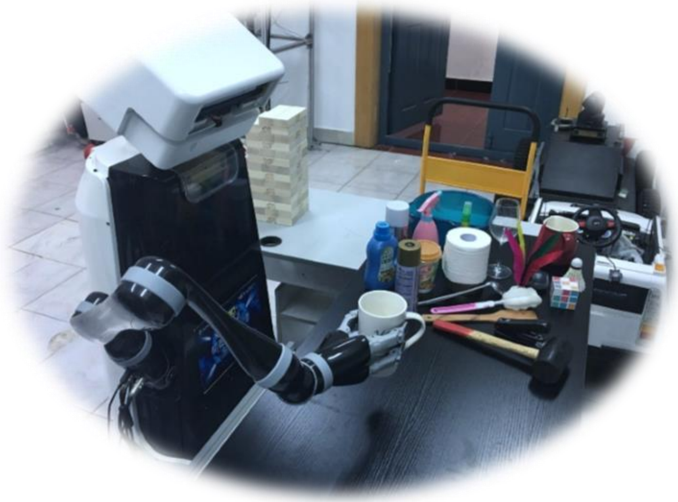
三维物体的表示与生成模型

Learning Representations and Generative Models for 3D Point Clouds

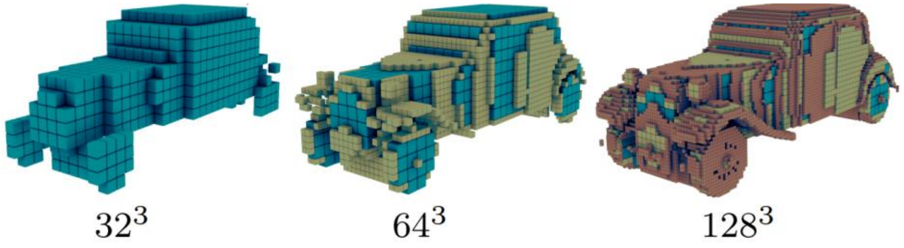


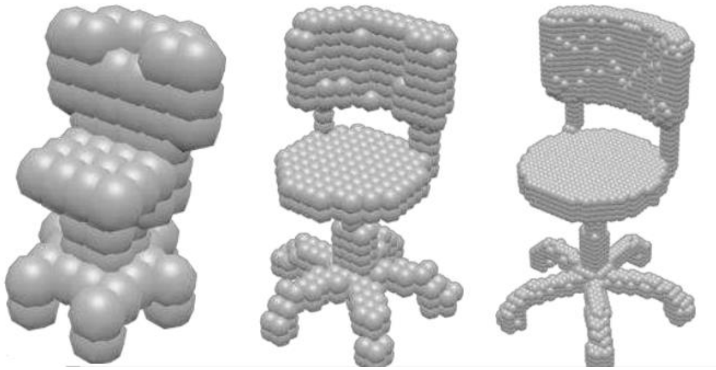
主　讲　人：潘 浩 洋

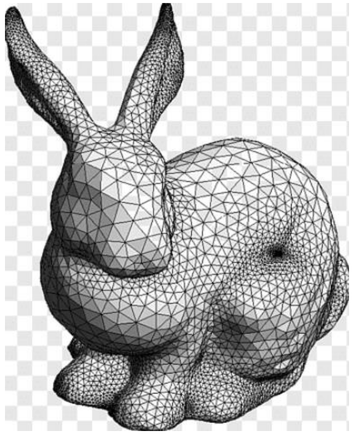
公　众　号：3D视觉工坊

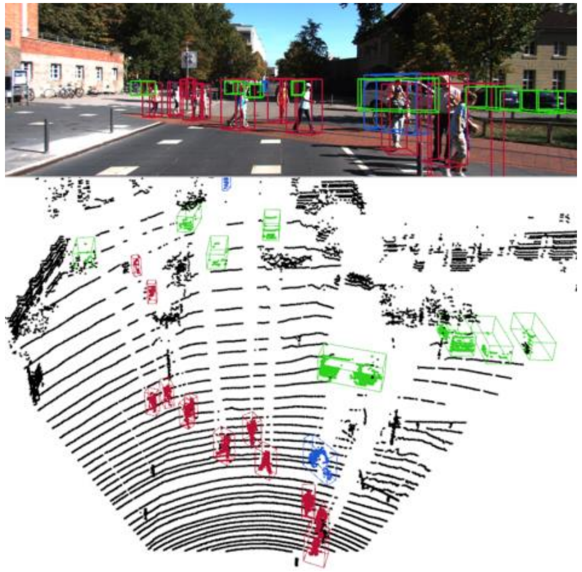
目 录

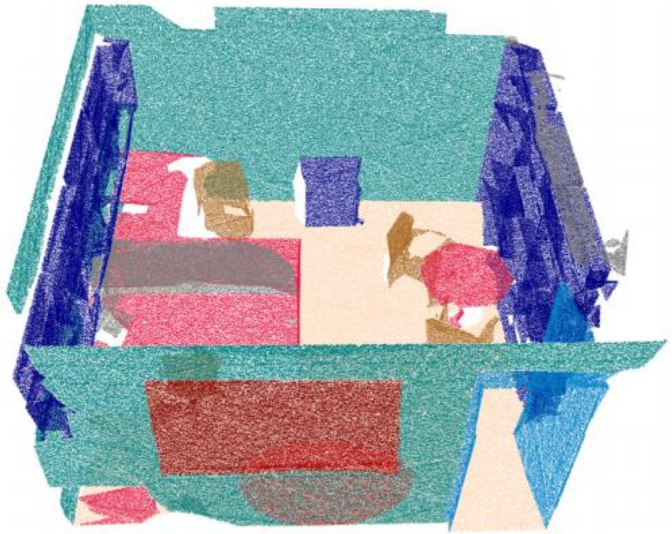
三维物体的表示







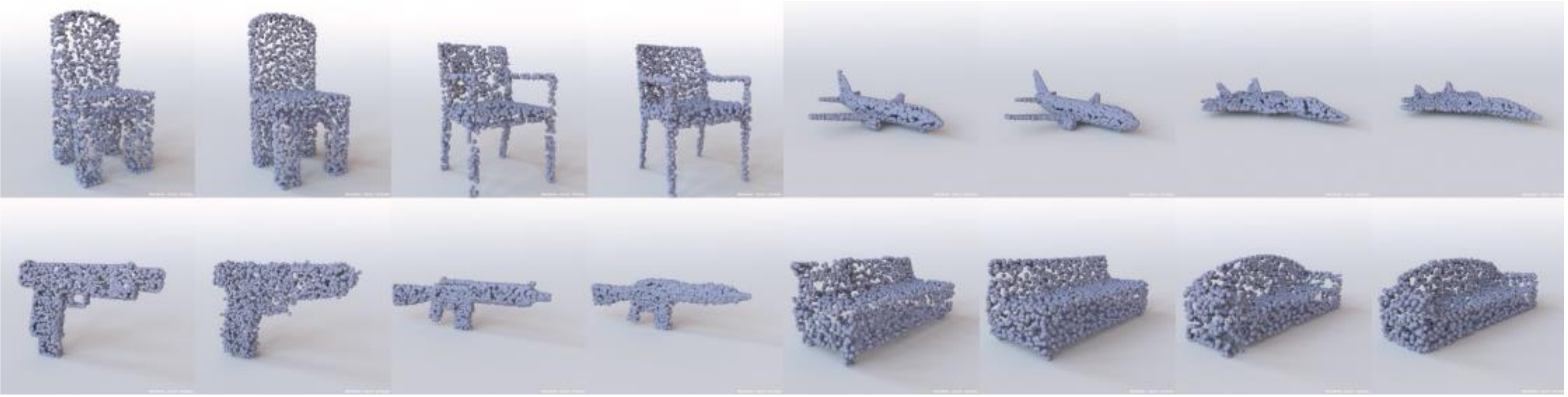




Learning Representations and Generative Models for 3D Point Clouds

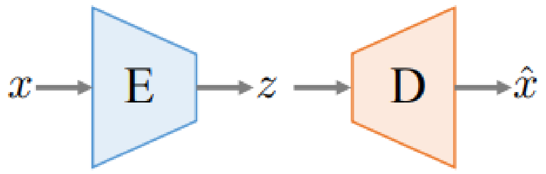
Panos Achlioptas [1] Olga Diamanti [1] Ioannis Mitliagkas [2] Leonidas Guibas [1]

[1]Department of Computer Science, Stanford University, USA
[2]MILA, Department of Computer Science and Operations Research, University of Montréal, Canada. Correspondence to: Panos Achlioptas <optas@cs.stanford.edu>.

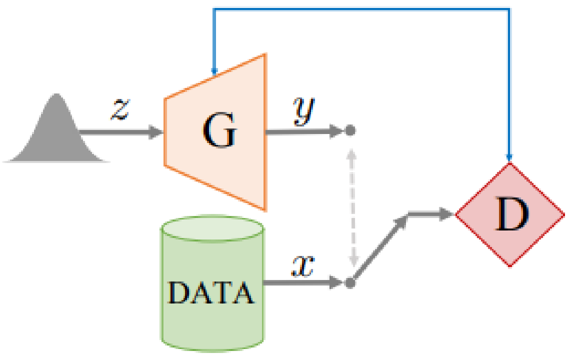


LatentGAN 模型

Autoencoders One of the main deep-learning components we use in this paper is the *AutoEncoder* (AE, inset), which is an architecture that learns to reproduce its input. AEs can be especially