

# 000

# 001 Energy-Based Residual Latent Transport for

# 002 Unsupervised Point Cloud Completion

# 003

# 004 (Supplementary Material)

# 005

006  
007 BMVC 2022 Submission # 48  
008  
009  
010  
011  
012

## 1 Discussion on Impractical Latent-Space EBM Learning

014 In this section, we discuss why a direct end-to-end training method for latent-space energy-based model (EBM) training is impractical.

016 To perform gradient descent training, we need gradients for each learnable parameter.  
017 Suppose we deploy an EBM ( $E_\theta$ ) in the latent-space of an encoder-decoder architecture.  
018 We denote the encoder and decoder as  $\mathcal{E}_\alpha$  and  $\mathcal{D}_\beta$ , respectively, where  $\alpha$  and  $\beta$  are the  
019 corresponding learnable parameters. Recall that we use Langevin dynamics [1] to sample  
020 from the EBM, which is formulated:

$$021 \quad z^{(t+1)} = z^{(t)} - \frac{\delta^2}{2} \frac{\partial}{\partial z} E_\theta(z^{(t-1)}) + \delta \varepsilon^{(t)}, \quad (1)$$

024 where  $t$  represents Langevin steps,  $\delta$  is the step size,  $\varepsilon^t \sim \mathcal{N}(0, \mathbf{I})$  is Gaussian random noise,  
025 and  $z^{(t)} \in \mathbb{R}^n$ . Subsequently, we can formulate the forward pass as:

$$026 \quad y = \mathcal{D}_\beta \circ S_\theta^{(T)} \circ S_\theta^{(T-1)} \circ \dots \circ S_\theta^{(1)} \circ \mathcal{E}_\alpha(x), \quad (2)$$

027 where  $S_\theta^{(t)}$  denotes the  $t$ -th Langevin dynamics sampling iteration as shown by Eq. 1 and  
028 there are  $T$  steps in total. Suppose the loss function is  $\mathcal{L}(y)$ , the gradient for the loss function  
029 w.r.t  $\beta$  can be derived as  $\frac{\partial \mathcal{L}(y)}{\partial \beta}$ . However, the gradient for updating  $\alpha$  involves unfolding the  
030 Langevin dynamics as shown by the equation below:

$$032 \quad \frac{\partial \mathcal{L}(y)}{\partial \alpha} = \frac{\partial \mathcal{L}(y)}{\partial z^{(T)}} \frac{\partial z^{(T)}}{\partial z^{(T-1)}} \dots \frac{\partial z^{(2)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial \alpha} \quad (3)$$

035 Eq. 3 can be expensive to evaluate since it contains a second-order derivative term. This is  
036 shown by deriving a general gradient form of a Langevin dynamics step in Eq. 4:

$$037 \quad \frac{\partial z^{(t)}}{\partial z^{(t-1)}} = \mathbf{I} - \frac{\delta^2}{2} \frac{\partial}{\partial z^2} E_\theta(z^{(t-1)}) \quad (4)$$

039 where  $\mathbf{I}$  is an identity matrix and  $\frac{\partial}{\partial z^2} E_\theta(z^{(t-1)})$  is the term that contains a second-order  
040 derivative. Since a second-order derivative requires quadratic time and space complexity  
041 for evaluation, it can be expensive or even infeasible to compute when the dimension of  $z^t$   
042 is large, e.g.  $n = 2^{10}$ . In conclusion, directly back-propagating gradients is impractical for  
043 large-scale problems.

### 3 Training Algorithm

We provide pseudo-code to illustrate the training algorithm:

---

**Algorithm 1:** Training Algorithm
 

---

**Input:** partial point cloud domain  $\mathcal{X}$ , complete point cloud domain  $\mathcal{Y}$

**while** not converged **do**

$x \sim \mathcal{X}$

$y \sim \mathcal{Y}$

▷ point cloud encoding

$z_x \leftarrow \mathcal{E}_\alpha(x)$

$z_y \leftarrow \mathcal{E}_\alpha(y)$

▷ latent transport

$z_{x \rightarrow y}^{(0)} \leftarrow z_x$

**for** sample step  $t = 1$  to  $T$  **do**

$\tilde{z}_{x \rightarrow y}^t \leftarrow \tilde{z}_{x \rightarrow y}^{t-1} - \frac{\delta^2}{2} \frac{\partial}{\partial z_{x \rightarrow y}} E_\theta(z_{x \rightarrow y}) + \delta \epsilon^t, \quad \epsilon^t \sim \mathcal{N}(0, \mathbf{I})$

