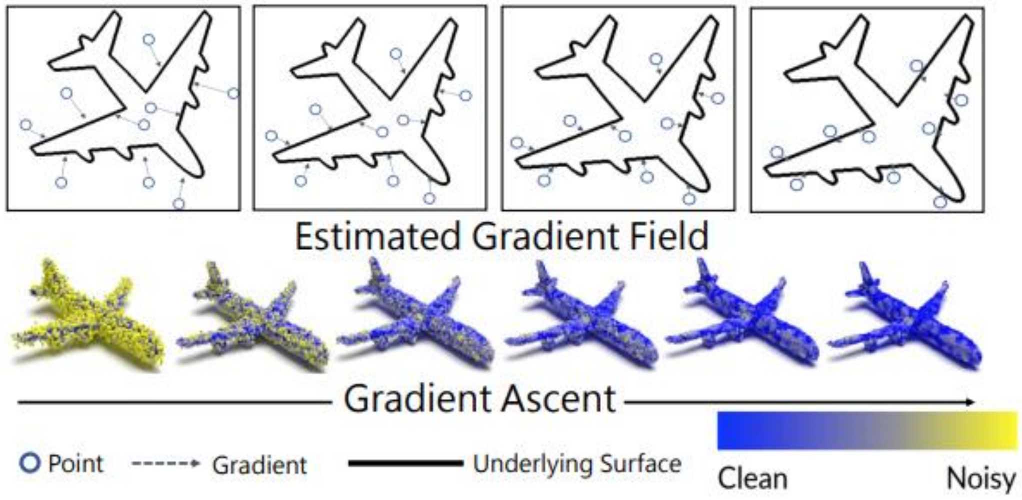
Paris-rue-Madame

Shitong Luo, Wei Hu, "Score-Based Point Cloud Denoising," ICCV 2021.
Haolan Chen, Bi'an Du, Shitong Luo, Wei Hu, "Deep Point Set Resampling via Gradient Fields," TPAMI, 2023.

Key Idea - Deep Point Set Resampling

Key observation: the distribution of a noisy point cloud can be viewed as the distribution of noise-free points $p(\boldsymbol{x})$ convolved with some noise model n, leading to $(p * n)(\boldsymbol{x})$



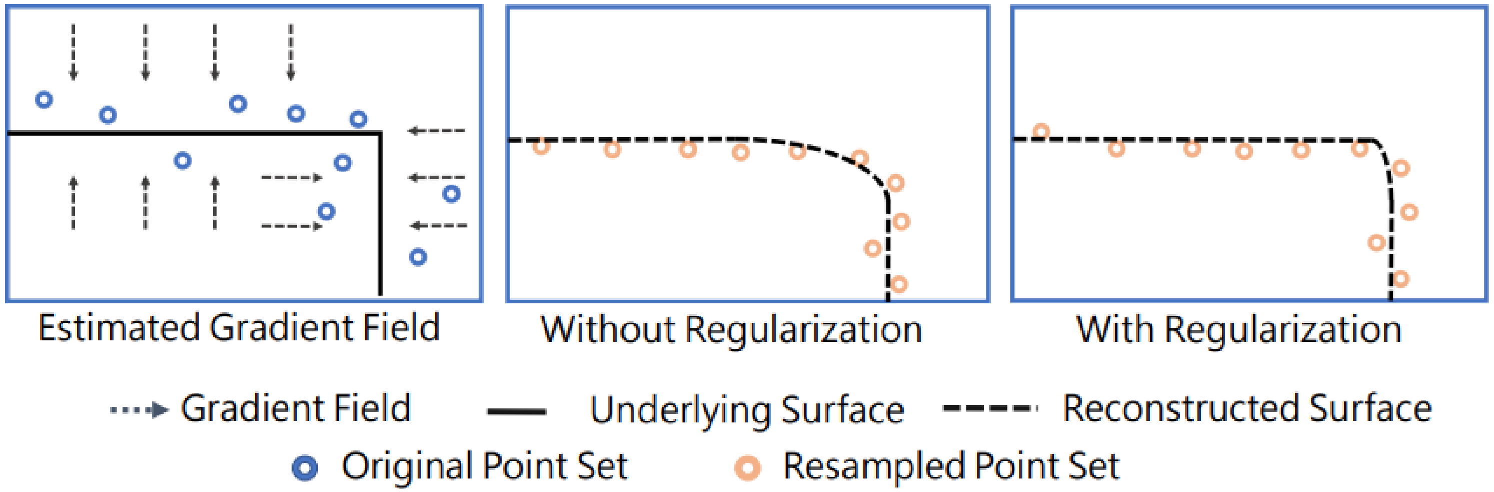


Perform gradient ascent on the log-probability function $\log[(p * n)(\boldsymbol{x})]$? $p * n$ is unknown!