useful, when they contain a narrow *bottleneck layer* between input and output. Upon successful training, this layer provides a low-dimensional representation, or *code*, for each data point. The Encoder (E) learns to compress a data point $\boldsymbol{x}$ into its latent representation, $\boldsymbol{z}$. The Decoder (D) can then produce a reconstruction $\hat{\boldsymbol{x}}$, of $\boldsymbol{x}$, from its encoded version $\boldsymbol{z}$.

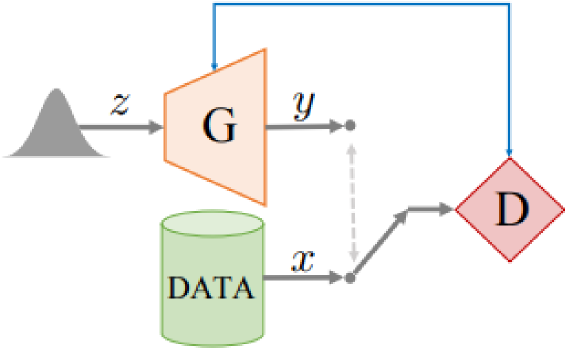


Generative Adversarial Networks In this paper we also work with Generative Adversarial Networks (GANs), which are state-of-the-art generative models. The basic architecture (inset) is based on a adversarial game between a *generator* (G) and a *discriminator* (D). The generator aims to synthesize samples that look indistinguishable from real data (drawn from $\boldsymbol{x} \sim p_{\text{data}}$) by passing a randomly drawn sample from a simple distribution $\boldsymbol{z} \sim p_z$ through the generator function. The discriminator is tasked with distinguishing synthesized from real samples.



