Data Compression Conference

南京大學 NANJING UNIVERSITY

Multiscale Point Cloud Geometry Compression

Jianqiang Wang[1], Dandan Ding[2], Zhu Li[3], Zhan Ma[1]

[1]Nanjing University, [2]Hangzhou Normal Univ., [3]Univ. of Missouri at Kansas

Background: Demand for Point Cloud Compression

NANJING UNIVERSITY

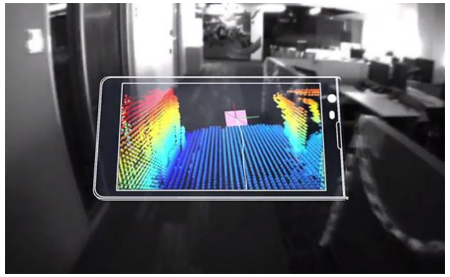
□ Point Cloud (PC)

- **Geometry** (x, y, z)
- Attribute (color, etc)

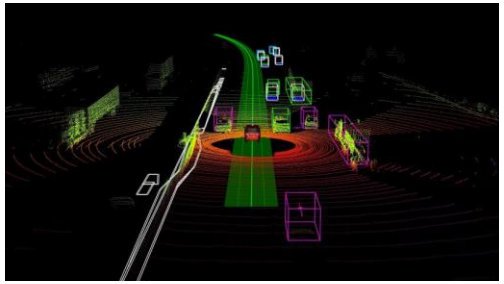
Various PCs

□ Emerging Applications

- Immersive Media (AR/VR)
- Autonomous Driving



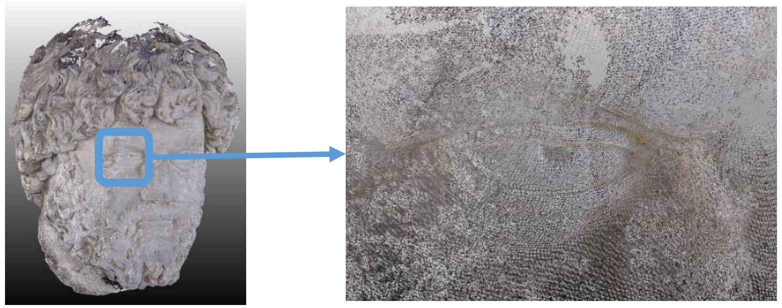
AR\VR



Autonomous Driving

□ Demand for Compression

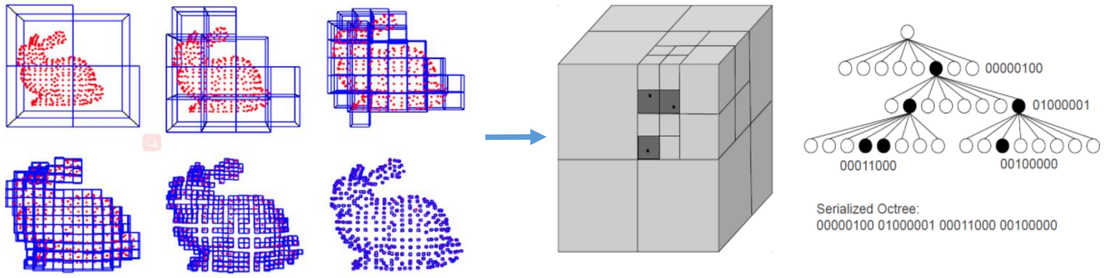
- Huge amount of data
- Unordered and unstructured data



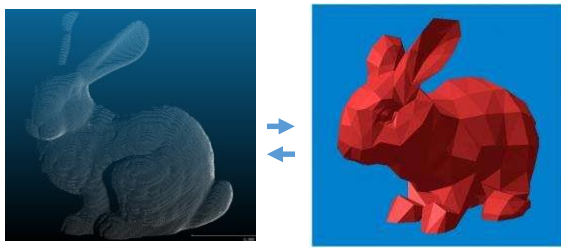
Unstructured Data

Background: PCC Methods

- **Geometry based PCC (G-PCC)**
  - **Octree Geometry Codec**
    - Occupancy Information Coding
  - **Trisoup Geometry Codec**
    - Triangle Mesh Vertices Coding