**end**

$r_{xy} \leftarrow z_x - \tilde{z}_{x \rightarrow y}^t$

$\tilde{z}_{x \rightarrow y}^t \leftarrow z_x + \Omega(r_{xy})$

▷ latent variable decoding

$\tilde{x} \leftarrow \mathcal{D}_\beta(\tilde{z}_{x \rightarrow y}^t)$

$\tilde{y} \leftarrow \mathcal{D}_\beta(z_y)$

▷ update parameters

$\Delta\alpha \leftarrow \nabla_\alpha \mathcal{L}_{\text{recon}} + \lambda_1 \nabla_\alpha \mathcal{L}_{\text{fidelity}} - \lambda_2 \nabla_\alpha \mathcal{D}_\gamma(\tilde{x})$

$\Delta\beta \leftarrow \nabla_\beta \mathcal{L}_{\text{recon}} + \lambda_1 \nabla_\beta \mathcal{L}_{\text{fidelity}} - \lambda_2 \nabla_\beta \mathcal{D}_\gamma(\tilde{x})$

$\Delta\gamma \leftarrow \nabla_\gamma \mathcal{L}_{\text{adv}}$

$\Delta\theta \leftarrow \nabla_\theta \mathcal{L}_{\text{ebm}}$

update parameters based on  $\Delta\alpha, \Delta\beta, \Delta\gamma, \Delta\theta$  using Adam optimizer

**end**

---

### 4 Implementation of Residual Sampling

We provide PyTorch-style implementation of the proposed residual sampling method:

---

```
def residual_sampling(z0, energy_function, n_step, step_size):
    z = z0.clone().detach()
    z.requires_grad = True
    with torch.enable_grad():
        for _ in range(n_step):
            noise = step_size * torch.randn_like(z)
            grad = torch.autograd.grad(energy_function(z), z, only_inputs=True)[0]
            z = z - torch.square(step_size) * grad + noise
    z = z0 - z.detach()
    return z
```

---

## 092 5 Inference Cost

093  
 094 We compare inference cost between our method and other methods, *i.e.* Cycle4Completion [8],  
 095 and ShapeInversion [9]. As shown in Table. 1, ours is much more efficient than Cy-  
 096 cle4Completion and ShapeInversion while achieving the best performance in terms of average  
 097 Chamfer Distance (CD) on the 3D-EPN [8] dataset. Note that our models can perform  
 098 inference faster than Cycle4Completion by  $2.9 \times 10^{-3}$  per sample and it is over 2000 times  
 099 faster than ShapeInversion. On the other hand, although our model contains 13M more

100 Table 1: Computational cost on our method compared with existing methods.

Method	#params	Time (s)	Avg. CD ( $\times 10^4$ )
C4C	21M	$9.3 \times 10^{-3}$	14.3
ShapeInv.	41M	$1.5 \times 10^1$	23.6
Ours	34M	$6.4 \times 10^{-3}$	9.4

106 parameters than Cycle4Completion, our method has a smaller inference time cost. This is  
 107 because Cycle4Completion contains a time-consuming folding-based decoder [8] while we  
 108 use an efficient attention-based decoder with a shallow MLP for decoding latent codes.  
 109

## 110 6 Additional ShapeNet samples

111 We provide additional ShapeNet completion results in this section. Our model, as well as two  
 112 existing methods (Cycle4Completion [8] and ShapeInversion [9]), are trained on the 3D-EPN  
 113 dataset [8]. For each of the eight categories in 3D-EPN dataset, 3 samples are visualized.  
 114 As illustrated by Figure 1, Figure 2, and Figure 3, our method can produce high-fidelity  
 115 completion with more details preserved.  
 116

