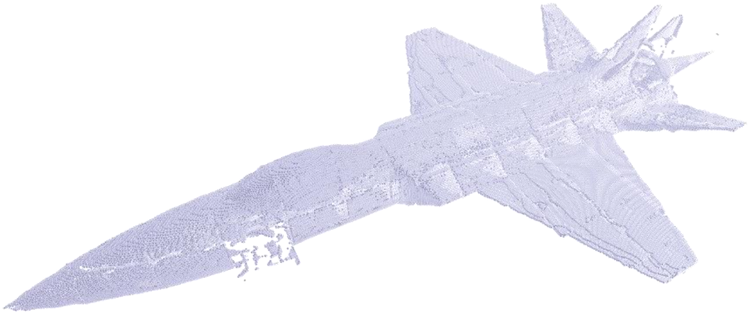
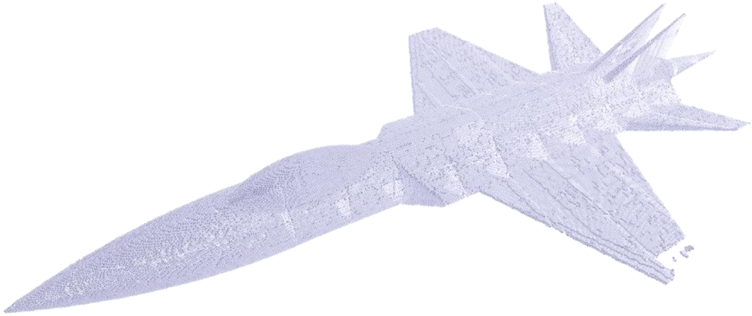
Quantitative Comparison of Point Cloud Compression Algorithms





Cheng Hao Wu (chenghao.nthu@gmail.com)

Advisor: Cheng-Hsin Hsu

Networking and Multimedia Systems Lab,
ISA, National Tsing Hua University

NMSL@NTHU
Networking and Multimedia Systems Lab

Outline

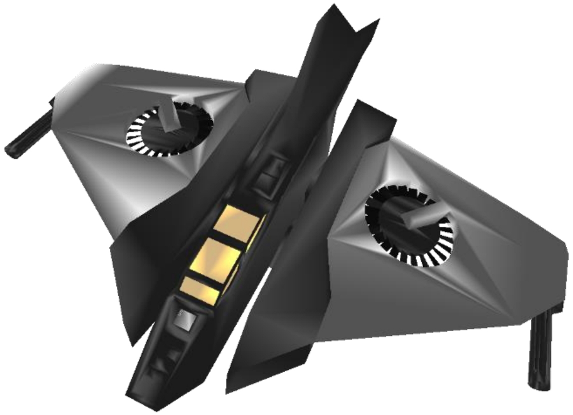
- Introduction
- Challenges
- Implementations
- Experimental Setup
- Objective Results
- Subjective Results
- Future of NN-based PCC algorithms
- Conclusion

INTRODUCTION

3D Representations

Meshes

- Better efficiency on rendering due to hardware acceleration and optimization
- Widely used in entertainment content industry



Point Clouds

- Native data format of the capture equipment
- No correlations among points
- Optional attributes
  - Colors
  - Normals
  - Reflectance

Applications Relying on Point Clouds

Holographic Telepresence

6DoF VR

AR applications on end devices

Scene Reconstruction

- For native objects, point clouds are more suitable than meshes
  - Save the computational overhead from converting point clouds to meshes
- Acceptable Visual Quality → 4 Gbps[1] (one object)

Point Cloud Compression (PCC) is essential

[1] C. Cao, M. Preda, and T. Zaharia, "3D point cloud compression: A survey," ACM International Conference on 3D Web Technology (Web3D'19), pages 1–9, July 2019.

Common PCC Algorithms

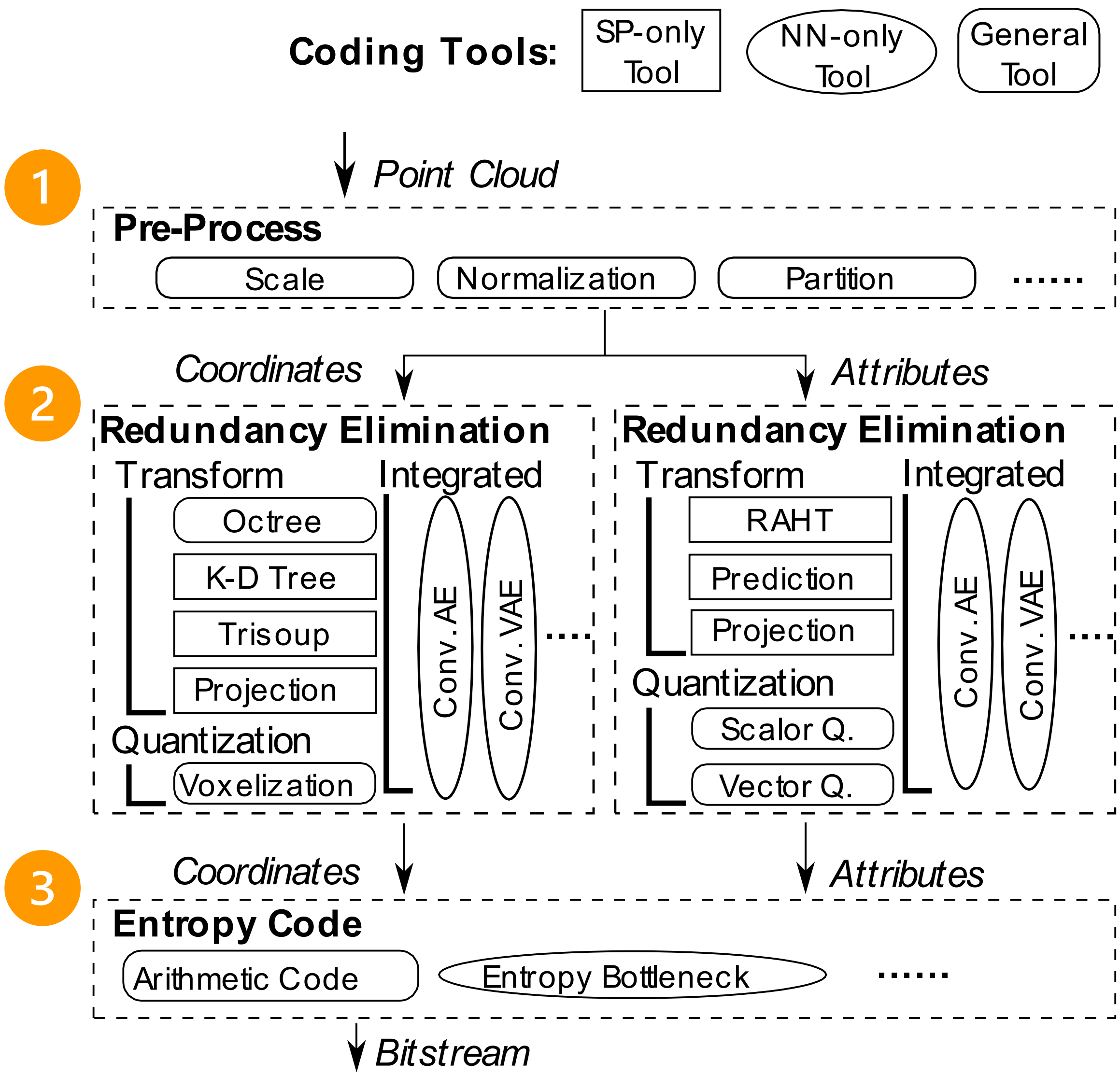
Signal Processing-based (SP-based)

- Relies on conventional techniques like, transformation, quantization, and entropy coding
  - Octree
  - K-d tree
  - Voxelization

Neural Network-based (NN-based)

- Takes advantages on feature extraction
  - AutoEncoder
  - Variational AutoEncoder
  - Generative Adversarial Network