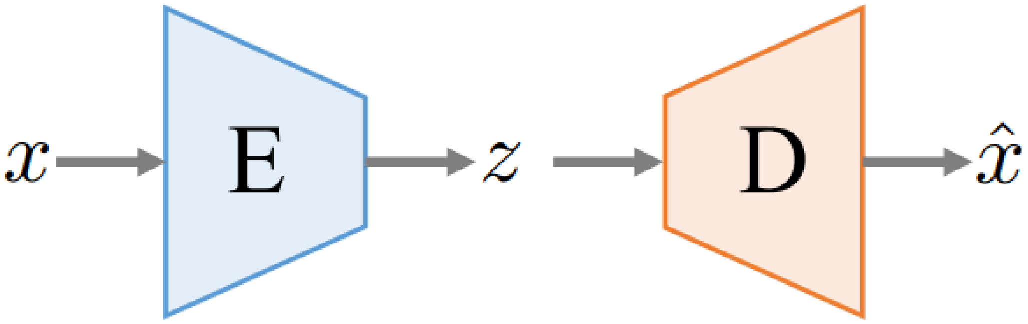
Gaussian Mixture Model A GMM is a probabilistic model for representing a population whose distribution is assumed to be multimodal Gaussian, i.e. comprising of multiple subpopulations, where each subpopulation follows a Gaussian distribution. Assuming the number of subpopulations is known, the GMM parameters (means and variances of the Gaussians) can be learned from random samples, using the Expectation-Maximization (EM) algorithm (Dempster et al., 1977). Once fitted, the GMM can be used to sample novel synthetic samples.

Generative Adversarial Networks In this paper we also work with Generative Adversarial Networks (GANs), which are state-of-the-art generative models. The basic architecture (inset) is based on a adversarial game between a *generator* (G) and a *discriminator* (D). The generator aims to synthesize samples that look indistinguishable from real data (drawn from $\boldsymbol{x} \sim p_{\text{data}}$) by passing a randomly drawn sample from a simple distribution $\boldsymbol{z} \sim p_z$ through the generator function. The discriminator is tasked with distinguishing synthesized from real samples.



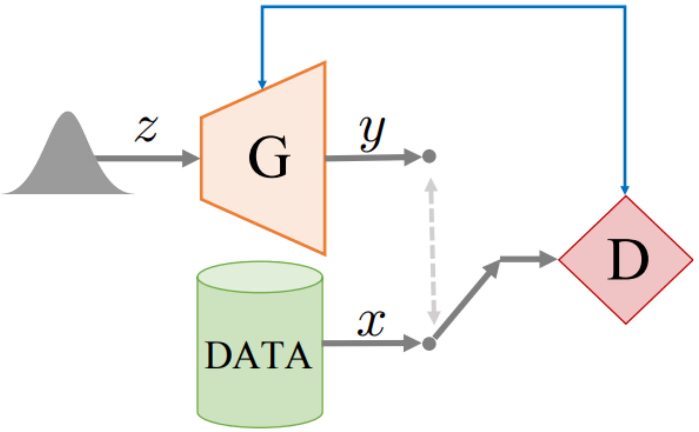
4.1. Learning representations of 3D point clouds

The input to our AE network is a point cloud with 2048 points (2048×3 matrix), representing a 3D shape. The encoder architecture follows the design principle of (Qi et al., 2016a): 1-D convolutional layers with kernel size 1 and an increasing number of features; this approach encodes every point *independently*. A "symmetric", permutation-invariant function (e.g. a max pool) is placed after the convolutions to produce a joint representation. In our implementation we use 5 1-D convolutional layers, each followed by a ReLU (Nair & Hinton, 2010) and a batch-normalization layer (Ioffe & Szegedy, 2015). The output of the last convolutional layer is passed to a feature-wise maximum to produce a k-dimensional vector which is the basis for our latent space. Our decoder transforms the latent vector using 3 fully connected layers, the first two having ReLUs, to produce a 2048×3 output. For a permutation invariant objective, we explore both the EMD approximation and the CD (Section 2) as our structural losses; this yields two distinct AE models, referred to as AE-EMD and AE-CD.

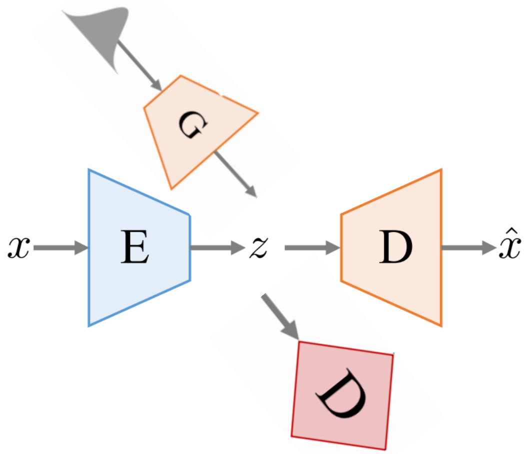


LatentGAN 模型

Raw point cloud GAN (r-GAN) Our first GAN operates on the raw 2048×3 point set input. The architecture of the discriminator is identical to the AE (modulo the filter-sizes and number of neurons), without any batch-norm and with leaky ReLUs (Maas et al., 2013) instead or ReLUs. The output of the last fully connected layer is fed into a sigmoid neuron. The generator takes as input a Gaussian noise vector and maps it to a 2048×3 output via 5 FC-ReLU layers.

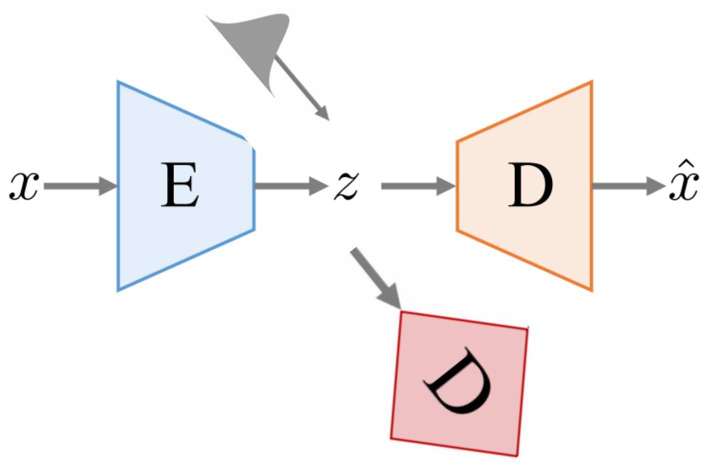


Latent-space GAN (l-GAN) For our l-GAN, instead of operating on the raw point cloud input, we pass the data through a pre-trained autoencoder, which is trained separately for each object class with the EMD (or CD) loss function. Both the generator and the discriminator of the l-GAN then operate on the bottleneck variables of the AE. Once the training of GAN is over, we convert a code learned by the generator into a point cloud by using the AE's decoder. Our chosen architecture for the l-GAN, which was used throughout our experiments, is *significantly* simpler than the one of the r-GAN. Specifically, an MLP generator of a single hidden layer coupled with an MLP discriminator of two hidden layers suffice to produce measurably good and realistic results.