- estimate the **gradient field** of the distribution: $\nabla_{\boldsymbol{x}} \log[(p * n)(\boldsymbol{x})]$,
- denoise the point cloud by gradient ascent to move noisy points towards the mode of $p * n$

Key Idea - Deep Point Set Resampling

- introduce **regularization** (GLR, etc.) into the point set resampling process, to enhance the intermediate resampled point cloud iteratively **during the inference**



$$
\begin{aligned}
\boldsymbol{x}_i^{(0)} &= \boldsymbol{x}_i,\ \boldsymbol{x}_i \in \boldsymbol{X}, \\
\tilde{\boldsymbol{x}}_i^{(t)} &= \boldsymbol{x}_i^{(t-1)} + \alpha_t \boldsymbol{g}(\boldsymbol{x}_i^{(t-1)}), \\
\boldsymbol{x}_i^{(t)} &= (\boldsymbol{I} + \lambda \cdot \mathcal{L})^{-1} \tilde{\boldsymbol{x}}_i^{(t)},\ t = 1, \ldots, T,
\end{aligned}
$$

Reweighted →

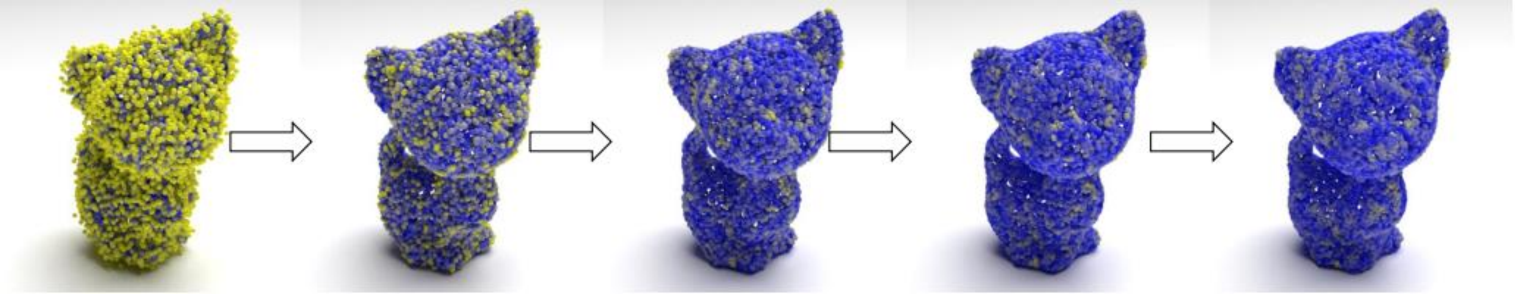
$$
\begin{aligned}
\boldsymbol{x}_i^{(0)} &= \boldsymbol{x}_i,\ \boldsymbol{x}_i \in \boldsymbol{X}, \\
\tilde{\boldsymbol{x}}_i^{(t)} &= \boldsymbol{x}_i^{(t-1)} + \alpha_t \boldsymbol{g}(\boldsymbol{x}_i^{(t-1)}), \\
\mathcal{L}^{(t)} &\leftarrow \tilde{\boldsymbol{X}}^{(t)}, \\
\boldsymbol{x}_i^{(t)} &= (\boldsymbol{I} + \lambda \cdot \mathcal{L}^{(t)})^{-1} \tilde{\boldsymbol{x}}_i^{(t)},\ t = 1, \ldots, T,
\end{aligned}
$$

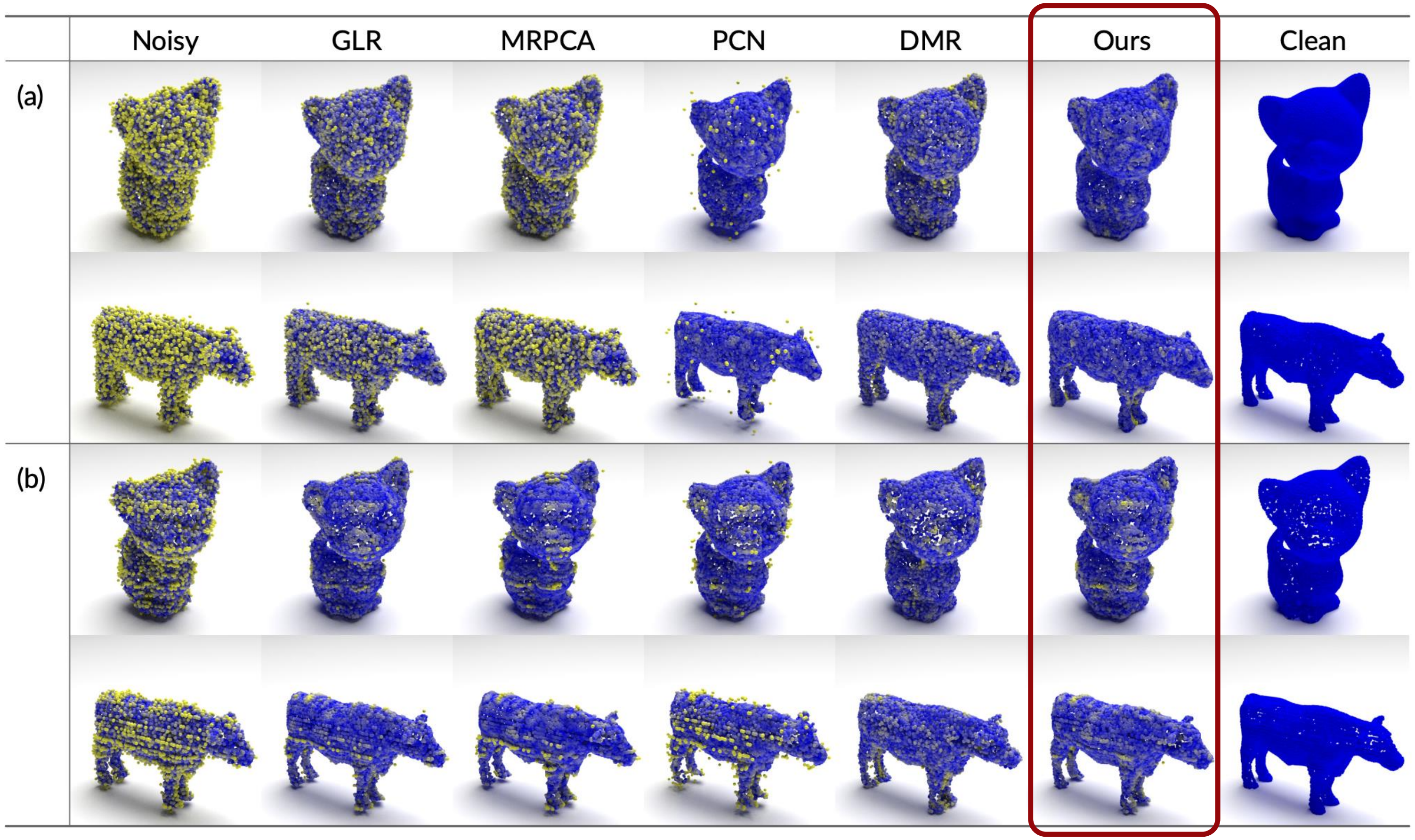
Results: Synthetic Point Cloud Denoising