General Encoder Architecture of PCC Algorithms



CHALLENGES

Inconsistency on Performance Evaluations Scheme

- For different PCC algorithms, evaluation results are inconsistent on

Datasets

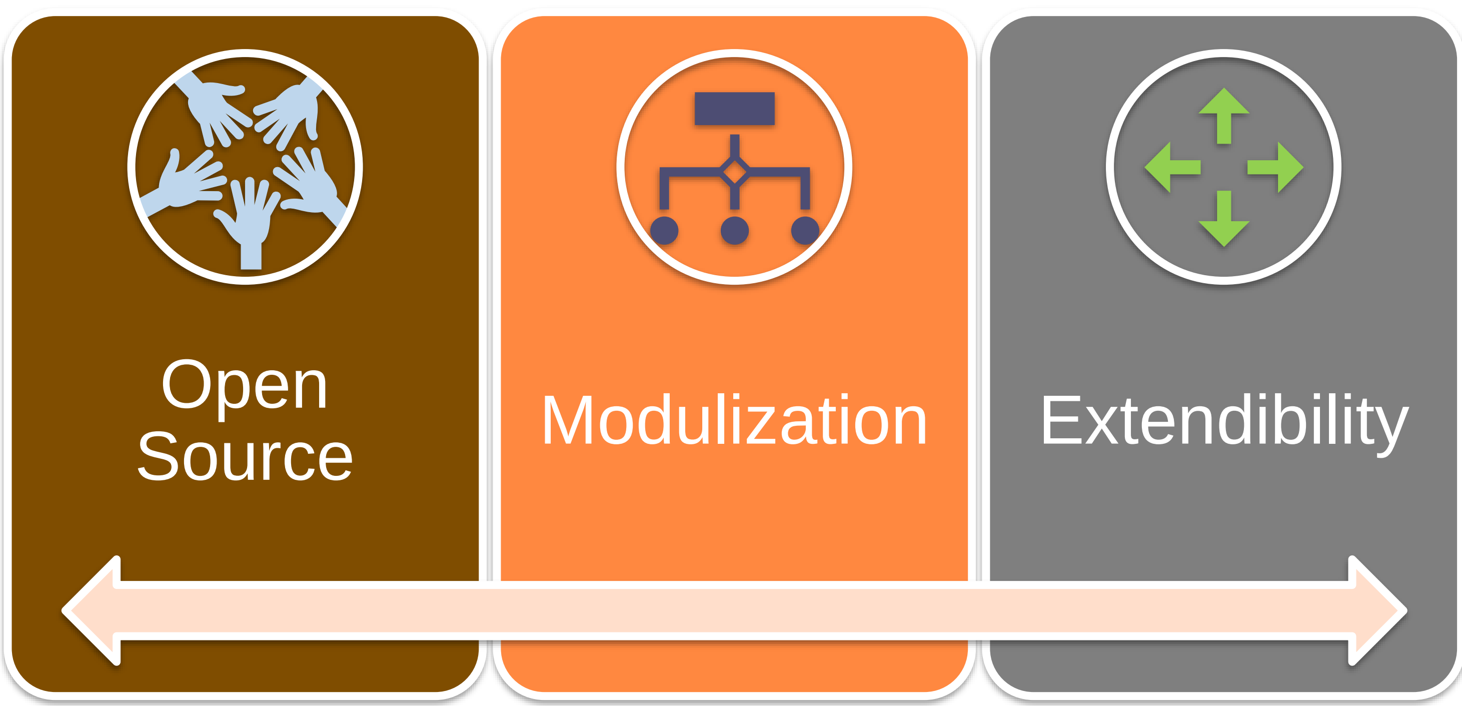
Performance Metrics

Coding Parameters

Hard to compare different PCC algorithms fairly and completely

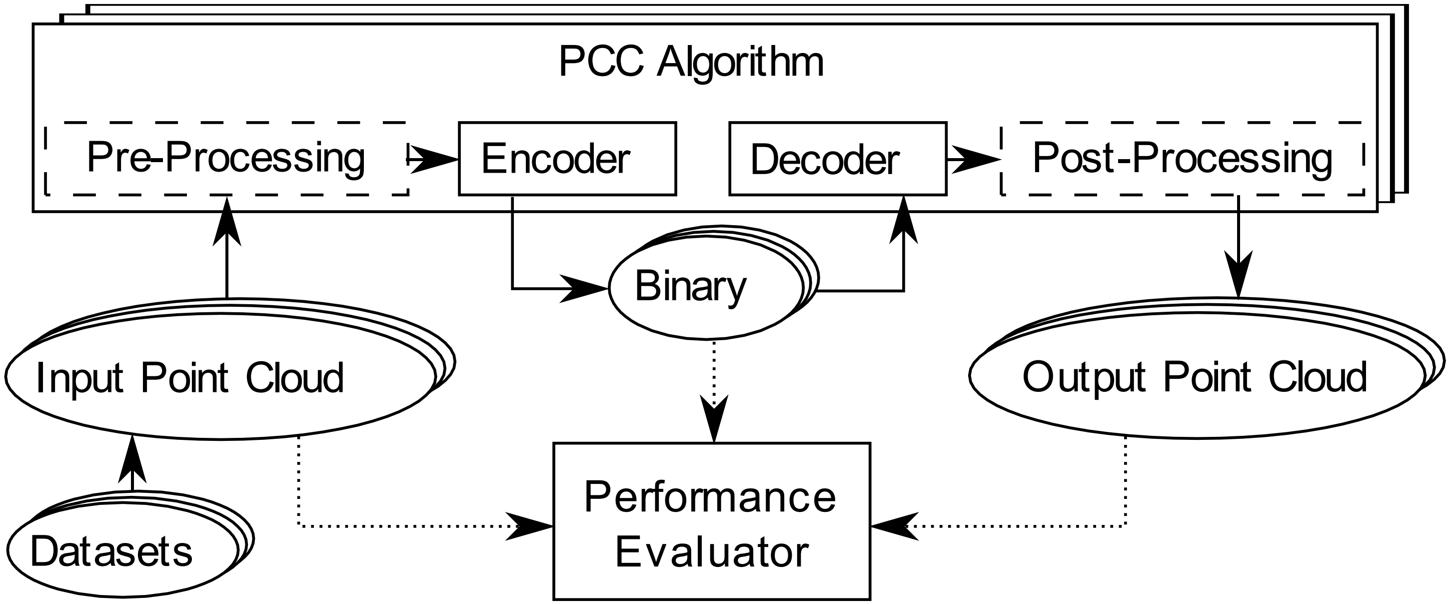
Therefore…

- We propose PCC Arena, a PCC algorithm benchmark platform [MMVE'20] and [TMM'21, submitted]
  - GitHub link: https://github.com/xtorker/PCC_Arena



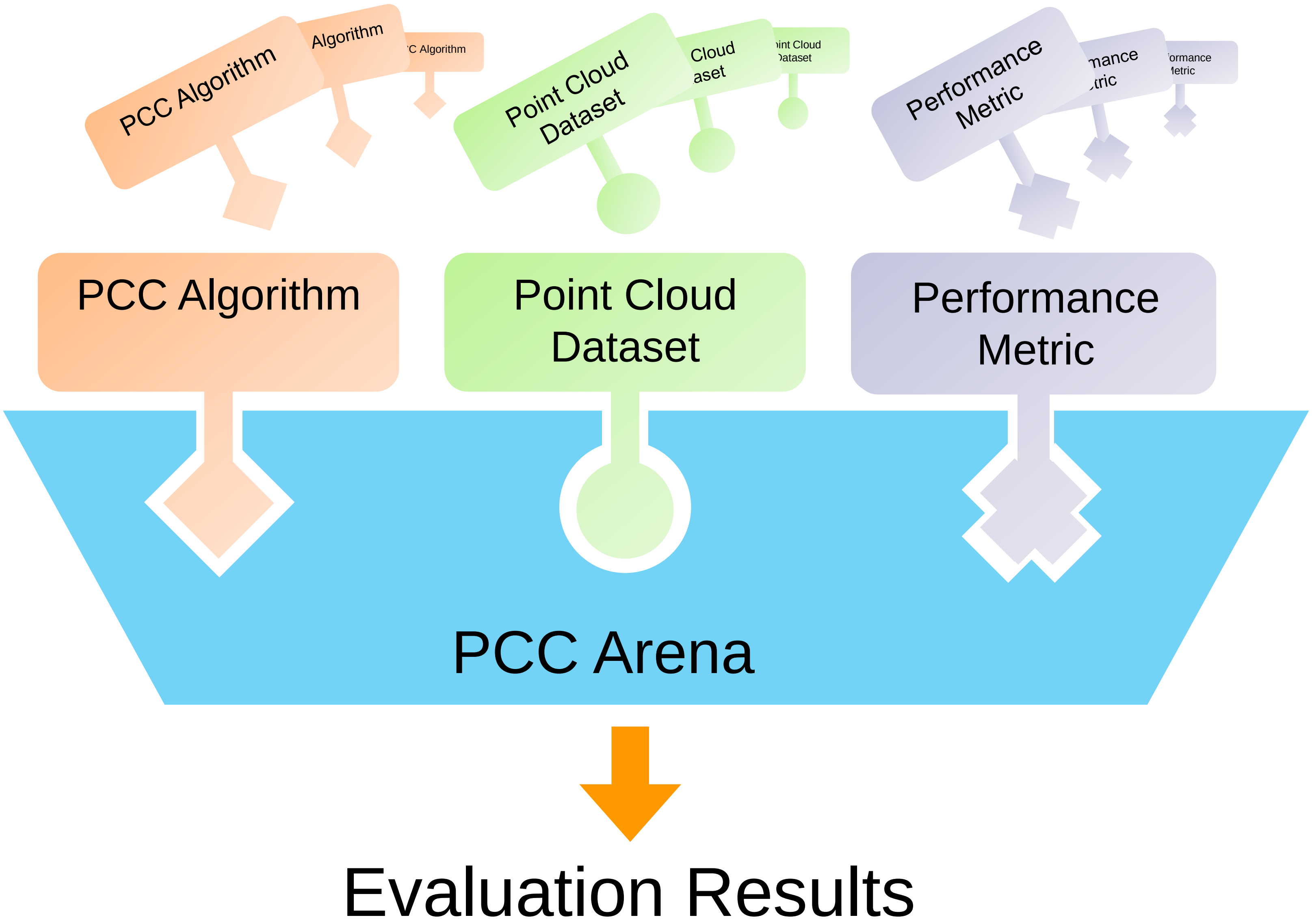
IMPLEMENTATIONS

High-Level Architecture of PCC Arena



- Each PCC algorithm has its own rate control method
- Performance evaluator analyzes the results for each
  - Input point cloud
  - PCC algorithm
  - Set of coding parameters

Extendibility of PCC Arena



EXPERIMENTAL SETUP

Performance Metrics

- Non-visual Metrics
  - bpp (bits-per-point)
  - Running time (Encoding/Decoding)