LatentGAN 模型

Gaussian mixture model In addition to the l-GANs, we also fit a family of Gaussian Mixture Models (GMMs) on the latent spaces learned by our AEs. We experimented with various numbers of Gaussian components and diagonal or full covariance matrices. The GMMs can be turned into point cloud generators by first sampling the fitted distribution and then using the AE's decoder, similarly to the l-GANs.



Metrics Two permutation-invariant metrics for comparing unordered point sets have been proposed in the literature (Fan et al., 2016). On the one hand, the *Earth Mover's* distance (EMD) (Rubner et al., 2000) is the solution of a transportation problem which attempts to transform one set to the other. For two equally sized subsets $S_1 \subseteq R^3, S_2 \subseteq R^3$, their EMD is defined by

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \rightarrow S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2$$

where ϕ is a bijection. As a loss, EMD is differentiable almost everywhere. On the other hand, the *Chamfer* (pseudo)-distance (CD) measures the squared distance between each point in one set to its nearest neighbor in the other set:

$$d_{CH}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2.$$

CD is differentiable and compared to EMD more efficient to compute.

AE loss	MMD-CD		MMD-EMD	
	Train	Test	Train	Test
CD	**0.0004**	**0.0012**	0.068	0.075
EMD	0.0005	0.0013	**0.042**	**0.052**

Table 1. Generalization of AEs as captured by MMD. Measurements for reconstructions on the training and test splits for an AE trained with either the CD or EMD loss and data of the chair class; Note how the MMD favors the AE that was trained with the same loss as the one used by the MMD to make the matching.

	A	B	C	D	E	ours EMD	ours CD
MN10	79.8	79.9	-	80.5	91.0	**95.4**	**95.4**
MN40	68.2	75.5	74.4	75.5	83.3	84.0	**84.5**

Table 2. Classification performance (in %) on ModelNet10/40. Comparing to A: SPH (Kazhdan et al., 2003), B: LFD (Chen et al., 2003), C: T-L-Net (Girdhar et al., 2016), D: VConv-DAE (Sharma et al., 2016), E: 3D-GAN (Wu et al., 2016).

Model	Type	JSD	MMD-CD	MMD-EMD	COV-EMD	COV-CD
A	MEM	0.017	0.0018	0.063	78.6	79.4
B	RAW	0.176	0.0020	0.123	19.0	52.3
C	CD	0.048	0.0020	0.079	32.2	59.4
D	EMD	0.030	0.0023	0.069	57.1	59.3
E	EMD	0.022	0.0019	0.066	66.9	67.6
F	GMM	**0.020**	**0.0018**	**0.065**	**67.4**	**68.9**

Table 3. Evaluating 5 generators on the *test* split of the chair dataset on epochs/models selected via minimal JSD on the validation-split. We report: A: sampling-based memorization baseline, B: r-GAN, C: l-GAN (AE-CD), D: l-GAN (AE-EMD) , E: l-WGAN (AE-EMD), F: GMM (AE-EMD).

LatentGAN 模型

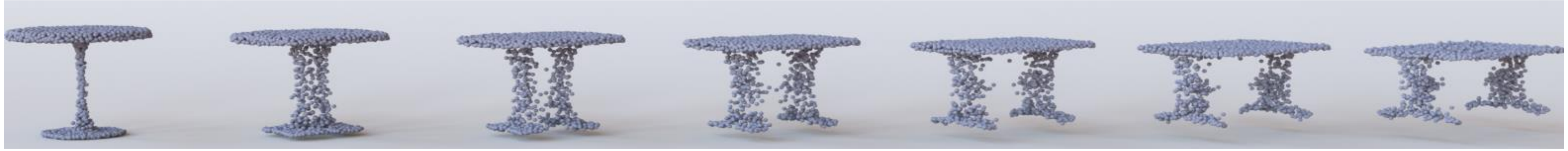


Figure 2. Interpolating between different point clouds, using our latent space representation.

Figure 11. Interpolating between different point clouds (left and right-most of each row), using our latent space representation. Note the interpolation between structurally and topologically different shapes. **Note**: for all our illustrations that portray point clouds we use the Mitsuba renderer (Jakob, 2010).