## 117 7 Real-World samples

118 We provide qualitative results of our method compared with Cycle4Complete [8] and ShapeIn-  
 119 version [9]. The real-world scans comes from two datasets, *i.e.* ScanNet [10] and Matter-  
 120 Port3D [11].  
 121

## 122 8 Additional Uncertainty Estimation Samples

123 This section provides additional uncertainty estimation results of our method as shown in  
 124 Figure 5 and Figure 6.  
 125

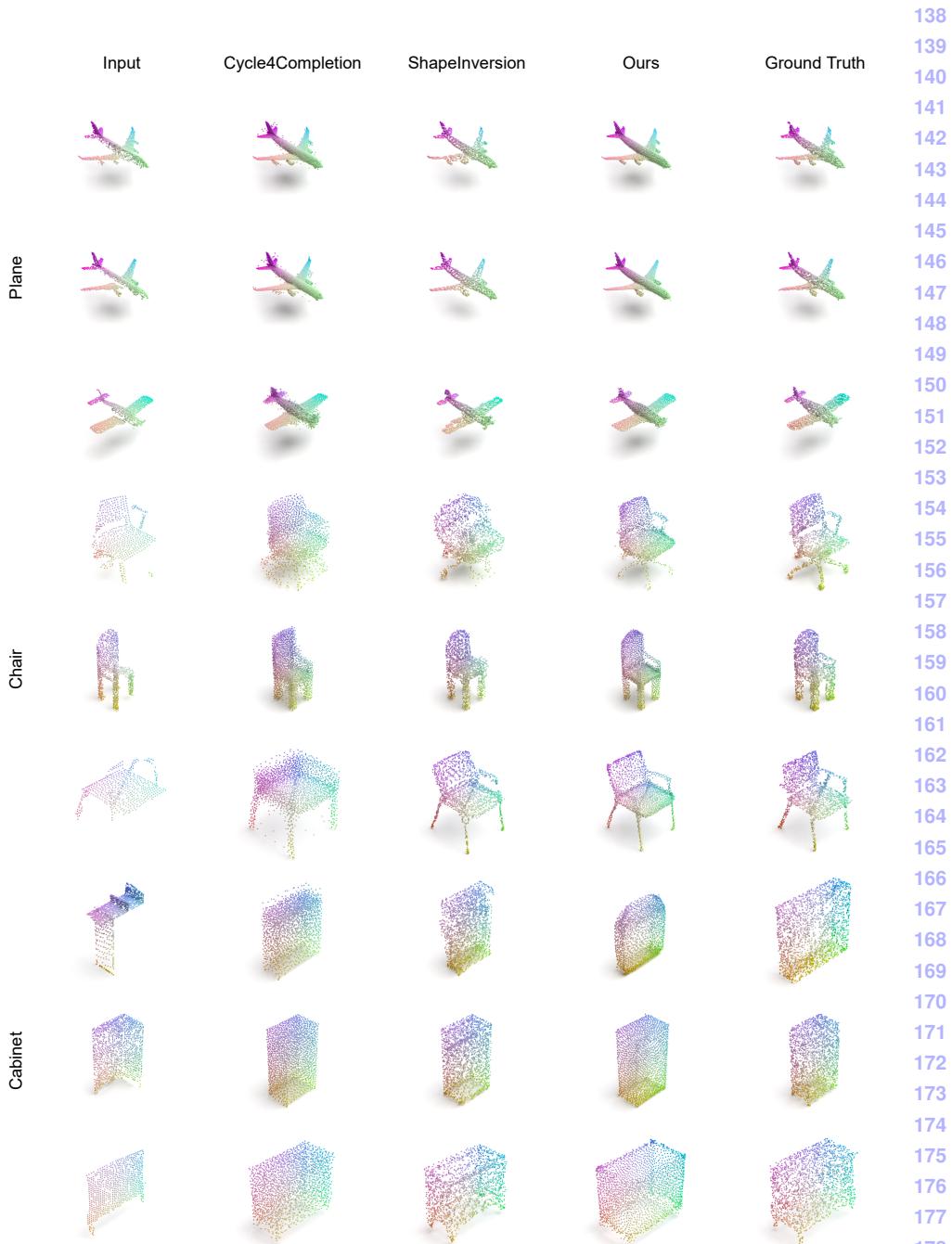


Figure 1: Additional Qualitative Result on the 3D-EPN dataset. From left to right by column: input incomplete point clouds, results from Cycle4Completion [6], ShapeInversion [7], ours, and ground truth.