# Points		10K (Sparse)						50K (Dense)					
Noise		1%		2%		3%		1%		2%		3%	
Dataset	Model	CD	P2M	CD	P2M	CD	P2M	CD	P2M	CD	P2M	CD	P2M
PU [36]	Bilateral [24]	3.646	1.342	5.007	2.018	6.998	3.557	0.877	0.234	2.376	1.389	6.304	4.730
	Jet [5]	2.712	0.613	4.155	1.347	6.262	2.921	0.851	0.207	2.432	1.403	5.788	4.267
	MRPCA [8]	2.972	0.922	3.728	1.117	5.009	1.963	0.669	0.099	2.008	1.033	5.775	4.081
	GLR [28]	2.959	1.052	3.773	1.306	4.909	2.114	0.696	0.161	1.587	0.830	3.839	2.707
	PCN [19]	3.515	1.148	7.467	3.965	13.067	8.737	1.049	0.346	1.447	0.608	2.289	1.285
	GPDNet [21]	3.780	1.337	8.007	4.426	13.482	9.114	1.913	1.037	5.021	3.736	9.705	7.998
	DMR [34]	4.482	1.722	4.982	2.115	5.892	2.846	1.162	0.469	1.566	0.800	2.432	1.528
	Score [22]	2.521	0.463	3.686	1.074	4.708	1.942	0.716	0.150	1.288	0.566	1.928	1.041
	Ours	**2.353**	**0.306**	**3.350**	**0.734**	**4.075**	**1.242**	**0.649**	**0.076**	**0.997**	**0.296**	**1.344**	**0.531**
PC [19]	Bilateral [24]	4.320	1.351	6.171	1.646	8.295	2.392	1.172	0.198	2.478	0.634	6.077	2.189
	Jet [5]	3.032	0.830	5.298	1.372	7.650	2.227	1.091	0.180	2.582	0.700	5.787	2.144
	MRPCA [8]	3.323	0.931	4.874	1.178	6.502	1.676	0.966	0.140	2.153	0.478	5.570	1.976
	GLR [28]	3.399	0.956	5.274	1.146	7.249	1.674	**0.964**	**0.134**	2.015	0.417	4.488	1.306
	PCN [19]	3.847	1.221	8.752	3.043	14.525	5.873	1.293	0.289	1.913	0.505	3.249	1.076
	GPDNet [21]	5.470	1.973	10.006	3.650	15.521	6.353	5.310	1.716	7.709	2.859	11.941	5.130
	DMR [34]	6.602	2.152	7.145	2.237	8.087	2.487	1.566	0.350	2.009	0.485	2.993	0.859
	Score [22]	3.369	0.830	5.132	1.195	6.776	1.941	1.066	0.177	1.659	0.354	2.494	0.657
	Ours	**2.873**	**0.783**	**4.757**	**1.118**	**6.031**	**1.619**	1.010	0.146	**1.515**	**0.340**	**2.093**	**0.573**