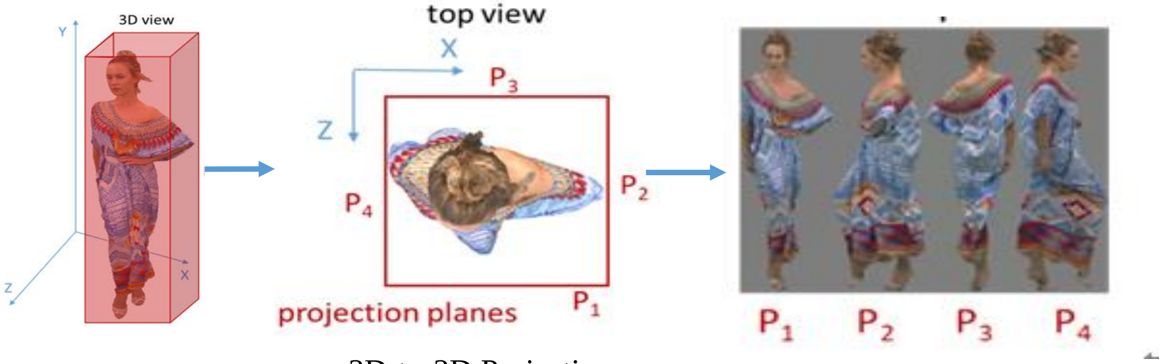


Octree Decomposition & Occupancy Information Coding



Point Cloud & Mesh Model

- **Video based PCC (V-PCC)**
  - 3D-2D Projection
  - Image/Video Coding
  - State-of-the-art Efficiency



3D-to-2D Projections

Background: Emerging Learning based Methods

☐ Voxel based Methods

- PC → Volumetric Model
- 3D CNN based Transform
- Classification Loss

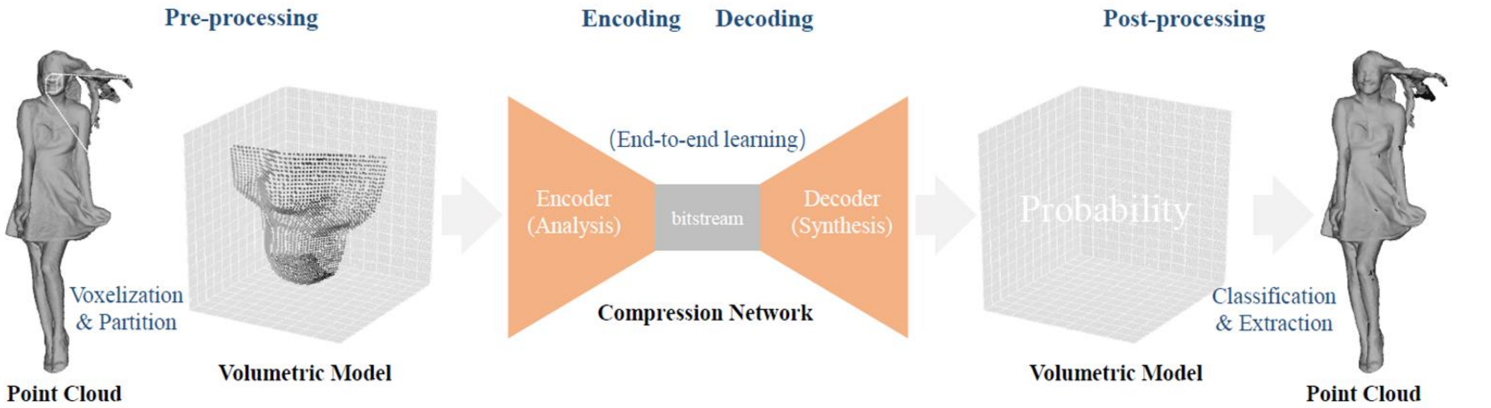
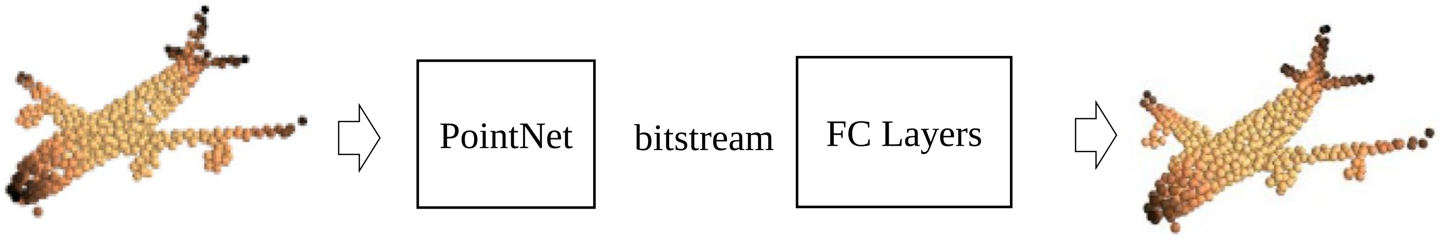