- 2D Visual Metrics (render 6 2D images along x, y, z axes)
  - PSNR
  - SSIM
- 3D Visual Metrics: Coordinates
- 3D Visual Metrics: With Colors

3D Visual Metrics: Coordinates

$\mathbf{P}_{\mathrm{r}}$: reference point cloud $\mathbf{P}_{\mathrm{t}}$: target point cloud

- Asymmetric Chamfer Distance (ACD)

$$\mathrm{ACD}(\mathbf{P}_1, \mathbf{P}_2) = \frac{1}{|\mathbf{P}_1|} \sum_{p \in \mathbf{P}_1} \min_{p\prime \in \mathbf{P}_2} \|p - p\prime\|_2^2$$

$$\mathrm{ACD}_{\mathrm{rt}} = \mathrm{ACD}(\mathbf{P}_{\mathrm{r}}, \mathbf{P}_{\mathrm{t}})$$
$$\mathrm{ACD}_{\mathrm{tr}} = \mathrm{ACD}(\mathbf{P}_{\mathrm{t}}, \mathbf{P}_{\mathrm{r}})$$

- Chamfer Distance (CD)

$$\mathrm{CD} = \frac{1}{2}(\mathrm{ACD}_{\mathrm{rt}} + \mathrm{ACD}_{\mathrm{tr}})$$

Average error

- CD Peak Signal-to-Noise Ratio (CD-PSNR)

$$\text{CD-PSNR} = 10 \log_{10} \frac{M_r^2}{\mathrm{CD}}$$

M_r is the maximal distance between any two points in $\mathbf{P}_{\mathrm{r}}$

- Hausdorff Distance (HD)

$$\mathrm{HD} = \max\left(\max_{p \in \mathbf{P}_{\mathrm{r}}}\left(\min_{p\prime \in \mathbf{P}_{\mathrm{t}}} \|p - p\prime\|_2^2\right), \max_{p\prime \in \mathbf{P}_{\mathrm{t}}}\left(\min_{p \in \mathbf{P}_{\mathrm{r}}} \|p - p\prime\|_2^2\right)\right)$$

Largest error

Two Definitions of Distance

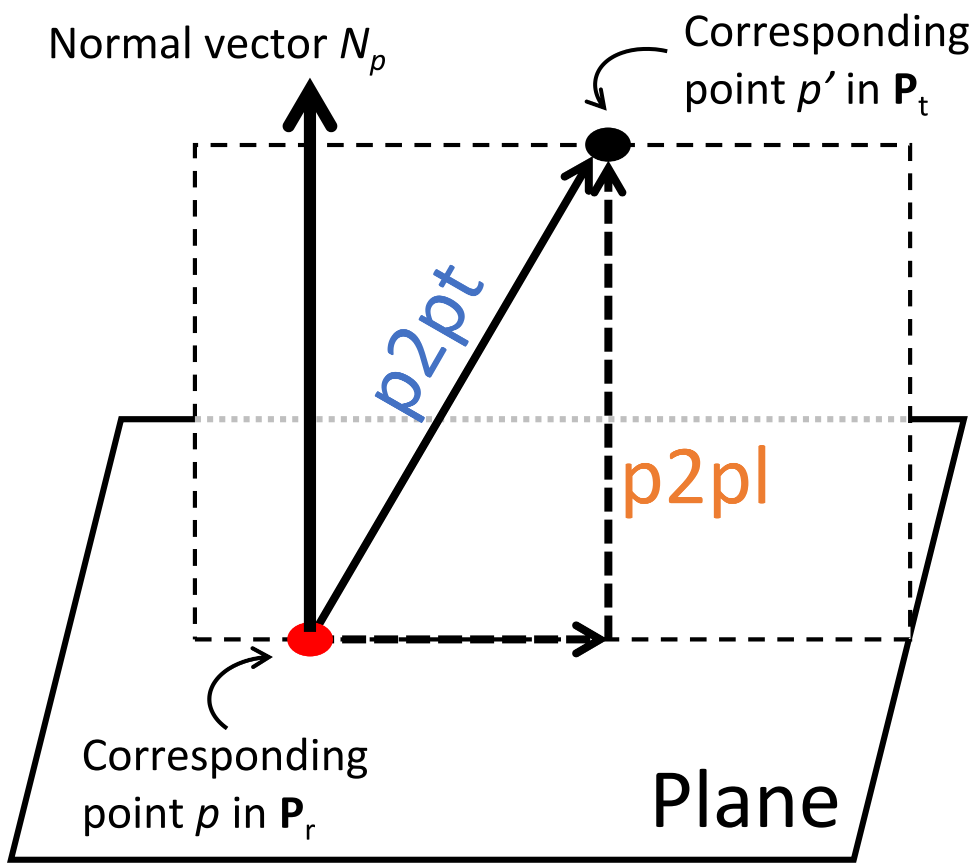
- Point-to-point (p2pt)

$$\|p - p'\|_2$$

- Point-to-plane (p2pl)[1]

$$(p - p') \cdot N_p$$

N_p is the normal vector of the plane of $\mathbf{P}_r$ that contains p



[1] D. Tian, H. Ochimizu, C. Feng, R. Cohen and A. Vetro, "Geometric distortion metrics for point cloud compression," *IEEE International Conference on Image Processing (ICIP'17)*, pp. 3460-3464, September 2017

3D Visual Metrics: With Colors

- Luminance Color PSNR (L-CPSNR)
  - PSNR on luminance channel with MSE as distance

$$\text{L-CPSNR} = 10 \log_{10} \frac{M^2}{\text{L-MSE}(\mathbf{P}_{\mathrm{r}}, \mathbf{P}_{\mathrm{t}})}$$

- Viola et al.'s QoE (VQoE)[1]
  - QoE metric
  - Consider both coordinate and color
  - Empirical derived α=0.6597

$$\text{VQoE} = \alpha \cdot \text{CD} + (1 - \alpha) \cdot H_{\mathrm{L}_2}^{\mathrm{Y}}$$

[1] I. Viola, S. Subramanyam, and P. Cesar, "A color-based objective quality metric for point cloud contents," IEEE International Conference on Quality of Multimedia Experience (QoMEX'20), pages 1–6, May 2020

Candidate PCC Algorithms

- SP-based
  - Draco [Google]
  - G-PCC [MPEG 3DG]
  - V-PCC [MPEG 3DG]
- NN-based
  - GeoCNNv1 [Université Paris-Saclay, FR] [ICIP'19]
  - GeoCNNv2 [Université Paris-Saclay, FR] [MMSP'20]
  - PCGCv1 [NJU, CN] [TCSVT'21]
  - PCGCv2 [NJU, CN] [DCC'21]

Rate Control

- Draco: quantization parameter qp
  - Quantize the input value to the specified bits
- G-PCC: `positionQuantizationScale`
  - Similar mechanism to Draco
- V-PCC: preset config file
  - 2D image qp value (and other parameters), recommended by MPEG
- GeoCNN/GeoCNNv2/PCGCv1/PCGCv2: different models
  - Train different models with different rate-distortion parameters

Training Process (for NN-based)

- Use pre-trained model if the authors have provided
  - PCGCv1, PCGCv2
- If not, we follow the same procedure to train the model
  - GeoCNNv1, GeoCNNv2
- Generating training dataset for all NN-based PCC algorithms with SNC (mesh)
  - Use scripts provided by the authors first
  - If it's not the case, use our scripts (as same as the script we used to generate the testing datasets) to generate point clouds from meshes

Testing Datasets

- Sampled from meshes with *CloudCompare*[1]
- Number of points: 500k
- Coordinates only
  - MN40 (ModelNet40)
  - SNC (ShapeNetCore)
  - CAPOD
  - 8i dataset (**avatars**)