Results: Synthetic Point Cloud Denoising

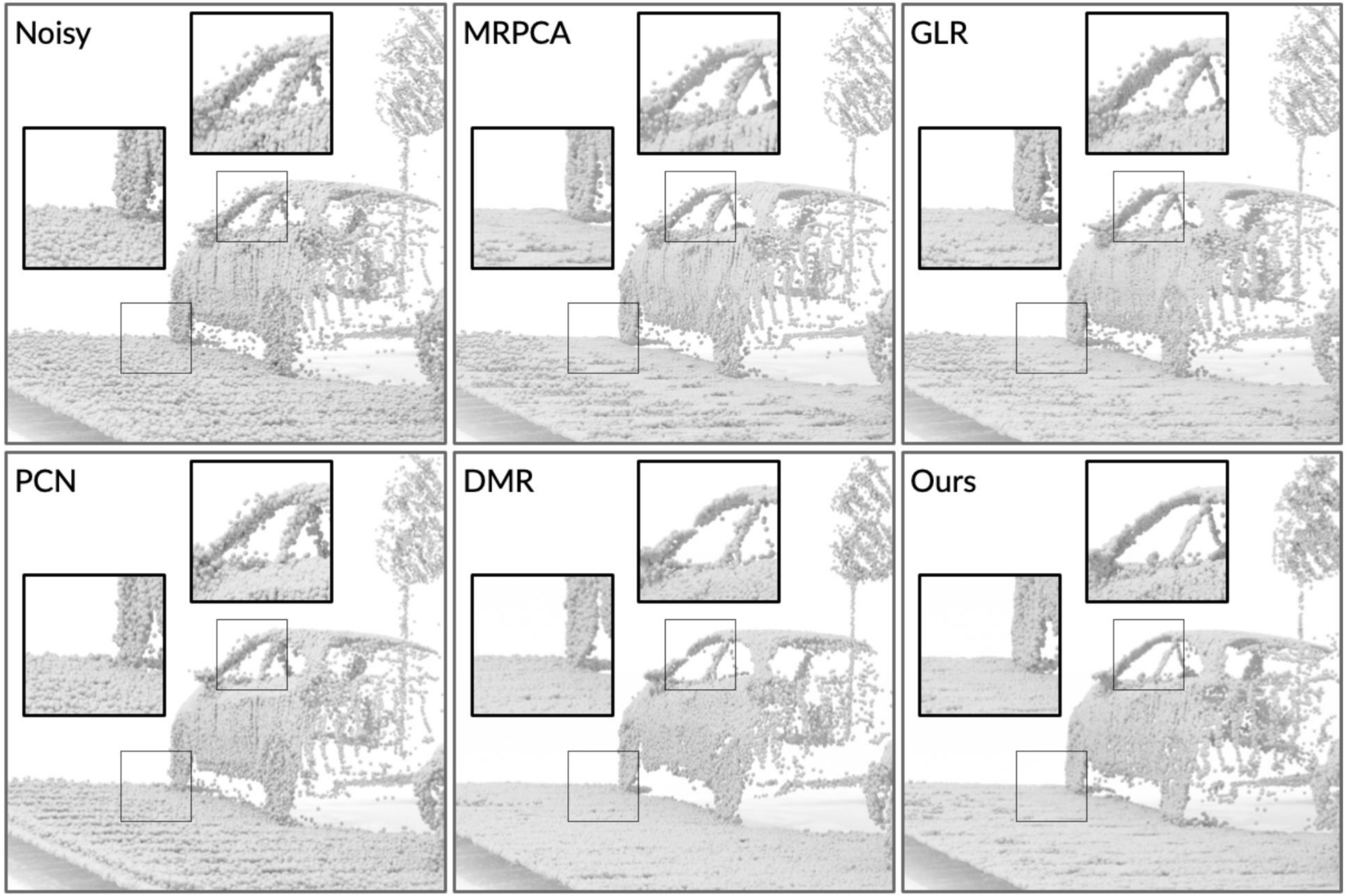
A gradient ascent trajectory of our point cloud denoising every other 10 steps.





Comparison with other methods
(a) Gaussian noise
(b) Synthetic Lidar noise

Results: Real-world Point Cloud Denoising



Outline

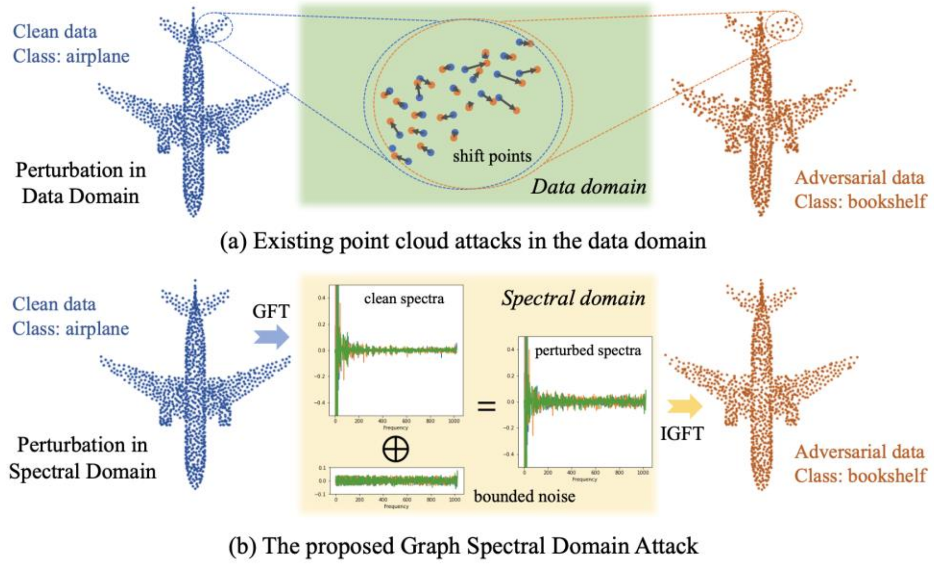
- Introduction to geometric data processing and analysis over graphs
- Basics in Graph Signal Processing and Graph-based Machine Learning
- Point cloud **representation** from feature graph learning
- Point cloud **reconstruction** from graph spectral prior
- Point cloud **analysis** in the graph spectral domain
- Summary and future works