Figure from Wang et al, "Lossy Point Cloud Geometry Compression via End-to-End Learning." accepted by IEEE TCSVT, Jan. 2021

☐ Point based Methods

- Input Raw Points
- PointNet Structure
- Distance Loss



PCC based on PointNet AE

Background: Emerging Learning based Methods

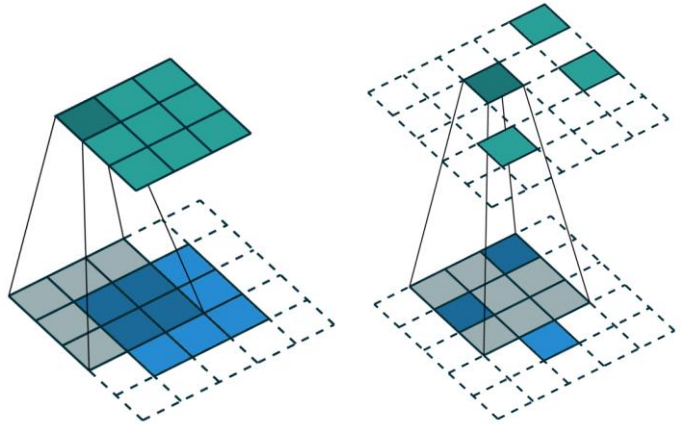
NANJING UNIVERSITY

Drawbacks of Existing Learning based Methods

- **Voxel based Methods**
  - Huge Computation and Memory Cost
- **Point based Methods**
  - Poor Performance at High Bit Rates

How to Improve?

- **Low Complexity Representation and Computation**
  - Sparse Tensor and Sparse Convolution
- **Geometry Details Description and Reconstruction**
  - Hierarchical reconstruction
  - Key points lossless compression



Dense Convolution vs Sparse Convolution

Figure from Choy, C. et al. "4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks." 2019 IEEE/CVF CVPR.

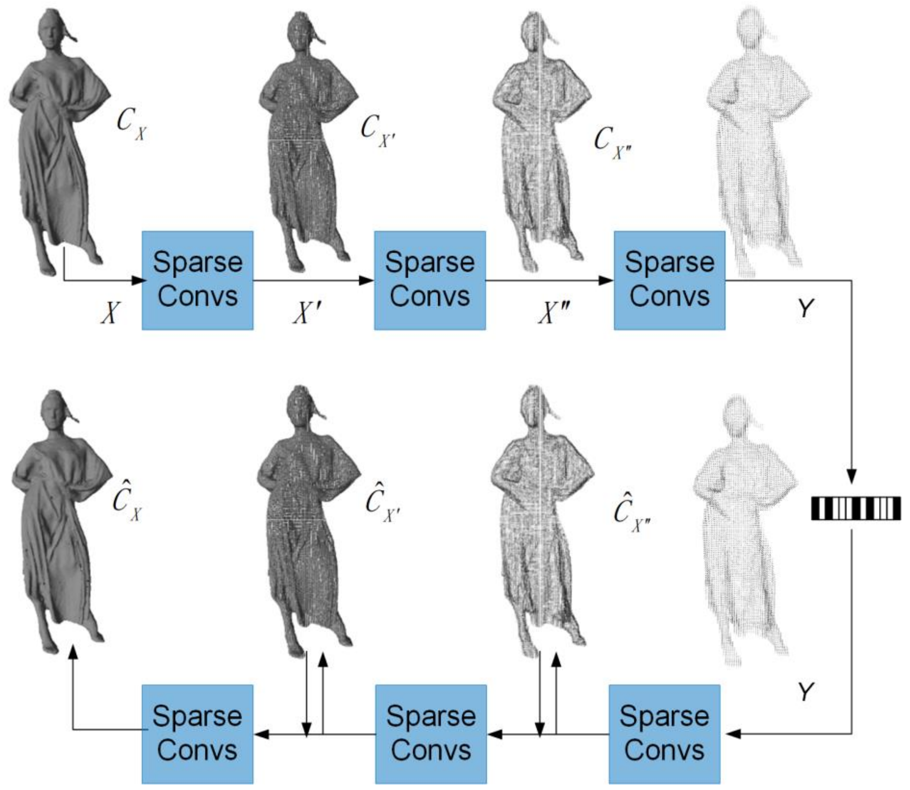
Overview of Multiscale PCGC

NANJING UNIVERSITY

☐ Pipeline

- AutoEncoder (AE) based on Sparse Convolution
- All data is in representation of **Sparse Tensor** {***C*, *F***}