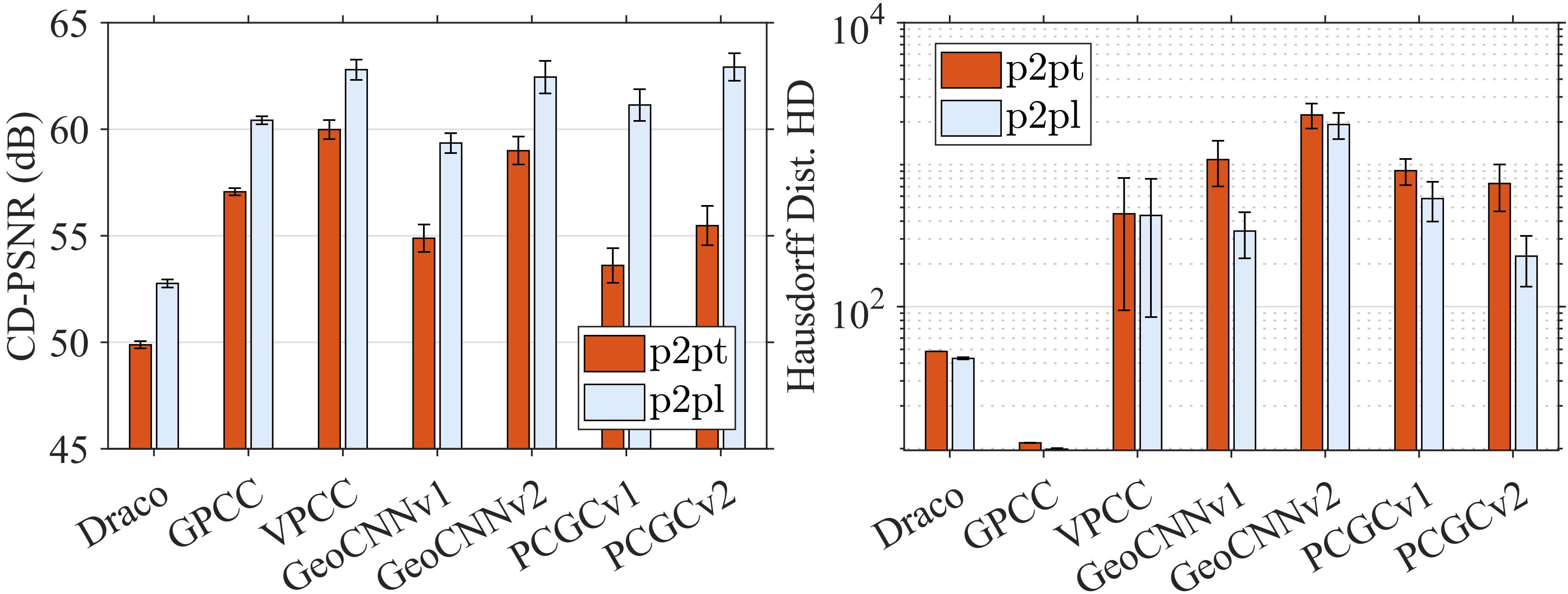
Objects

- With color
  - SNCC (ShapeNetCore with color)
  - 8iC dataset (avatars with color)
- All datasets are prepared a version with normal included for evaluation purpose (point2plane metrics)

[1] CloudCompare.org. CloudCompare - open source project. https://www.danielgm.net/cc/

OBJECTIVE RESULTS

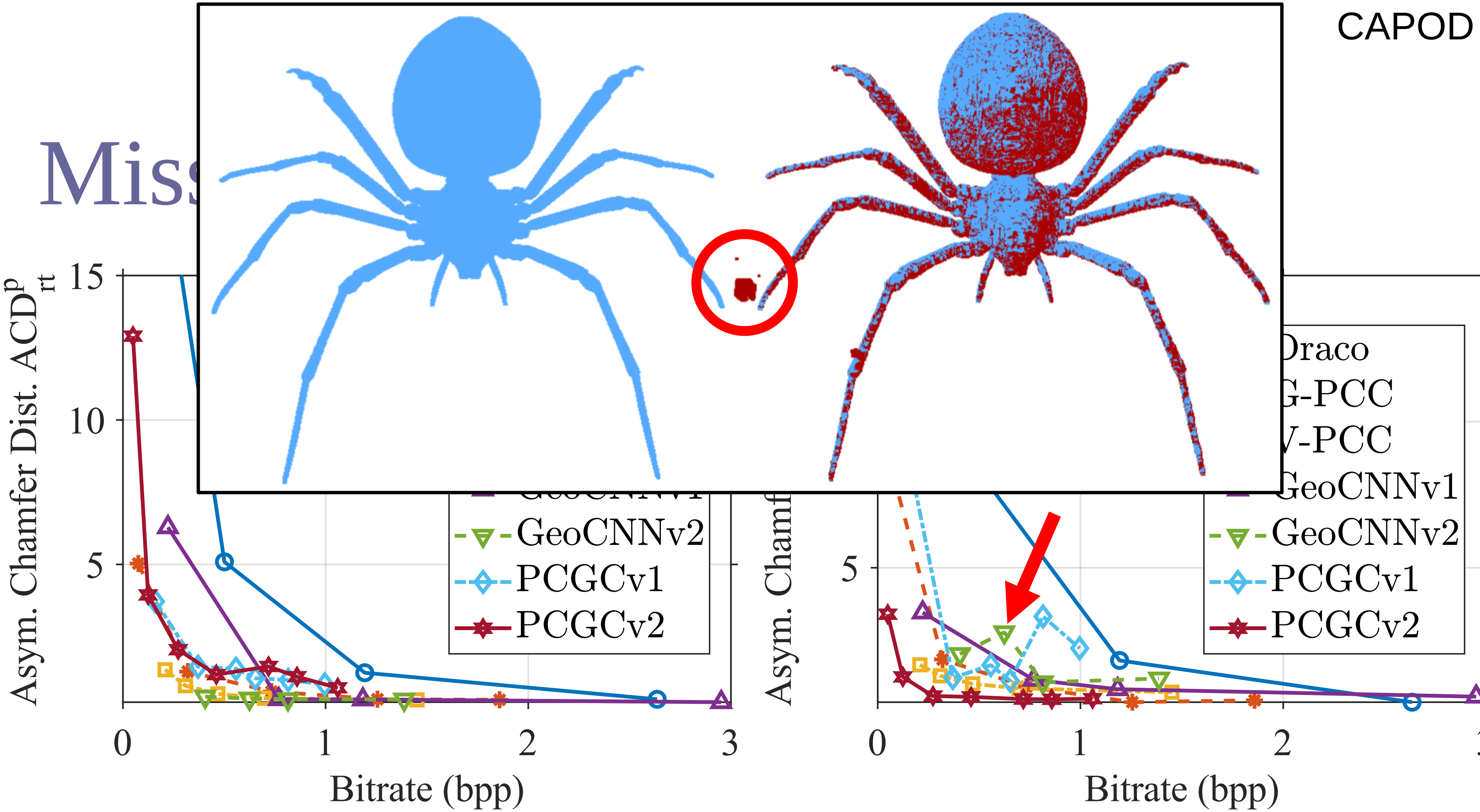
Point-to-Plane (p2pl) Is Better



- Overall, point-to-plane metrics have similar trend with point-to-point ones
- Point-to-plane metrics are more related to the visual quality [1]

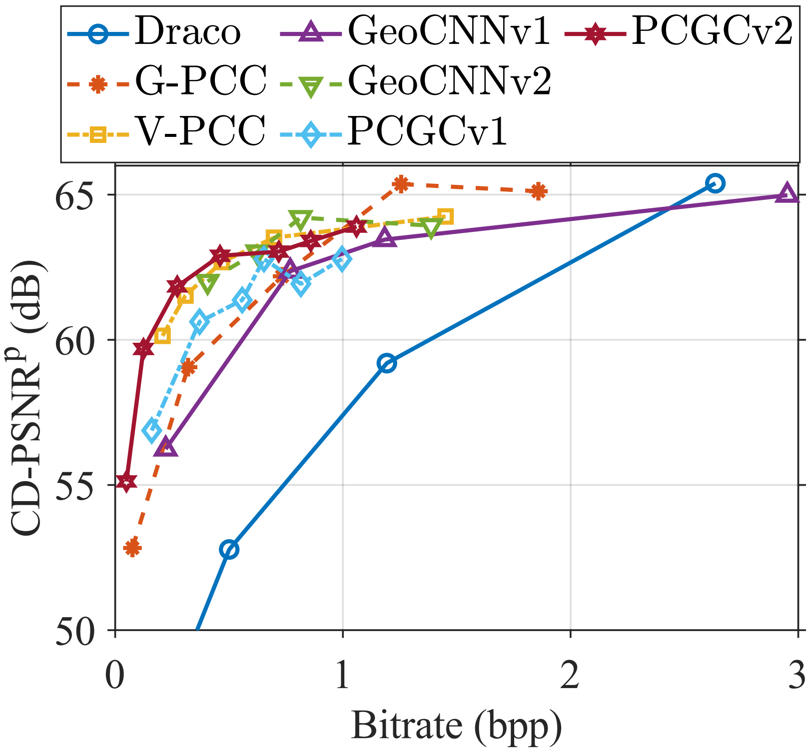
[1] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro, "Geometric distortion metrics for point cloud compression," in 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017, pp. 3460–3464.

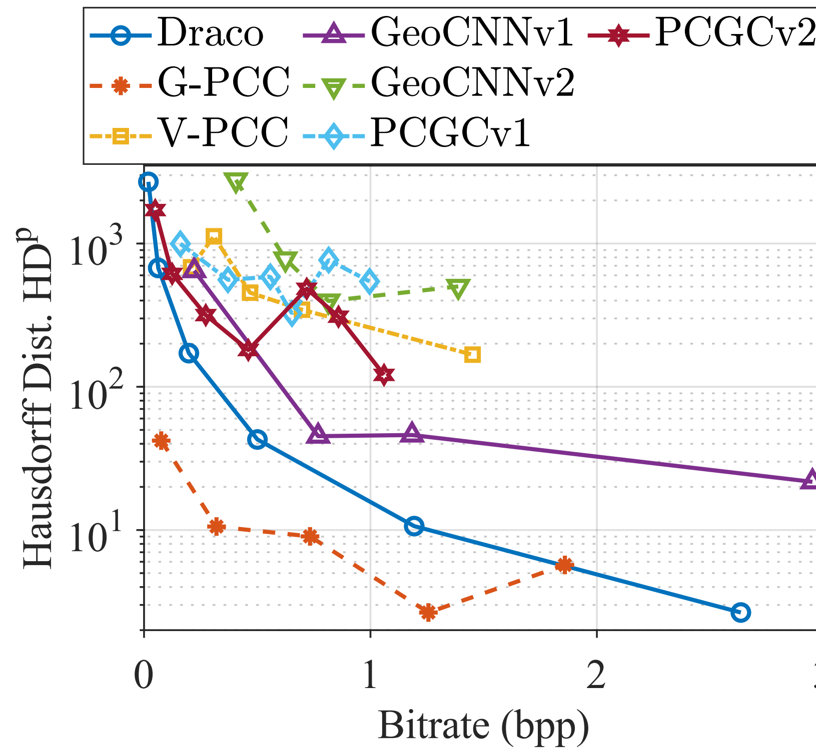
Miss



- High ACD^{p}_{rt} value indicates missing points in the reconstructed point cloud
- High ACD^{p}_{tr} value indicates extra points in the reconstructed point cloud