Point Cloud Attacks in the Graph Spectral Domain

- **Problem:** Deep learning models have shown to be vulnerable to adversarial attacks, while **adversarial attacks on 3D point clouds** are still relatively under-explored.

- **Previous works: Attacks in the data domain**
  - **Limitation**: neglect the geometric characteristics of point clouds, which makes the perturbed point clouds perceivable to humans

- **Contributions:**
  - We propose a novel paradigm of point cloud attacks—**Graph Spectral Domain Attack (GSDA)**
  - Provide in-depth graph spectral analysis of point clouds



Qianjiang Hu, Daizong Liu, and Wei Hu, "Exploring the devil in graph spectral domain for 3D point cloud attacks," *ECCV*, 2022.

Reminder: Why Graph Fourier Transform

- Offer compact transform domain representation

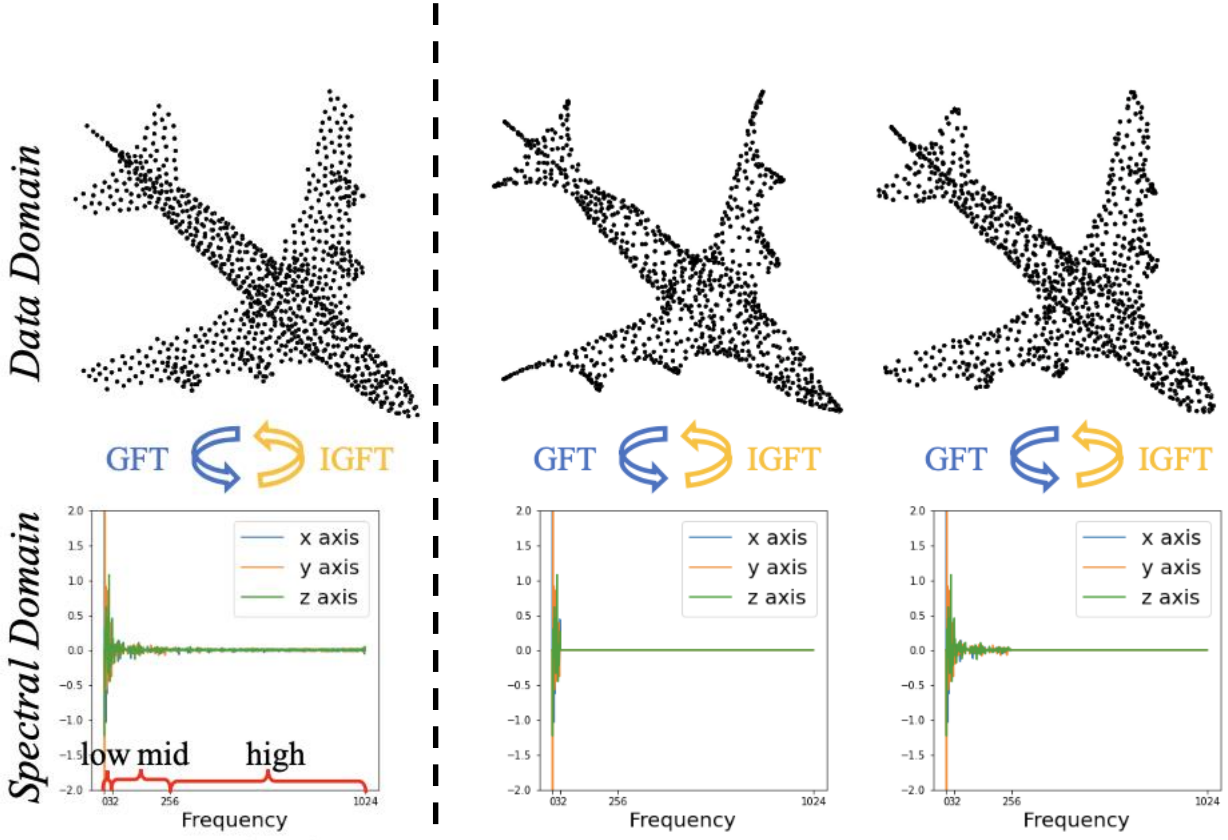
(a) Original. (b) 10 frequencies. (c) 100 frequencies. (d) 400 frequencies. (e) Graph spectral distribution.

- Reason: the graph adaptively captures the correlation in the graph signal

- $\approx$ KLT for a family of statistical models

Wei Hu, Gene Cheung, Antonio Ortega, Oscar C. Au, "Multi-resolution Graph Fourier Transform for Compression of Piecewise Smooth Images," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 419-433, January 2015.