Input	X	• **F**: all-ones vector
↓ Encoder: ↓	X', X'', Y	• **C**: Downscaled Coordinates • **F**: Embedded Local Structures
Latent Representation	Y	• **C** & **F** will be Entropy Encoded into a Binary String
↓ Decoder: ↓	X'', X', X	• **C** & **F** will be decoded and hierarchically Reconstructed
Output	X	• **C** is decoded based on Binary Classification

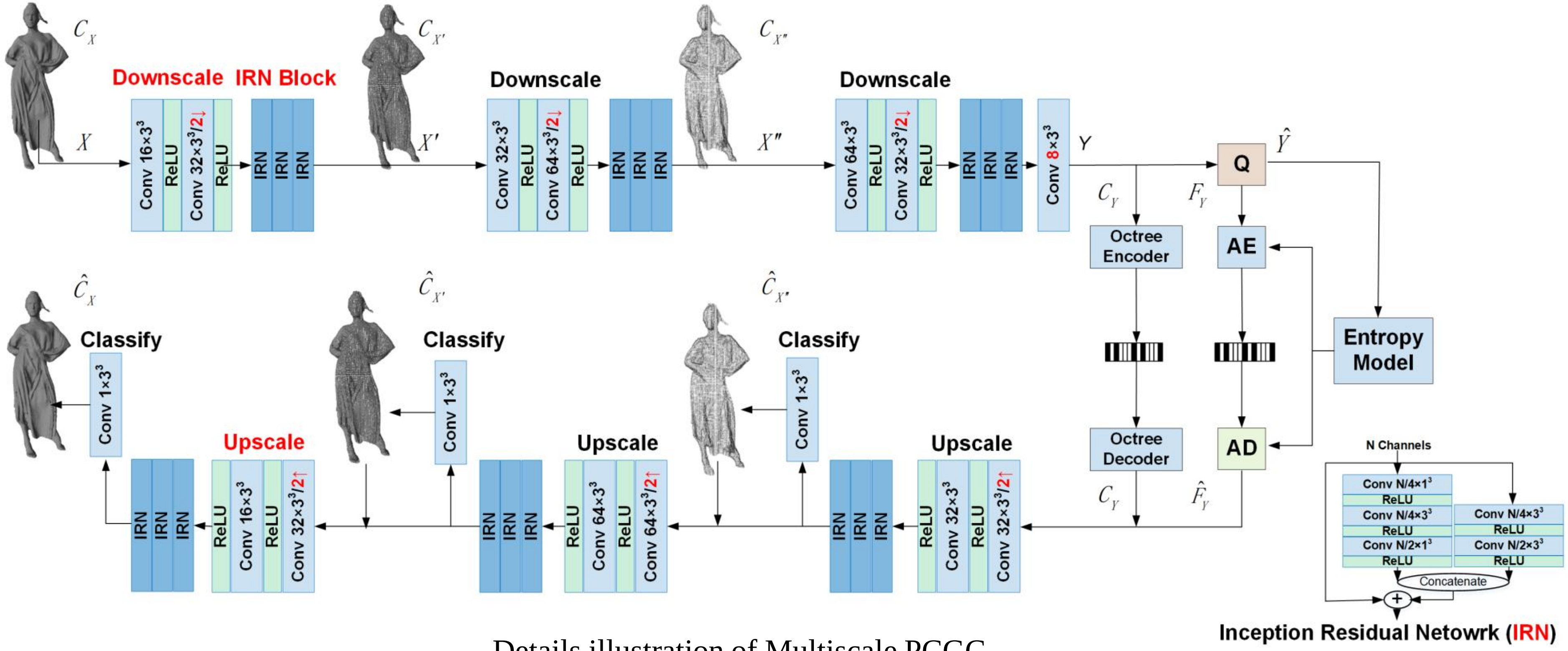


Overview of Multiscale PCGC

Sparse Convolution based Multiscale Resampling

◻ Network Details

- **Basic Unit**: Inception Residual Network (IRN)
- **Down-scaling**: Convolution with a stride of two
- **Up-scaling**: Transpose convolution with a stride of two