138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183

184

185



186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

Figure 2: Additional Qualitative Results on the 3D-EPN dataset (continued 1). From left to right by column: input incomplete point clouds, results from Cycle4Completion, ShapeInversion, ours, and ground truth.

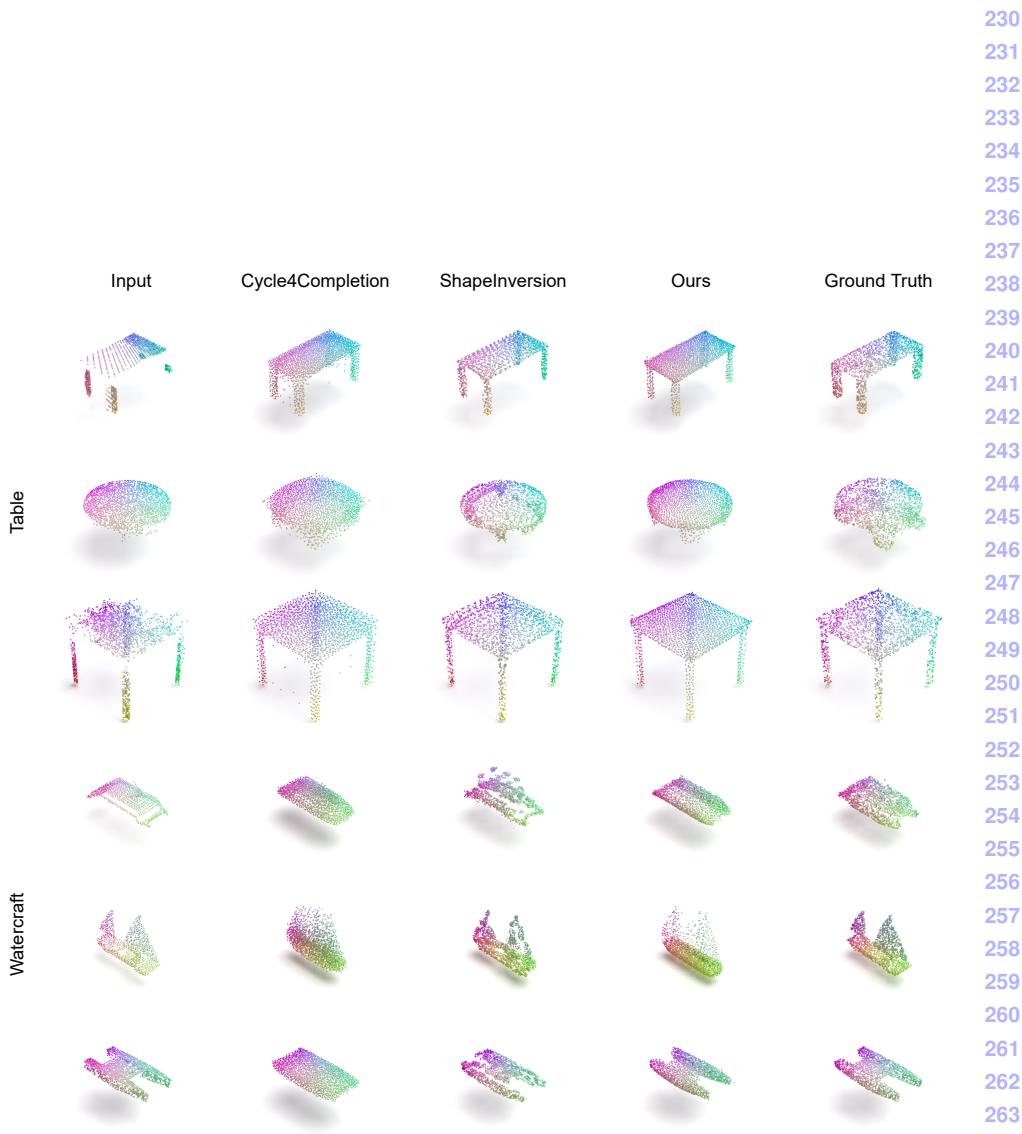


Figure 3: Additional Qualitative Results on the 3D-EPN dataset (continued 2). From left to right by column: input incomplete point clouds, results from Cycle4Completion, ShapeInversion, ours, and ground truth.