Key Idea – Attack in the Graph Spectral Domain

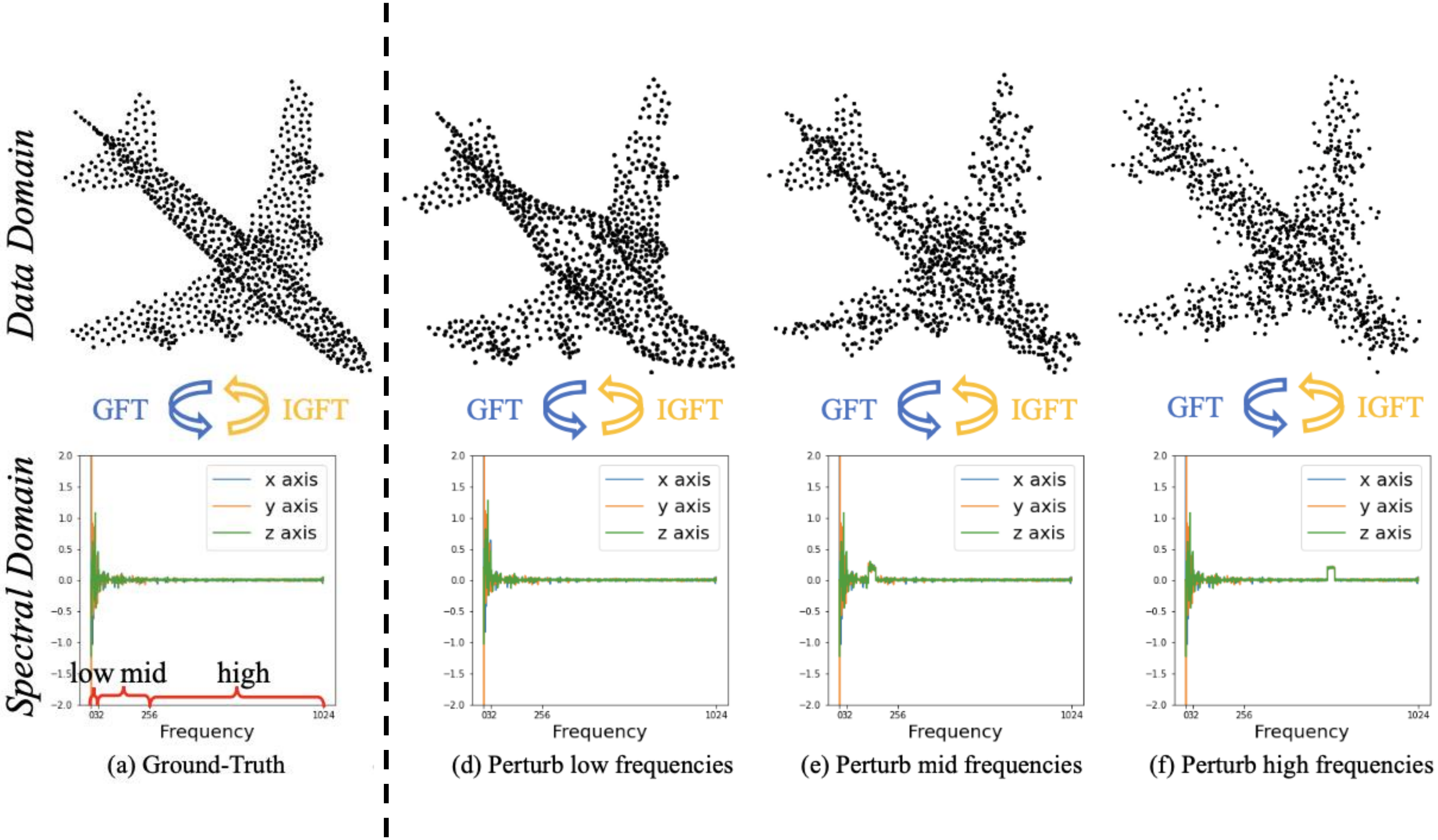


(a) Ground-Truth

(b) Remove mid-high frequencies

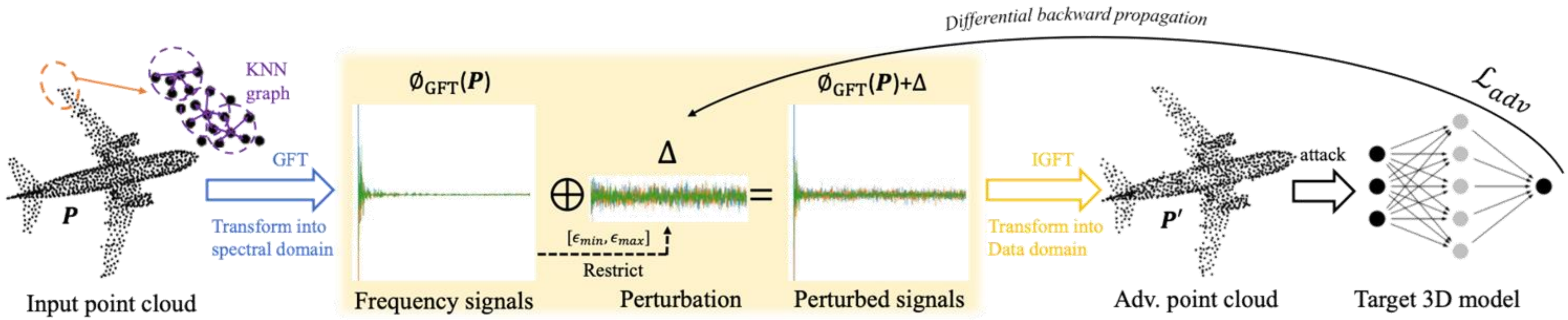
(c) Remove high frequencies

Key Idea – Attack in the Graph Spectral Domain



Key Idea – Attack in the Graph Spectral Domain

- Given a clean point cloud $\boldsymbol{P} = \{\boldsymbol{p}_i\}_{i=1}^{n} \in \mathbb{R}^{n \times 3}$ and a well-trained classifier $f(\cdot)$, the accurate label $y = f(\boldsymbol{P})$, output an adversarial point cloud $\boldsymbol{P}'$ that $f(\boldsymbol{P}') = y'$