Details illustration of Multiscale PCGC

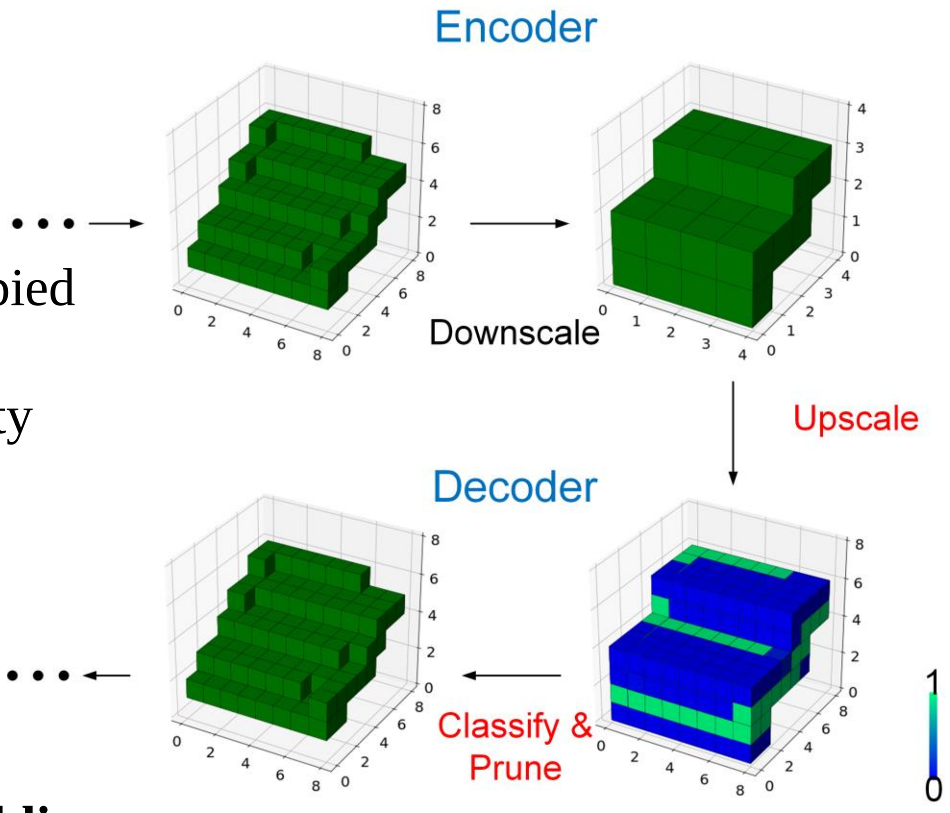
Binary Classification based Hierarchical Reconstruction

NANJING UNIVERSITY

Steps

For each scale:

- **Upsampling**
  - Generate 8 sub-voxels from 1 voxel
- **Feature Augmentation**
  - Generate the probabilities of voxel-being-occupied
- **Classifying & Pruning**
  - Whether generated voxels are occupied or empty