LatentGAN 模型

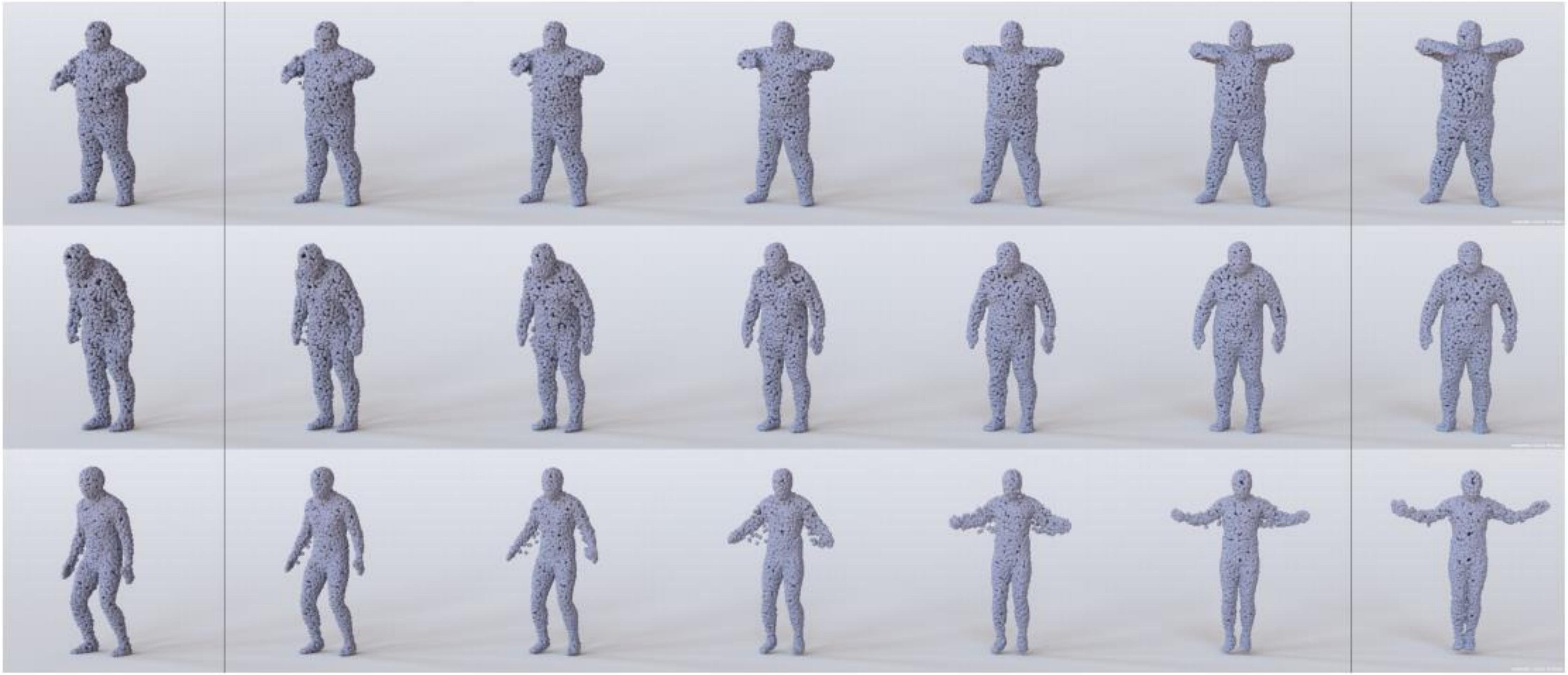


Figure 14. Interpolating between different point clouds from the *test* split (left and right-most of each row) of the D-FAUST dataset of (Bogo et al., 2017). These linear interpolations have captured some of the dynamics of the corresponding motions: 'chicken-wings' (first row), 'shake shoulders' (second row) and 'jumping jacks' (third row). Compare to Fig.13 that contains ground-truth point clouds in the same time interval.

LatentGAN 模型

Figure 4. Point cloud *completions* of a network trained with partial and complete (input/output) point clouds and the EMD loss. Each triplet shows the partial input from the test split (left-most), followed by the network's output (middle) and the complete ground-truth (right-most).

Figure 5. Synthetic point clouds generated by samples produced with l-GAN (top) and 32-component GMM (bottom), both trained on the latent space of an AE using the EMD loss.

LatentGAN 模型

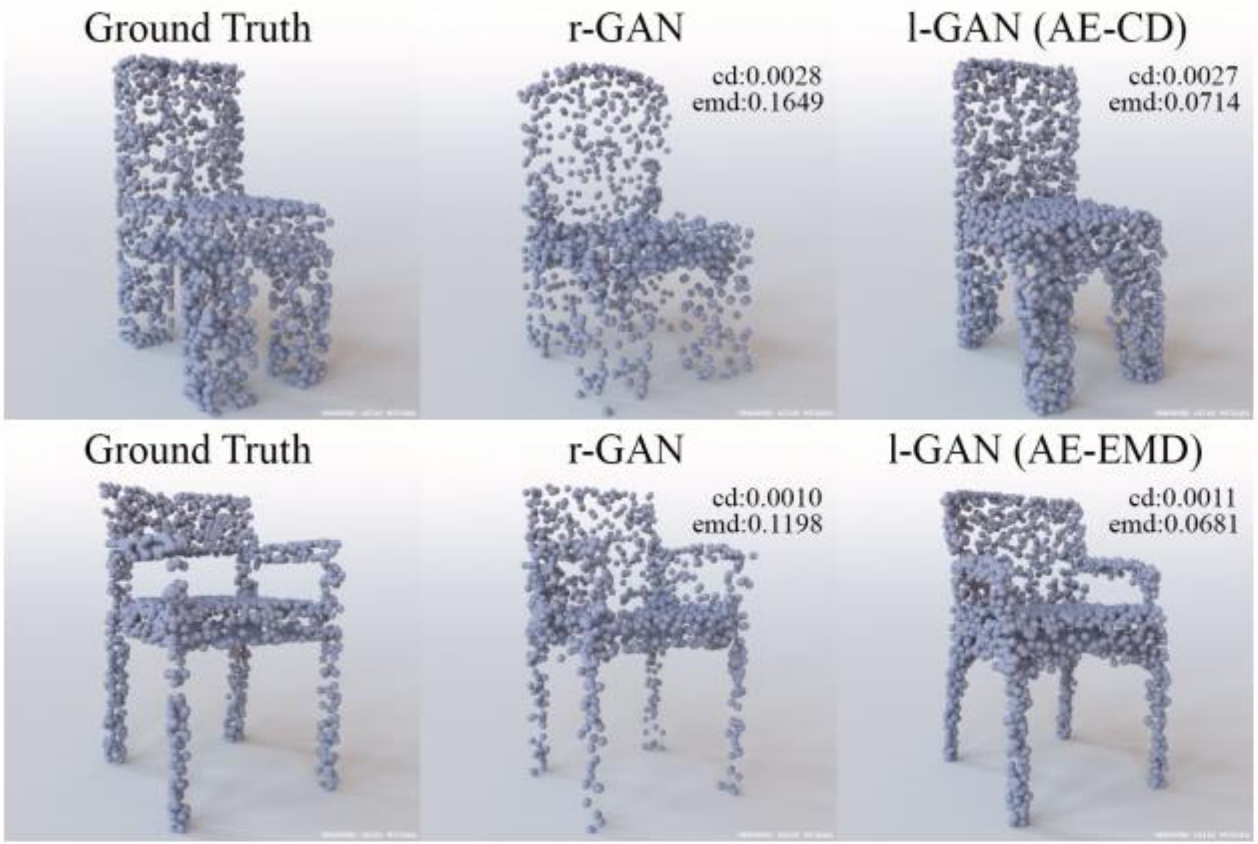


Figure 7. The CD distance is less faithful than EMD to visual quality of synthetic results; here, it favors r-GAN results, due to the overly high density of points in the seat part of the synthesized point sets.