$$\min_{\boldsymbol{\Delta}} \mathcal{L}_{adv}(\boldsymbol{P}', \boldsymbol{P}, y), \text{s.t.} ||\phi_{\mathrm{GFT}}(\boldsymbol{P}') - \phi_{\mathrm{GFT}}(\boldsymbol{P})||_p < \epsilon,$$

$$\text{where } \boldsymbol{P}' = \phi_{\mathrm{IGFT}}(\phi_{\mathrm{GFT}}(\boldsymbol{P}) + \boldsymbol{\Delta}), \quad \phi_{\mathrm{GFT}}(\boldsymbol{P}) = \boldsymbol{U}^{\top}\boldsymbol{P}$$

Qianjiang Hu, Daizong Liu, and Wei Hu, "Exploring the devil in graph spectral domain for 3D point cloud attacks," *ECCV*, 2022.

Results: perturbation sizes of adversarial point clouds

北京大學 PEKING UNIVERSITY

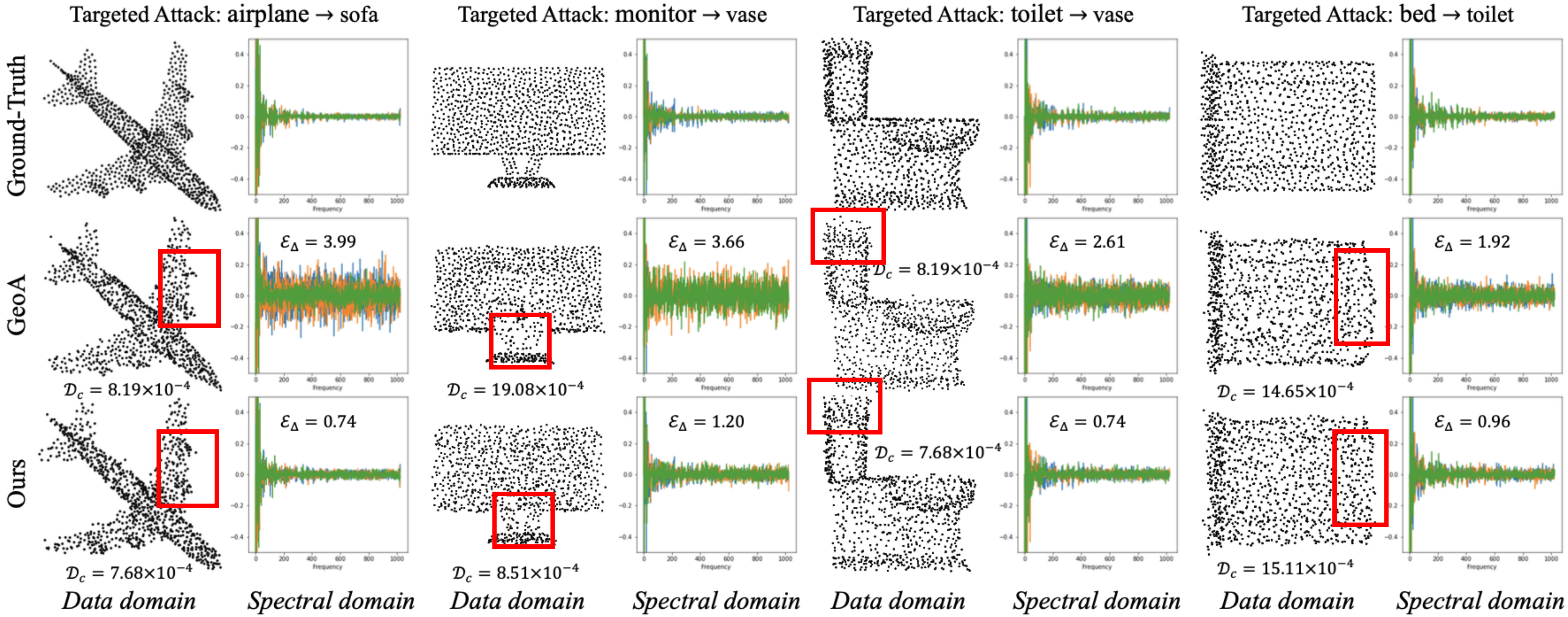
Dataset: ModelNet40

3D Models:

- PointNet
- PointNet++
- DGCNN

Attack Model	Methods	Success Rate	Perturbation Size		
			$\mathcal{D}_{norm}$	$\mathcal{D}_c$	$\mathcal{D}_h$
PointNet	FGSM	100%	0.7936	0.1326	0.1853
	3D-ADV^p	100%	0.3032	**0.0003**	0.0105
	3D-ADV^c	92.1%	-	0.1652	-
	3D-ADV^o	81.9%	-	0.1321	-
	GeoA	100%	0.4385	0.0064	0.0175
	Ours	100%	**0.1741**	0.0007	**0.0031**
PointNet++	FGSM	100%	0.8357	0.1682	0.2275
	3D-ADV^p	100%	0.3248	**0.0005**	0.0381
	GeoA	100%	0.4772	0.0198	0.0357
	Ours	100%	**0.2072**	0.0081	**0.0248**
DGCNN	FGSM	100%	0.8549	0.189	0.2506
	3D-ADV^p	100%	0.3326	**0.0005**	0.0475
	GeoA	100%	0.4933	0.0176	**0.0402**
	Ours	100%	**0.2160**	0.0104	0.1401

Results: visualization



Wen, Y., Lin, J., Chen, K., Chen, C.P., Jia, K., "Geometry-aware generation of adversarial point clouds," *IEEE TPAMI*, 2020.

Outline

- Introduction to geometric data processing and analysis over graphs
- Basics in Graph Signal Processing and Graph-based Machine Learning
- Point cloud **representation** from feature graph learning
- Point cloud **reconstruction** from graph spectral prior
- Point cloud **analysis** in the graph spectral domain
- Summary and future works

Summary

- Graph is flexible abstraction of geometric data residing on **irregular** domains

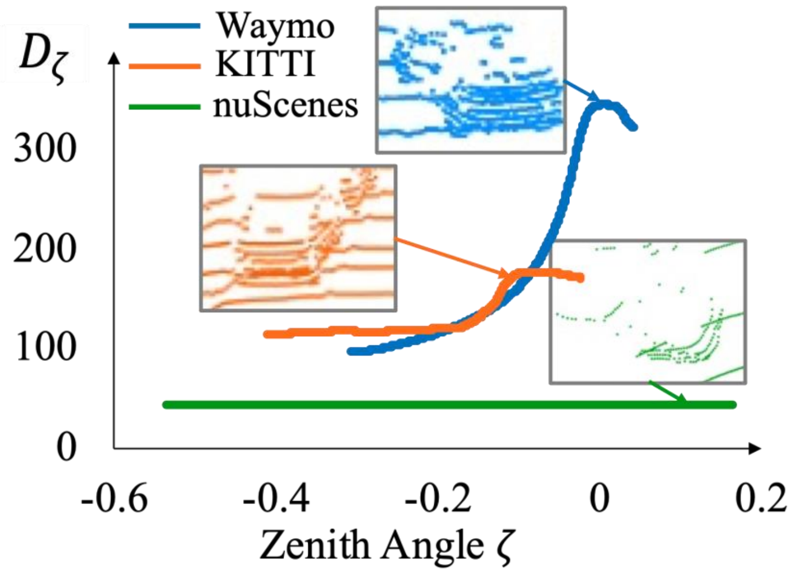
- Propose graph spectral methods for **robust** & **interpretable** processing and analysis
    - Point cloud representation: feature graph learning
    - Point cloud reconstruction: graph spectral prior
    - Point cloud analysis: low-pass property in the graph spectral domain

- Achieve efficient, robust and interpretable geometric data processing & analysis!

Ongoing & Future Works

- GSP for enhancing model interpretability
  - ➢ e.g., the effect of graph optimization on the depth of GNNs

- Generalization / domain adaptation of point cloud learning
  - ➢ Out-Of-Distribution, domain adaptation in Lidar point clouds



- Functional brain network analysis with GSP & GNNs
  - ➢ e.g., neuron classification

Acknowledgement

PEKING UNIVERSITY

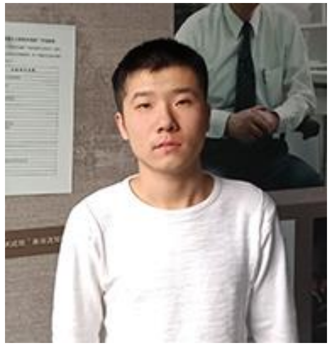
Prof. Gene Cheung
(York University, Canada)

Prof. Antonio Ortega
(USC, USA)

Dr. Anthony Vetro
(MERL, USA)

Prof. Chia-Wen Lin
(NTHU, Taiwan)

Zhimin Zhang
(Peking University)



Xiang Gao
(Peking University)

Shitong Luo
(Peking University)

Qianjiang Hu
(Peking University)

Thank you!

Homepage: https://www.wict.pku.edu.cn/huwei/
Email: forhuwei@pku.edu.cn