NN-based PCC Algorithms Perform Well But Not Stable



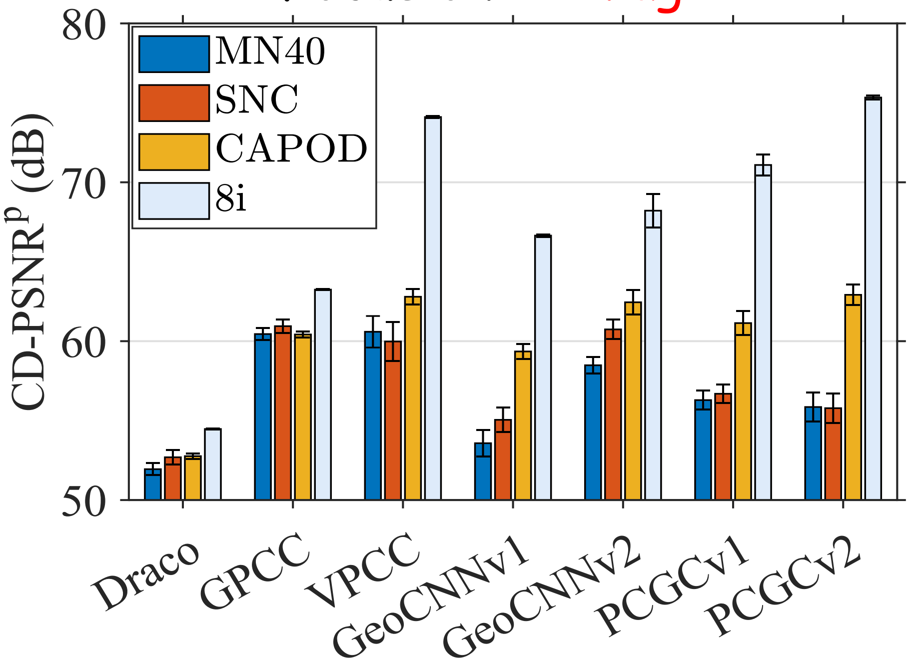


- GeoCNNv2 and PCGCv2 have the leading position, but face severe outlier problem
- G-PCC performs the best over 1 bpp and has stable results on the reconstructed point cloud

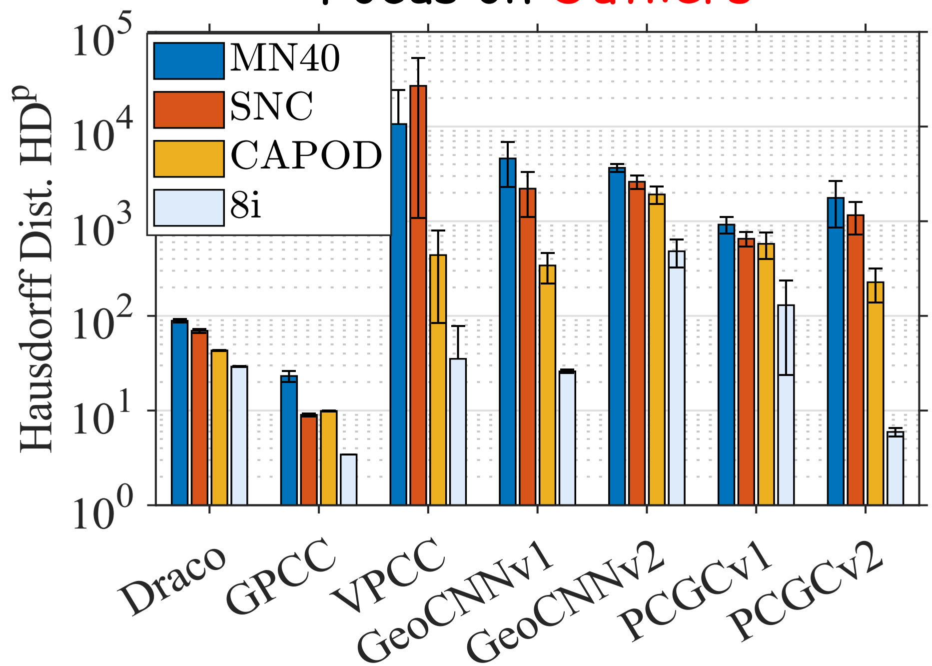
Avatars Are Easier to Compress than Objects?

0.5 bpp

Focus on Average

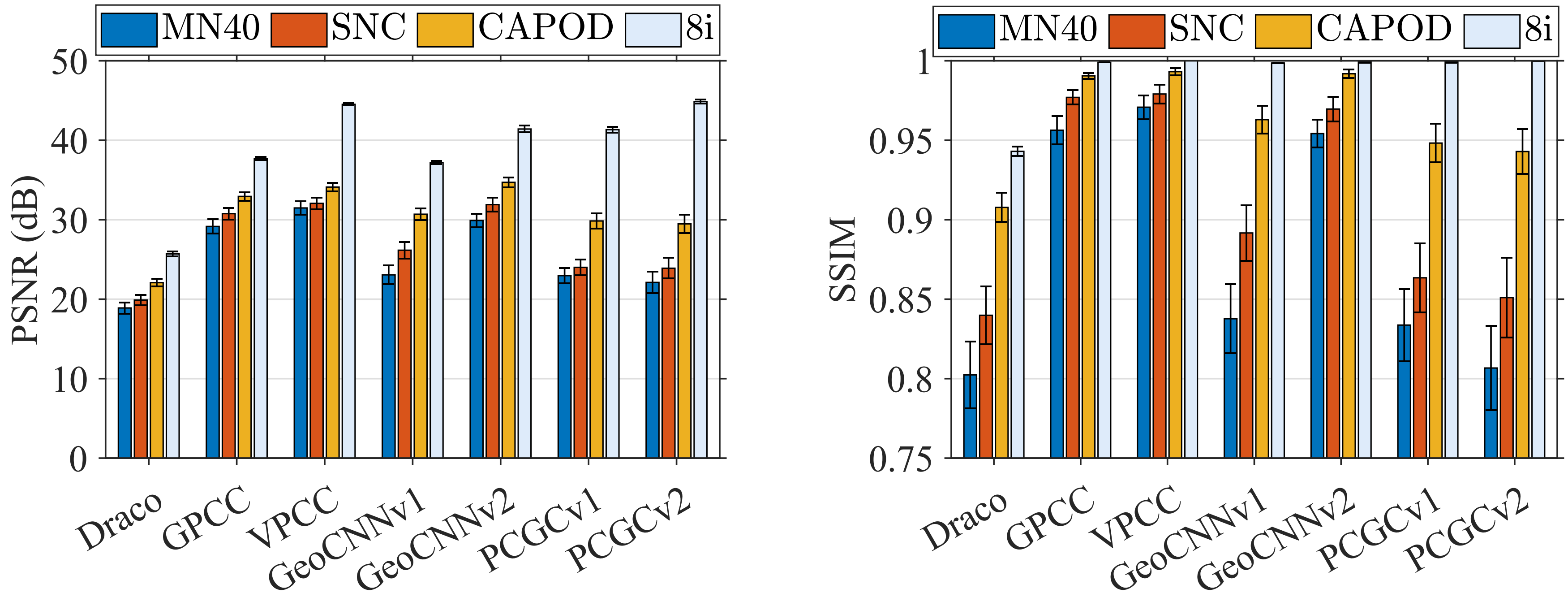


Focus on Outliers



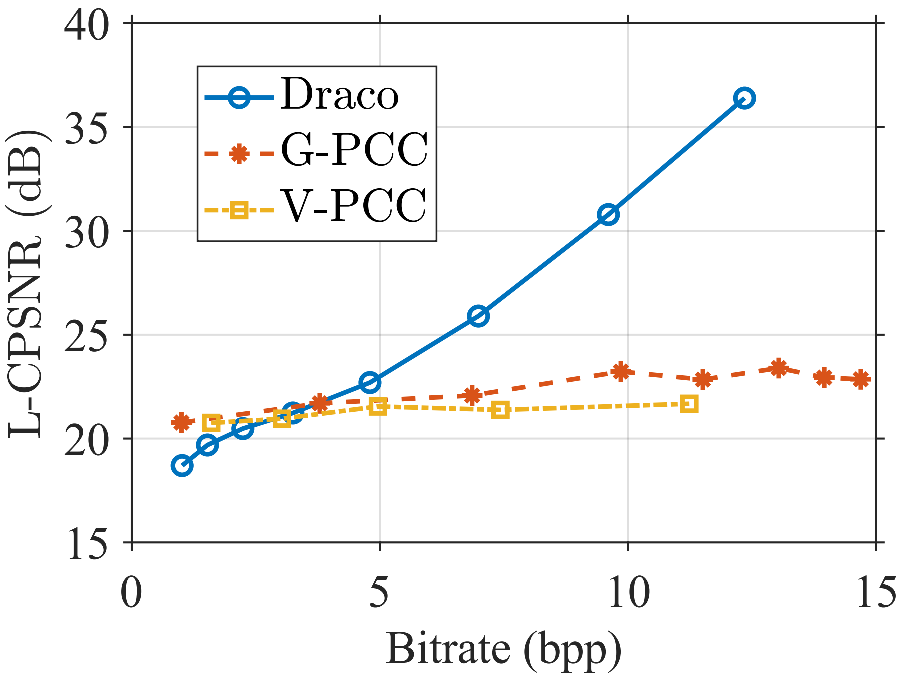
- All NN-based PCC algorithms and V-PCC have much better quality and higher stability on 8i (avatars) than other datasets (objects)
- All NN-based PCC algorithms are trained with object datasets