AE loss	MMD-CD		MMD-EMD	
	Train	Test	Train	Test
CD	**0.0004**	**0.0012**	0.068	0.075
EMD	0.0005	0.0013	**0.042**	**0.052**

Table 1. Generalization of AEs as captured by MMD. Measurements for reconstructions on the training and test splits for an AE trained with either the CD or EMD loss and data of the chair class; Note how the MMD favors the AE that was trained with the same loss as the one used by the MMD to make the matching.

Mandikal, etc. Dense 3D Point Cloud Reconstruction Using a Deep Pyramid Network. WACV19

LatentGAN 模型

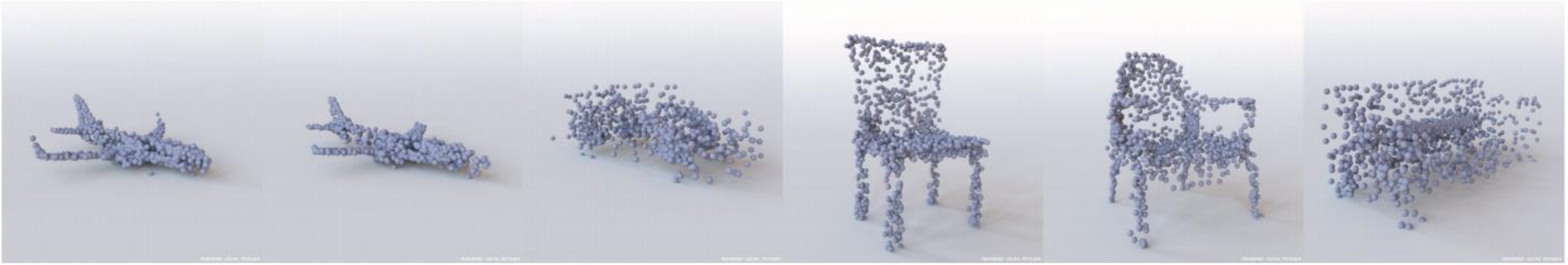


Figure 12. Synthetic results produced by the r-GAN. From left to right: airplanes, car, chairs, sofa.

PC2PC 模型

Published as a conference paper at ICLR 2020

UNPAIRED POINT CLOUD COMPLETION ON REAL SCANS USING ADVERSARIAL TRAINING

Xuelin Chen
Shandong University
University College London

Baoquan Chen
Peking University

Niloy J. Mitra
University College London
Adobe Research London

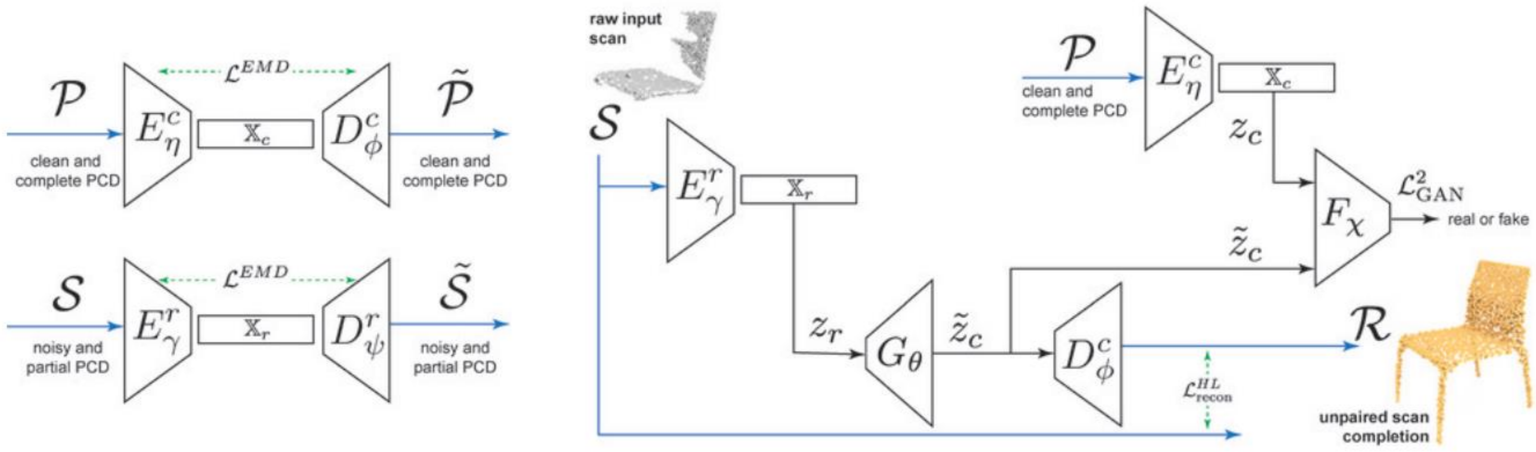


Figure 2: Unpaired Scan Completion Network.