276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321

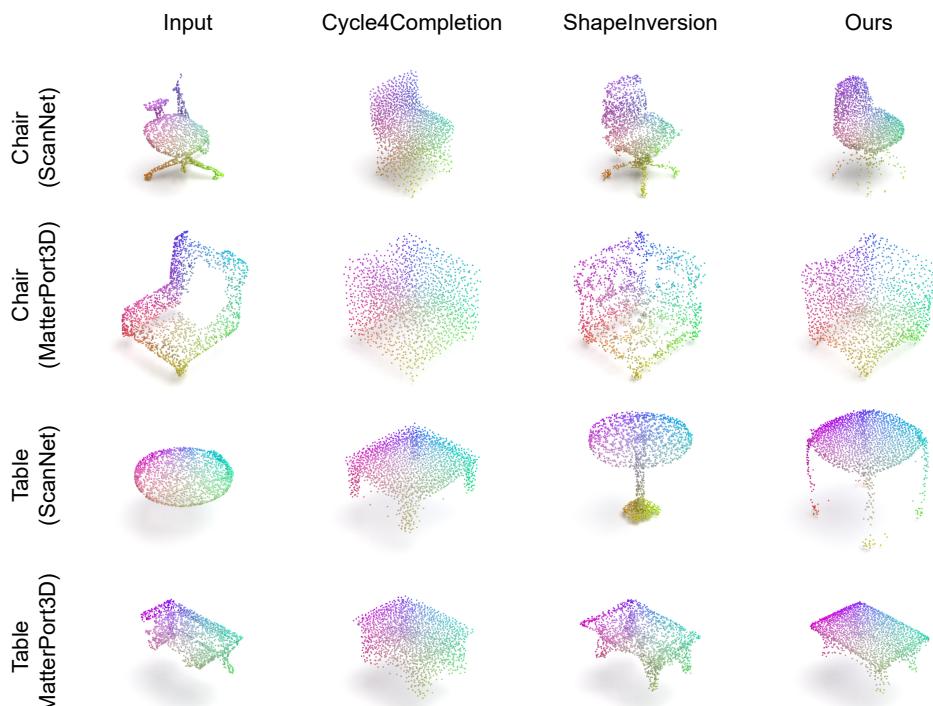


Figure 4: Qualitative Result on the real-world dataset. From left to right by column: incomplete point clouds, results from Cycle4Completion, ShapeInversion, and ours.