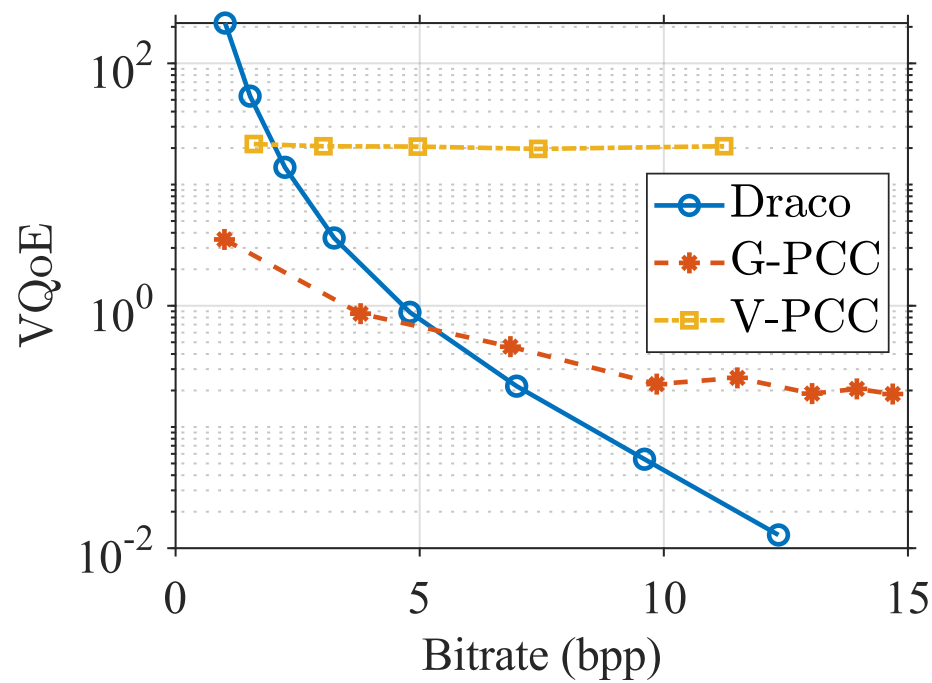
How About 2D Visual Quality?



- SP-based PCC algorithms achieve more robust performance across different datasets than NN-based ones
- NN-based PCC algorithms may not be general enough to handle arbitrary object classes

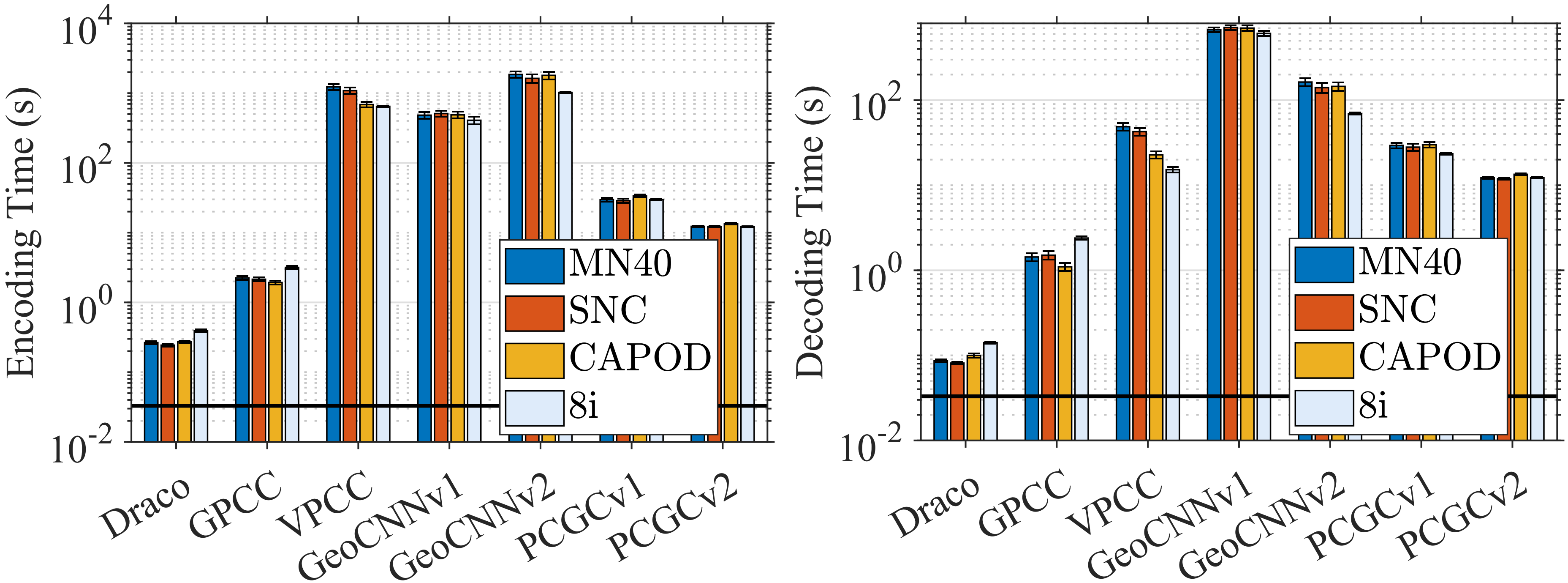
Coding Efficiency with Colors





- Draco preserve more color information at higher bitrate
- Draco has better control on trading off the quality and bitrate

Real-time Encoding/Decoding?



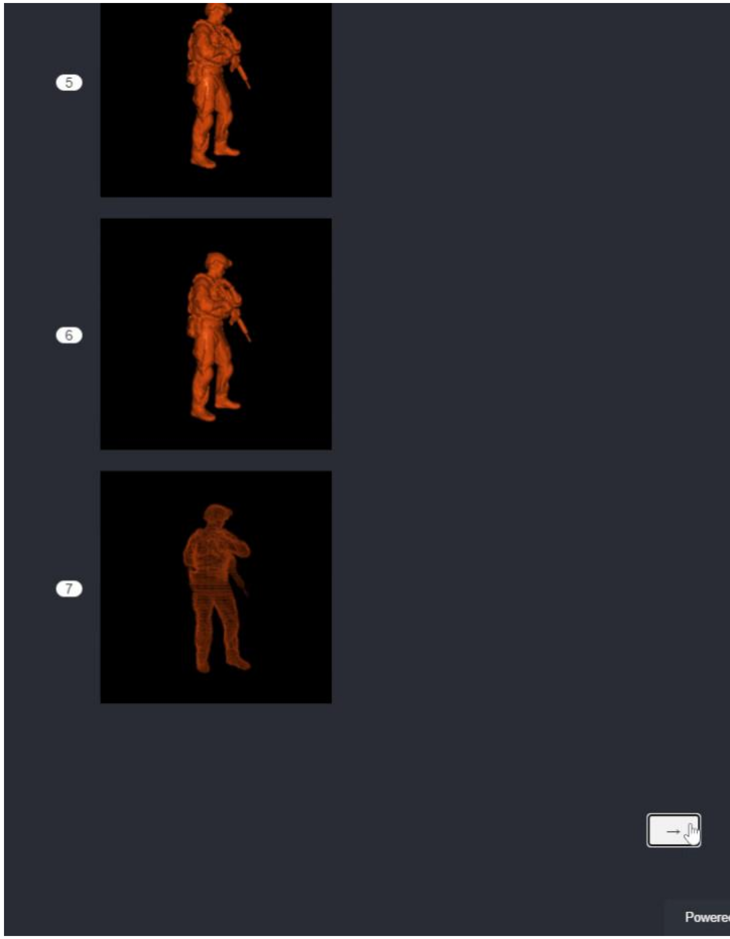
- Draco has the lowest running time, but none of the PCC algorithms encode/decode in real-time
- The more recent proposed NN-based PCC algorithm has lower running time

SUBJECTIVE RESULTS

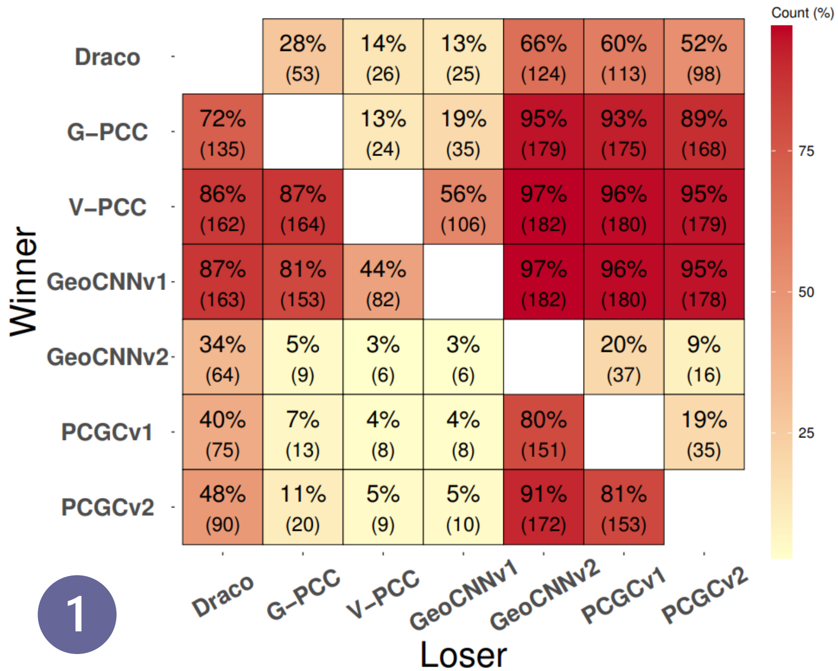
User Study Setup

- Web-based questionnaire consists of 2 parts
  - Perceived image quality
  - Perceived point cloud similarity
- Each part consists of 4 types of point cloud
  - Coordinate-only objects (chair)
  - Colored objects (chair)
  - Coordinate-only avatars
  - Colored avatars
- We recruit 47 subjects in total