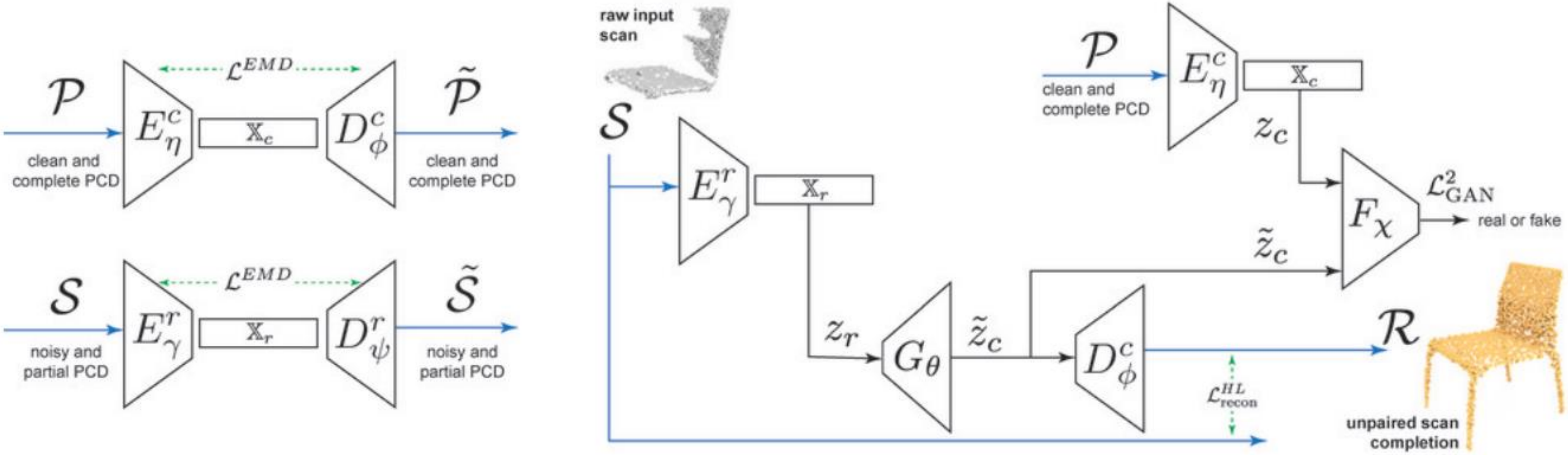


Figure 2: Unpaired Scan Completion Network.

$$\mathcal{L}^{\mathrm{EMD}}(\eta, \phi) = \mathbb{E}_{\mathcal{P}\sim p_{\mathrm{complete}}} d(\mathcal{P}, D^c_\phi(E^c_\eta(\mathcal{P})))$$

$$\mathcal{L}^{\mathrm{EMD}}(\gamma, \psi) = \mathbb{E}_{\mathcal{S}\sim p_{\mathrm{raw}}} d(\mathcal{S}, D^r_\psi(E^r_\gamma(\mathcal{S})))$$

$$\gamma = \eta \text{ and } \psi = \phi$$

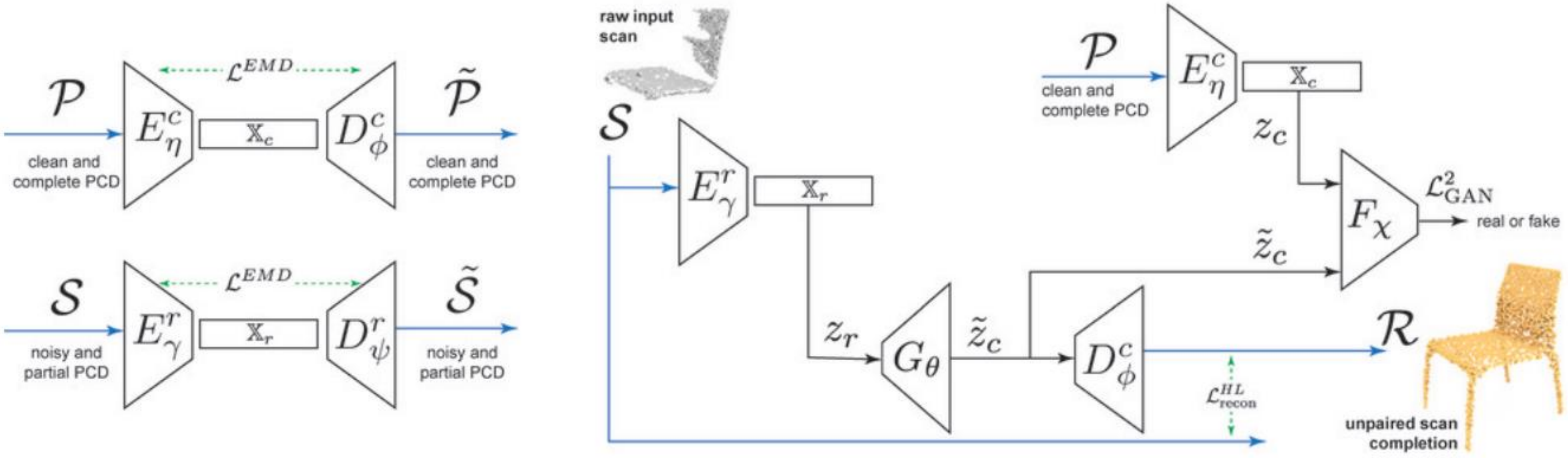


Figure 2: Unpaired Scan Completion Network.

$$\min_{\theta}\max_{\chi}\mathbb{E}_{x\sim p_{\text{clean-complete}}}\left[\log\left(F_\chi\left(E^c_\eta(x)\right)\right)\right]+\mathbb{E}_{y\sim p_{\text{noisy-partial}}}\left[\log\left(1-F_\chi\left(G_\theta(E^r_\gamma(y))\right)\right)\right]. \quad (3)$$

$$\mathcal{L}_F(\chi) = \mathbb{E}_{x\sim p_{\text{clean-complete}}}\left[F_\chi\left(E^c_\eta(x)\right)-1\right]^2+\mathbb{E}_{y\sim p_{\text{noisy-partial}}}\left[F_\chi\left(G_\theta(E^r_\gamma(y))\right)\right]^2 \quad (4)$$

$$\mathcal{L}_G(\theta) = \mathbb{E}_{y\sim p_{\text{noisy-partial}}}\left[F_\chi\left(G_\theta(E^r_\gamma(y))\right)-1\right]^2. \quad (5)$$