Binary Classification based Reconstruction

Training Loss

- Multiscale Binary Cross Entropy (**BCE)** Loss

Inferring Method

- Binary Classification based on **Adaptive Thresholding**

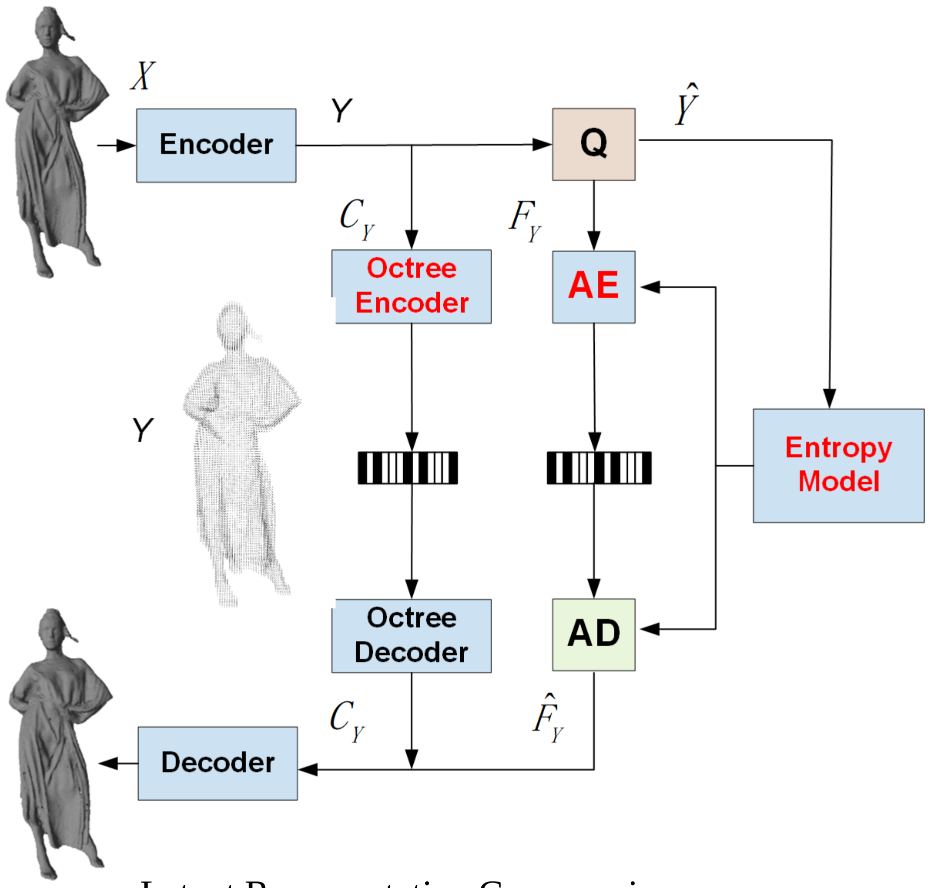
Latent Representation Compression

☐ Coordinates C_Y

- Skeleton Key points
- Lossless Compression using **Octree** Codec
- $\approx$ 0.025bpp

☐ Features F_Y

- Implicitly Embedded Geometry Features
- Lossy Compression using Arithmetic Encoding (**AE**)
- Entropy Model based on Factorized Prior
  - Autoregressive Prior and Hyper prior can achieve additional BD-Rate gains



Latent Representation Compression

Experimental results

☐ Experiment Settings

■ **Dataset**

- Training: ShapeNet
- Testing: 8iVFB, Owlii dataset, MVUB

■ **Anchors**

- G-PCC: octree、trisoup
- V-PCC
- Learned-PCGC

■ **Objective Metrics**

- Point-to-point Distance (D1)
- Point-to-plane Distance (D2)