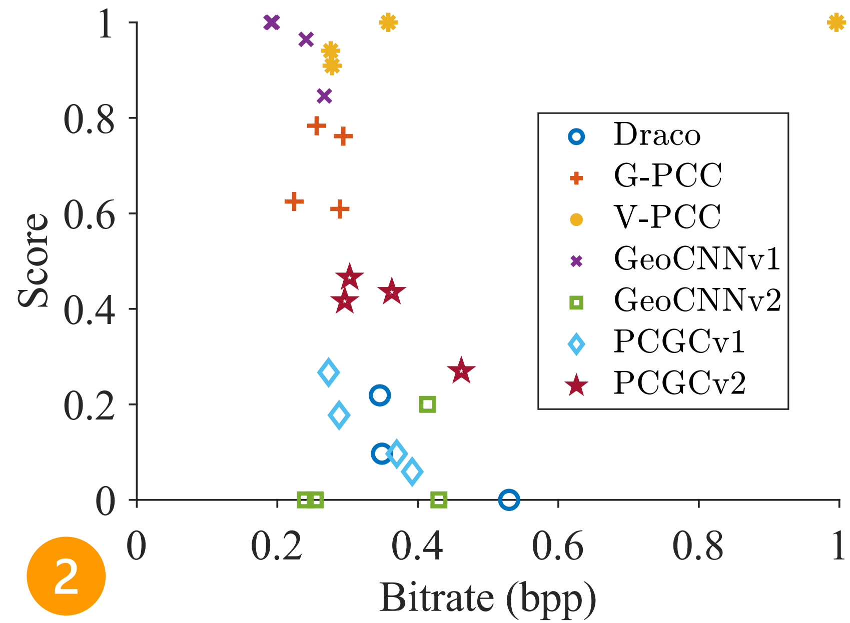
Subjects are asked to rank the GIF images from the best to the worse



Subjects prefer V-PCC and GeoCNNv1 in Image Quality on Coordinate-only Objects



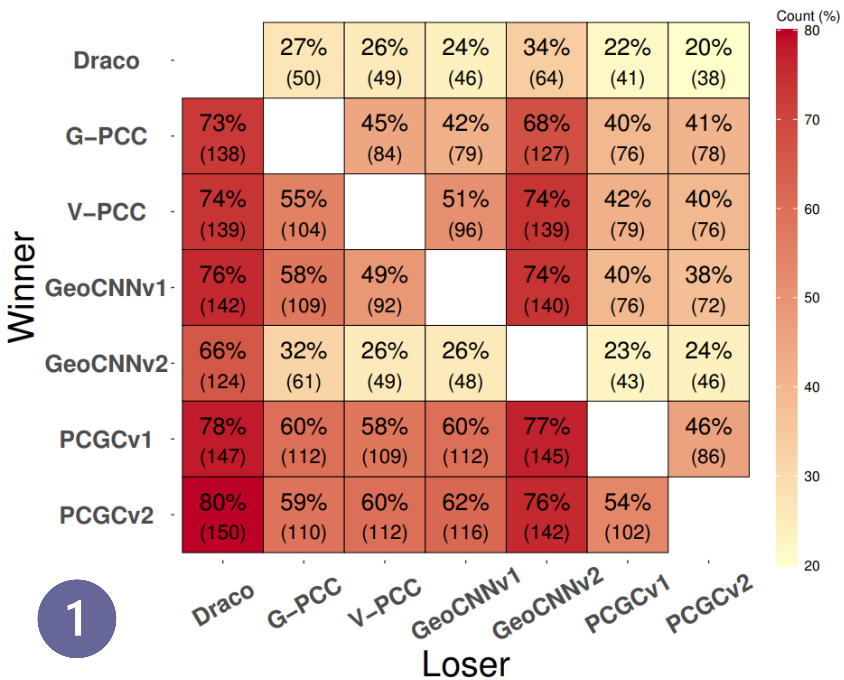
1



2

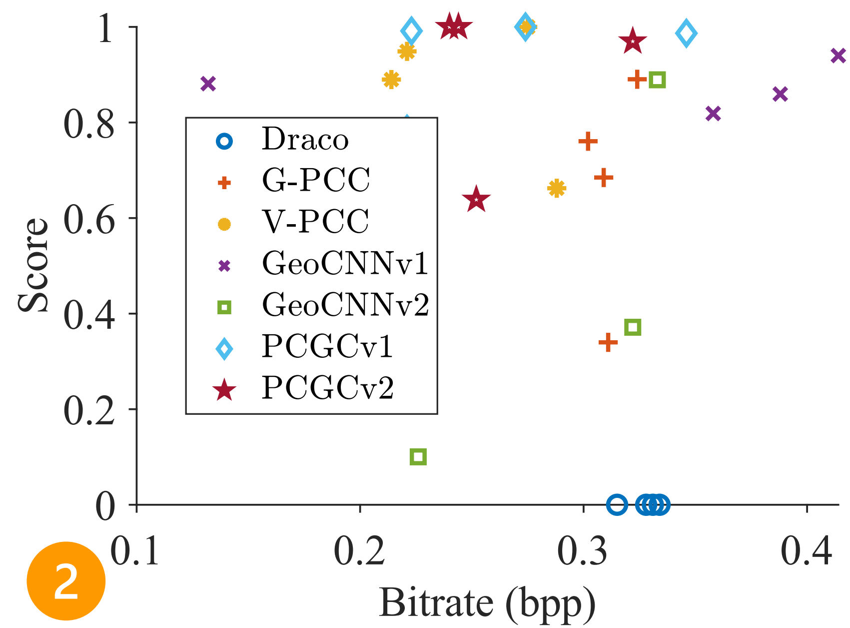
- Rank → Pairwise comparison matrix 1
- Plackett-Luce model → normalized model coefficients 2
- V-PCC and GeoCNNv1 take the lead, while GeoCNNv2 performs the worst
- GeoCNNv2 suffers from non-trivial artifacts

It Is Hard to Tell The Difference Among PCC Algorithms on Coordinate-only Avatars