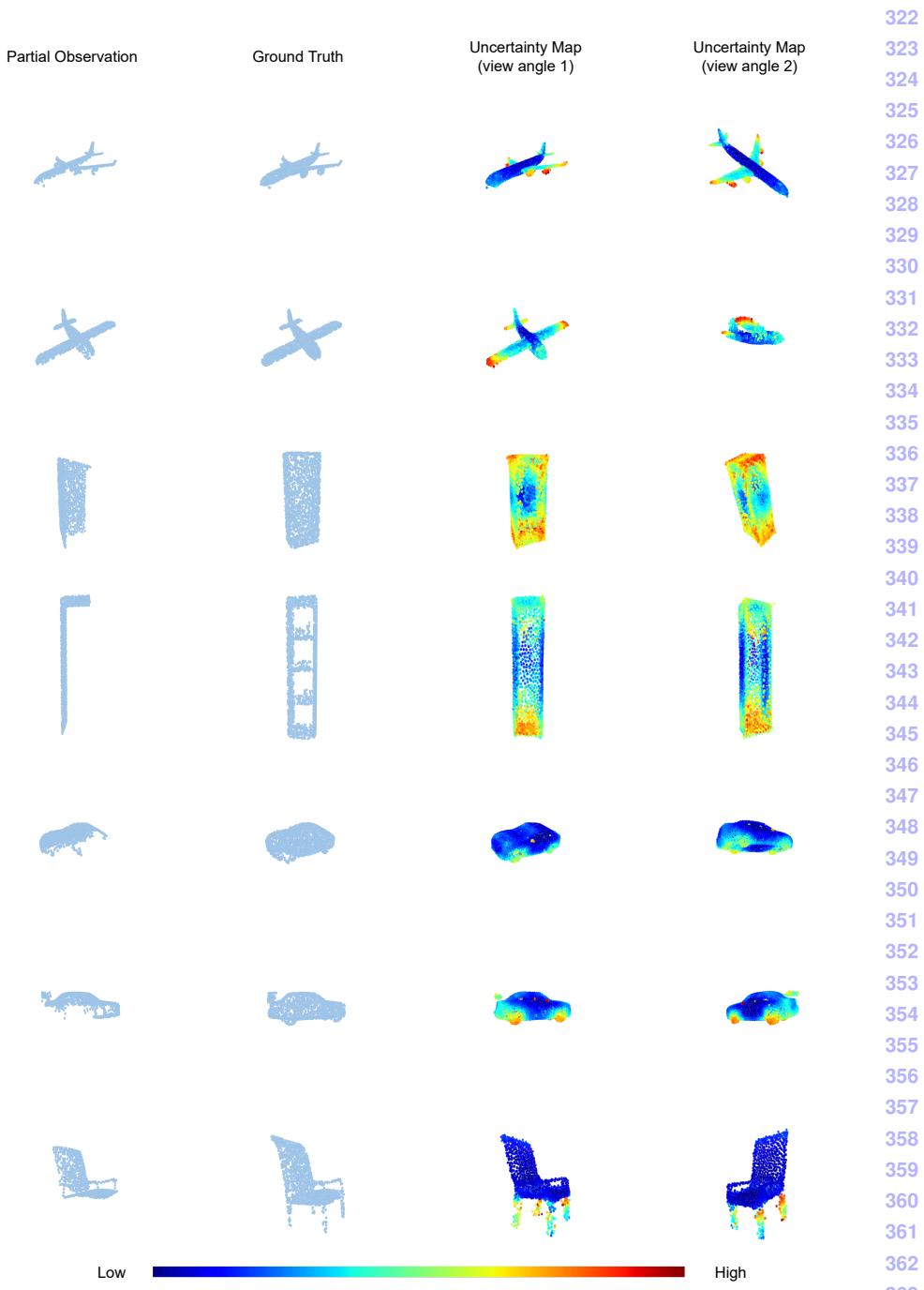


Figure 5: Additional uncertainty maps produced by our method. The input and ground truth are shown in the first and second rows followed by two viewing angles of their uncertainty map.

368

369

Partial Observation

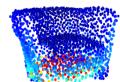


370

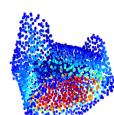
Ground Truth



371

Uncertainty Map  
(view angle 1)

372

Uncertainty Map  
(view angle 2)

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

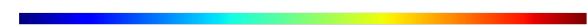
406

407

408

409

Low



High

Figure 6: Additional uncertainty maps produced by our method (continued). The input and ground truth are shown in the first and second rows followed by two viewing angles of their uncertainty map.

413

## References

- [1] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 414  
415  
416  
417  
418  
419
- [2] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 420  
421  
422  
423
- [3] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 424  
425  
426
- [4] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 427  
428  
429  
430  
431
- [5] Lyne P. Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 383–392, 2019. doi: 10.1109/CVPR.2019.00047. 432  
433  
434  
435
- [6] Xin Wen, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Cycle4completion: Unpaired point cloud completion using cycle transformation with missing region coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 436  
437  
438  
439  
440
- [7] Junzhe Zhang, Xinyi Chen, Zhongang Cai, Liang Pan, Haiyu Zhao, Shuai Yi, Chai Kiat Yeo, Bo Dai, and Chen Change Loy. Unsupervised 3d shape completion through gan inversion. In *CVPR*, 2021. 441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459