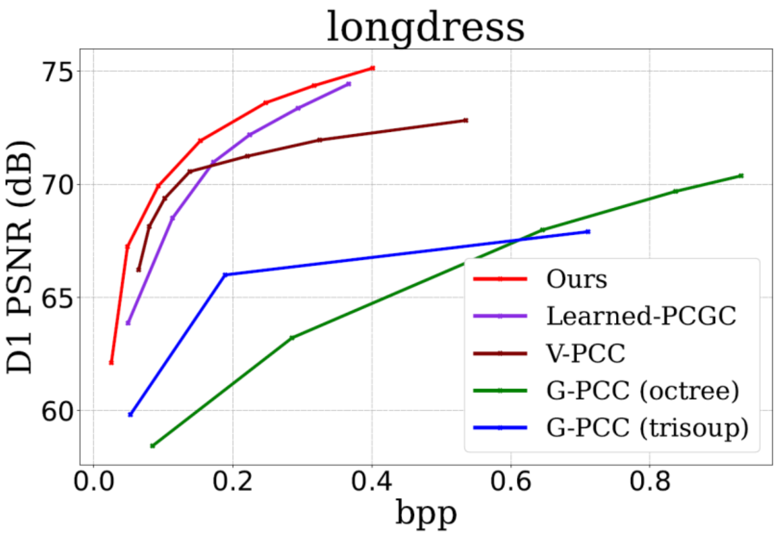
☐ Performance Evaluation

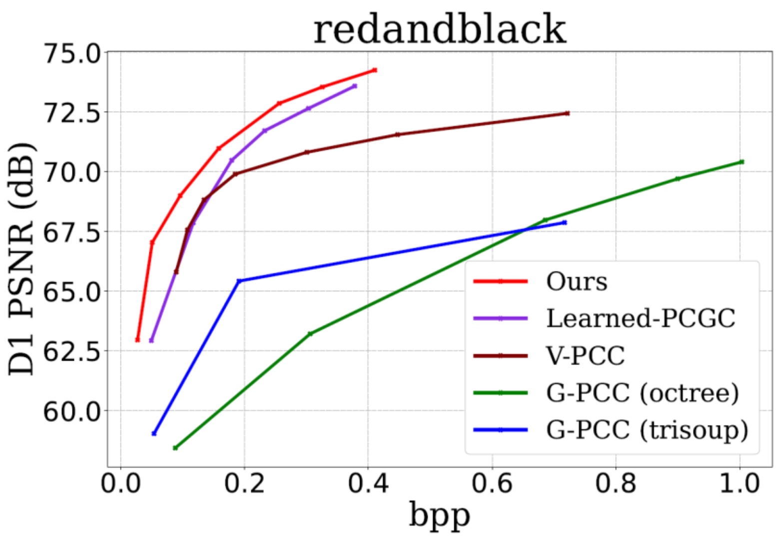
- ≈ **40%** BD-Rate gains against **V-PCC**
- > **70%** BD-Rate gains against **G-PCC**
- > **30%** BD-Rate gains against **Learned PCGC**

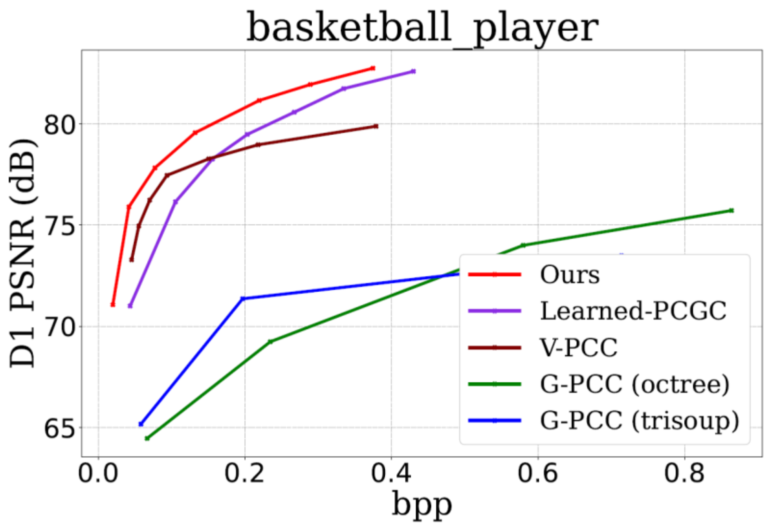
BD-Rate Gains against other compression methods using D1 and D2 distortion measurements.

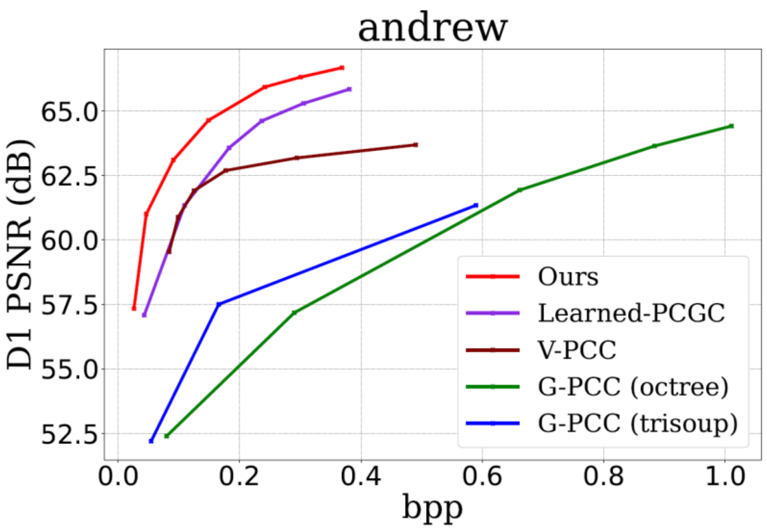
	V-PCC		G-PCC (octree)		G-PCC (trisoup)		Learned-PCGC [4]	
	D1	D2	D1	D2	D1	D2	D1	D2
8iVFB	-39.4	-41.8	-90.8	-84.7	-78.5	-72.8	-36.4	-32.6
Owlii	-38.4	-35.2	-94.8	-90.5	-90.9	-79.1	-50.8	-43.8
MVUB	-60.6	-53.5	-90.4	-83.4	-87.8	-79.7	-46.7	-39.0
Average	**-47.7**	**-45.1**	**-91.5**	**-85.4**	**-84.7**	**-76.9**	**-43.4**	**-37.4**