1

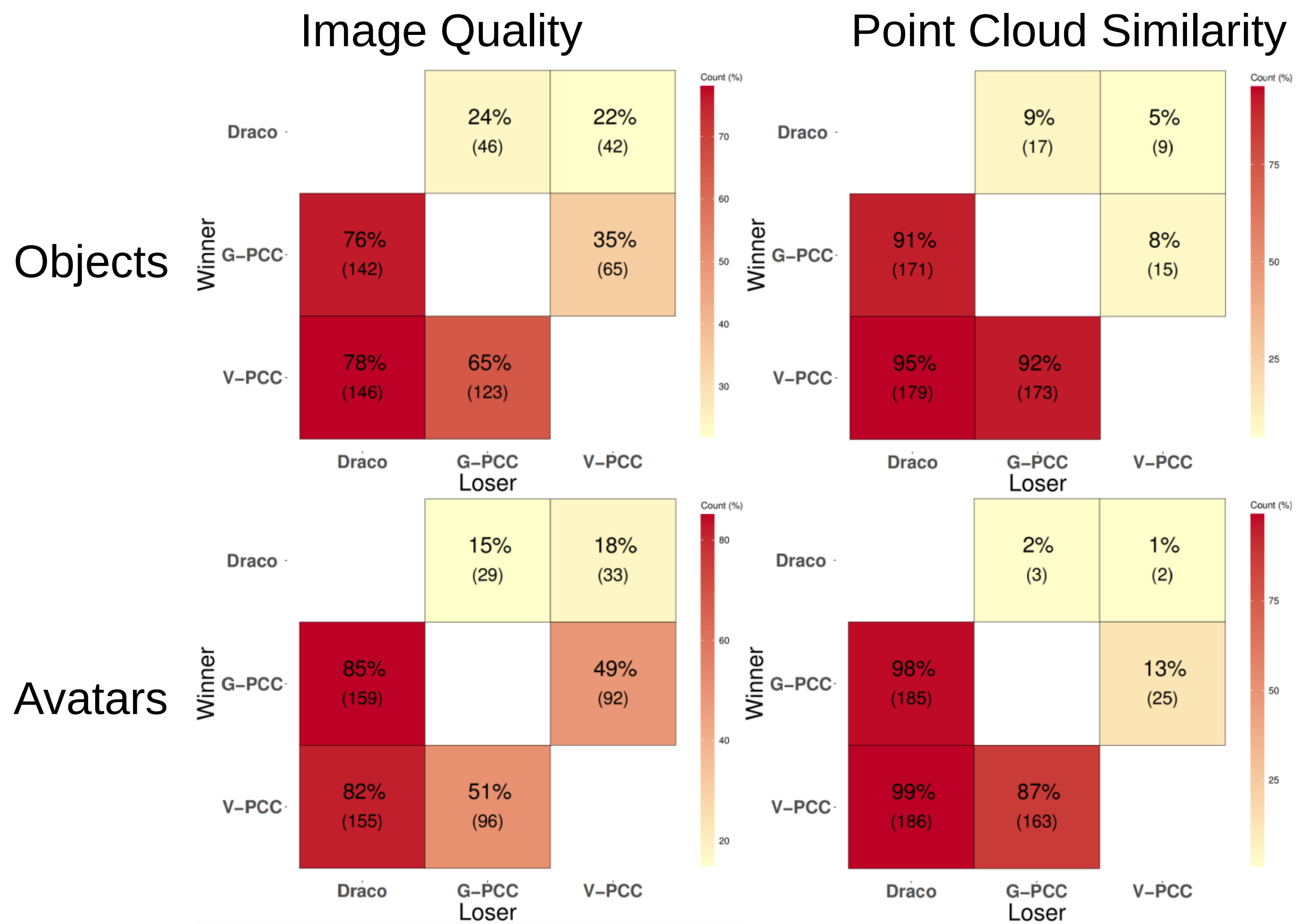
2



- Winning percentages are very close to 50% in most cases 1
- GeoCNNv2 delivers much better subjective image quality than Draco, which is opposite on objects 2

0.5 bpp

Very Similar Trend Are Found in Colored Objects And Avatars



- V-PCC performs the best, followed by G-PCC
- Draco suffers from the duplicated points[1]

[1]Draco is specifically designed to avoid merging duplicated points, see https://github.com/google/draco/issues/591#issuecomment-703820616

No Significant Correlation with Objective Metrics

			Type	Y-PSNR (dB)	SSIM	ACD^{p}_{rt}	ACD^{p}_{tr}	CD^{p}	CD-PSNR^{p} (dB)	HD^{p}	L-CPSNR (dB)	VQoE
Correlation Coefficient	Qua.	Coord.	Chair	0.19	**0.21**	0.07	-0.16	-0.02	0.03	-0.13	-	-
			Avatar	0.76	0.76	**-0.80**	-0.78	-0.79	0.31	-0.32	-	-
			All	**0.54**	0.49	-0.21	-0.49	-0.38	0.22	-0.25	-	-
		Color	Chair	0.78	0.81	-0.82	-0.80	-0.82	**0.83**	-0.76	0.79	-0.82
			Avatar	0.84	0.83	-0.91	-0.91	-0.91	0.89	-0.91	**0.95**	-0.91
			All	0.75	0.77	**-0.86**	-0.85	**-0.86**	0.79	-0.83	0.58	**-0.86**
	Sim.	Coord.	Chair	0.21	**0.23**	0.04	-0.12	-0.02	0.00	-0.20	-	-
			Avatar	0.81	0.79	**-0.83**	-0.82	**-0.83**	0.36	-0.34	-	-
			All	**0.57**	0.51	-0.23	-0.48	-0.39	0.23	-0.30	-	-
		Color	Chair	**0.79**	0.66	-0.69	-0.71	-0.71	0.75	-0.66	0.72	-0.71
			Avatar	0.78	0.76	-0.93	-0.93	-0.93	0.84	-0.93	**0.95**	-0.93
			All	0.72	0.64	-0.81	**-0.82**	**-0.82**	0.72	-0.80	0.57	**-0.82**
p-value	Qua.	Coord.	Chair	0.35	**0.29**	0.72	0.41	0.93	0.88	0.51	-	-
			Avatar	2.9×10^{-6}	3×10^{-6}	**3.1×10^{-7}**	8×10^{-7}	5×10^{-7}	0.11	0.09	-	-
			All	**1.9×10^{-5}**	1.4×10^{-4}	0.12	1.1×10^{-4}	4.2×10^{-3}	0.10	0.06	-	-
		Color	Chair	2.6×10^{-3}	1.4×10^{-3}	1.1×10^{-3}	1.7×10^{-3}	1.2×10^{-3}	**7.9×10^{-4}**	4×10^{-3}	2.2×10^{-3}	1.2×10^{-3}
			Avatar	5.4×10^{-4}	8.6×10^{-4}	4.3×10^{-5}	3.2×10^{-5}	3.1×10^{-5}	9.7×10^{-5}	3.6×10^{-5}	**3.4×10^{-6}**	3.1×10^{-5}
			All	2.4×10^{-5}	1.1×10^{-5}	**6.8×10^{-8}**	1.3×10^{-7}	7.1×10^{-8}	3.9×10^{-6}	4.4×10^{-7}	2.8×10^{-3}	7.1×10^{-8}
	Sim.	Coord.	Chair	0.27	**0.24**	0.83	0.54	0.92	0.99	0.31	-	-
			Avatar	1.6×10^{-7}	5.6×10^{-7}	**4.7×10^{-8}**	8.6×10^{-8}	5.9×10^{-8}	0.06	0.08	-	-
			All	**3.8×10^{-6}**	5.8×10^{-5}	0.09	1.7×10^{-4}	3.4×10^{-3}	0.09	0.03	-	-
		Color	Chair	**2×10^{-3}**	0.02	0.01	0.01	0.01	4.8×10^{-3}	0.02	0.01	0.01
			Avatar	2.7×10^{-3}	4.1×10^{-3}	1.4×10^{-5}	8.8×10^{-6}	8.8×10^{-6}	6.1×10^{-4}	9.6×10^{-6}	**2.6×10^{-6}**	8.8×10^{-6}
			All	7.5×10^{-5}	7.1×10^{-4}	1.9×10^{-6}	7.8×10^{-7}	**7.4×10^{-7}**	8.4×10^{-5}	3.1×10^{-6}	3.6×10^{-3}	**7.4×10^{-7}**

- **Bold font** indicates the highest value among all the considered objective metrics in each row.

- Avatar → some objective metrics have significant correlations
- Objects → no significant correlation
- None of objective metric can predict the quality well

FUTURE OF NN-BASED PCC ALGORITHMS

Potential Advantages and Disadvantages of the NN-based PCC Algorithms

- Not data-dependent
- Perform very well on 8i datasets (avatars)
  - Good news for 3D immersive teleconferencing
- Not stable, generate outlier points (blocks) in some cases
- The latest one (PCGCv2) has a much lower running time
  - Still slower than SP-based ones
- Few papers work on compressing attributes like colors
  - Worth further research

CONCLUSION

Conclusion

- Propose an open-source, modularized benchmark platform, PCC Arena
- Conduct an extensive comparison of seven PCC algorithms along with a wide spectrum of datasets and performance metrics
- Conduct a user study and analyze the correlations between subjective scores and objective metrics
- Discuss on some great potentials of NN-based PCC algorithms

Future Directions

- Offer the options for users to manipulate the input point cloud datasets
  - automatically alignment, rotation, scaling, etc.
- Consider application-wise performance metrics, even develop one for certain usage scenario
  - The performance metrics are independent of the usage scenarios

Publications and Cooperators

- **C. Wu**, C. Hsu, T. Kuo, C. Griwodz, M. Riegler, G. Morin, and C. Hsu, “PCC Arena: A benchmark platform for point cloud compression algorithms,” *ACM International Workshop on Immersive Mixed and Virtual Environment Systems (MMVE’20)*, pages 1–6, June 2020.
- **C. Wu**, X. Li, R. Rajesh, W. Ooi, and C. Hsu, “Dynamic 3D point cloud streaming: distortion and concealment,” *ACM Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV’21)*, pages 98–105, September 2021.
- **C. Wu**, C. Hsu, T. Hung, C. Griwodz, W. Ooi, and C. Hsu, “Quantitative comparison of point cloud compression algorithms with PCC Arena,” *IEEE Transactions on Multimedia*, July 2021, **Under Review**.

- Carsten Griwodz, *University of Oslo*
- Wei Tsang Ooi, *National University of Singapore*
- Chih-Fan Hsu, *National Yang Ming Chiao Tong University*
- Géraldine Morin, *Université de Toulouse — IRIT*
- Rahul Rajesh, *National University of Singapore*
- Michael Riegler, *Simula Research Lab, Norway*
- Tzu-Kuan Hung, *National Tsing Hua University*
- Ting-Chun Kuo, *National Tsing Hua University*
- Xiner Li, *Tsinghua University*

Thank you for listening

Q&A

BACKUP SLIDES

PCC Arena

- Algorithm Wrapper
  - Define a new class for each PCC algorithm inherited from the base class
  - Implement the virtual method in the base class, that are `encode()` and `decode()`
  - Base class provides public methods either for running over a dataset or running on a single point cloud
- Evaluator
  - class ViewIndependentMetric()
    - Wrap the metric software and parse the results
- Config Files
  - Set up all the config parameters with YAML files

Software for Quality Metrics

- Modified based on mpeg-pcc-dmetric
- Implement a QoE metric of combining coordinates and color from Prof. Pablo’s paper
- Bypass the built-in on the fly resolution calculation due to the unexpected behavior of it
  - Calculate the resolution with an open-source project, `gdiam-1.0.3`
  - resolution: Maximum distance of a pair of points among a point cloud

Modifications on Sample PCC Algorithms

- PCGCv1
  - Improve file I/O in testing phase
  - Change .ply loader for generality
- PCGCv2
  - Extract encoding and decoding part from the whole experiment evaluation script