Objective Comparison

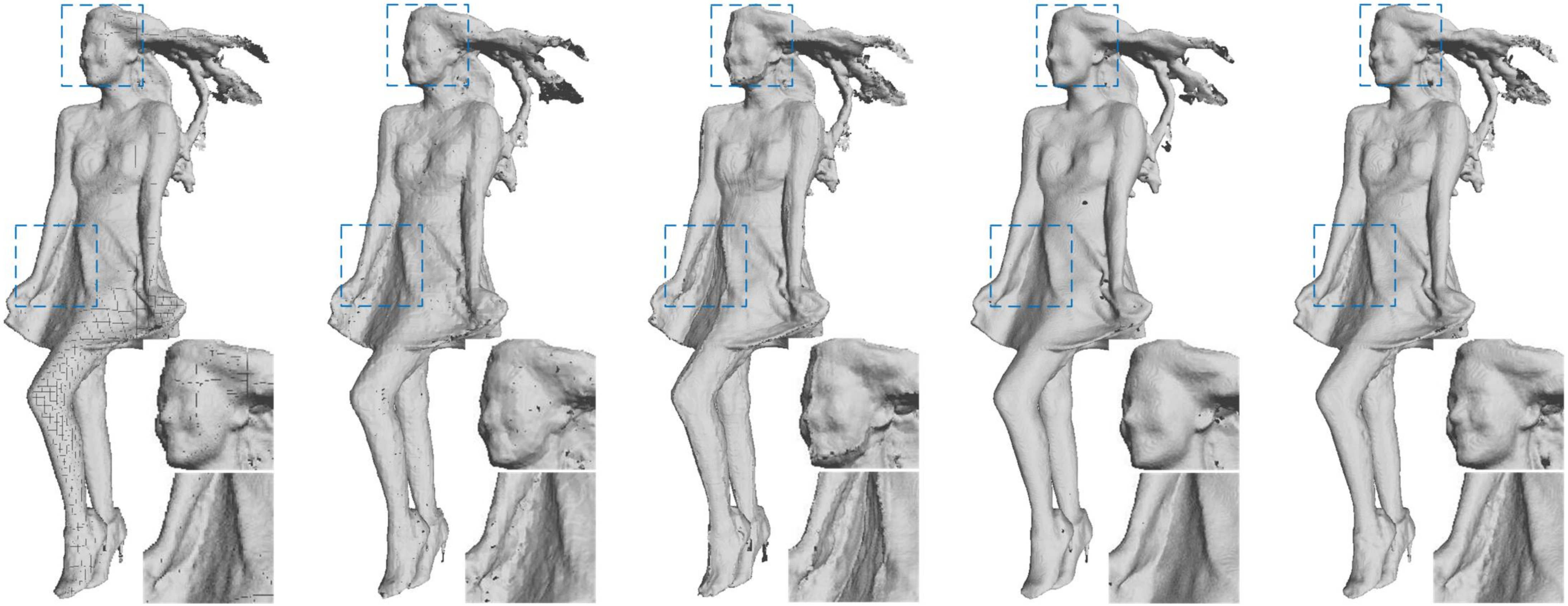








Subjective Comparison



G-PCC (octree)
211242
0.305bpp 63.20dB

G-PCC (trisoup)
559054
0.190bpp 65.42dB

V-PCC
791645
0.184bpp 69.89dB

Ours
757691
0.157bpp 70.97dB

Ground Truth
757691 points

Complexity Comparison

NANJING UNIVERSITY

Runtime comparison of different methods

	V-PCC	G-PCC (octree)	G-PCC (trisoup)	Learned-PCGC	Ours (Update)
Enc (s)	103.4	**1.6**	8.1	9.3	~~1.6~~ (**0.80**)
Dec (s)	**0.7**	**0.6**	6.6	9.5	~~5.4~~ (**0.82**)

Benchmarked on a workstation with an Intel Core i7-8700 CPU and Geforce GTX 1070 GPU

Complexity comparison of multi-scale and single-scale reconstruction

	Multi-scale reconstruction / Single-scale reconstruction
running memory	**1/4**
runtime	**1/3**

Contribution & Future Works

NANJING UNIVERSITY

Contribution

- Introduce a novel **multiscale PCGC** method based on **sparse convolution**.
- The proposed method is more **computationally and memory efficient** than the previous Learned PCGC.
- The proposed method achieves > **40%** BD-Rate gains over the SOTA **V-PCC**.

Future Woks

- Extend this work to include the color attributes and sparse PCs.
- Use quality measures that better match subjective perception.

Thank you for your attention!

More details can be found on https://njuvision.github.io/PCGCv2/