PC2PC 模型

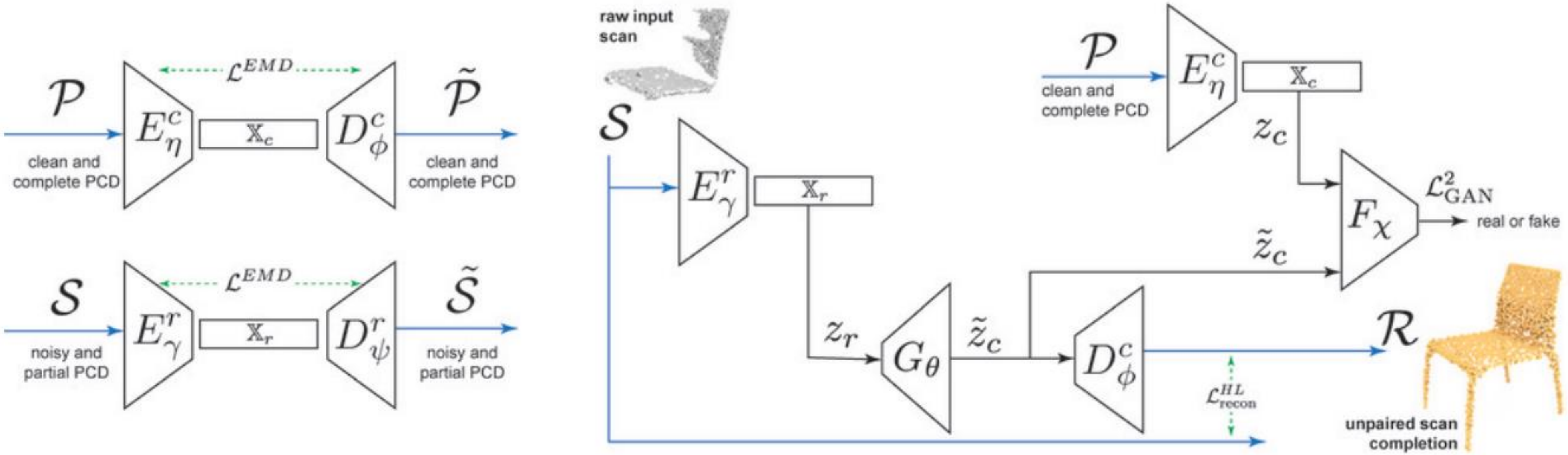


Figure 2: Unpaired Scan Completion Network.

Figure 3: Effect of unpaired scan completion without (Equation 5) and with HL term (Equation 6). Without the HL term, the network produces a clean point set for a complete chair, that is different in shape from the input. With the HL term, the network produces a clean point set that matches the input.

PC2PC 模型

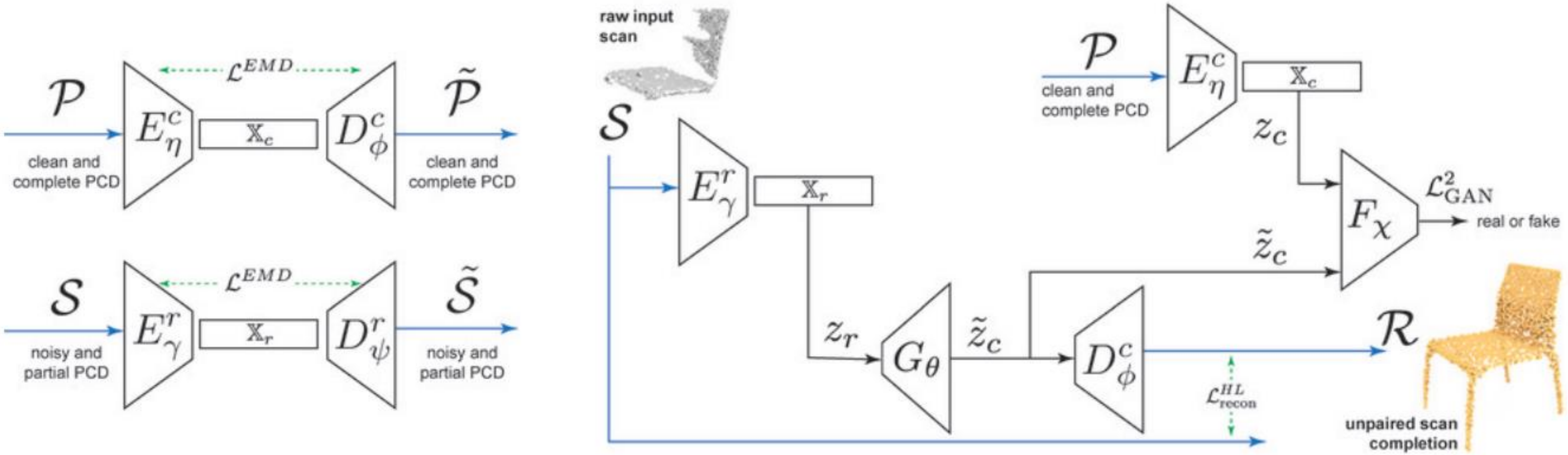
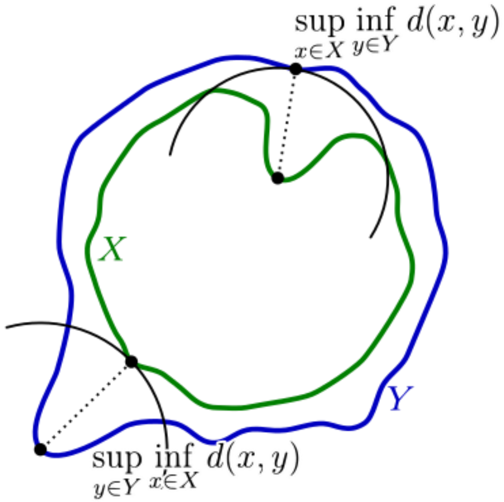


Figure 2: Unpaired Scan Completion Network.



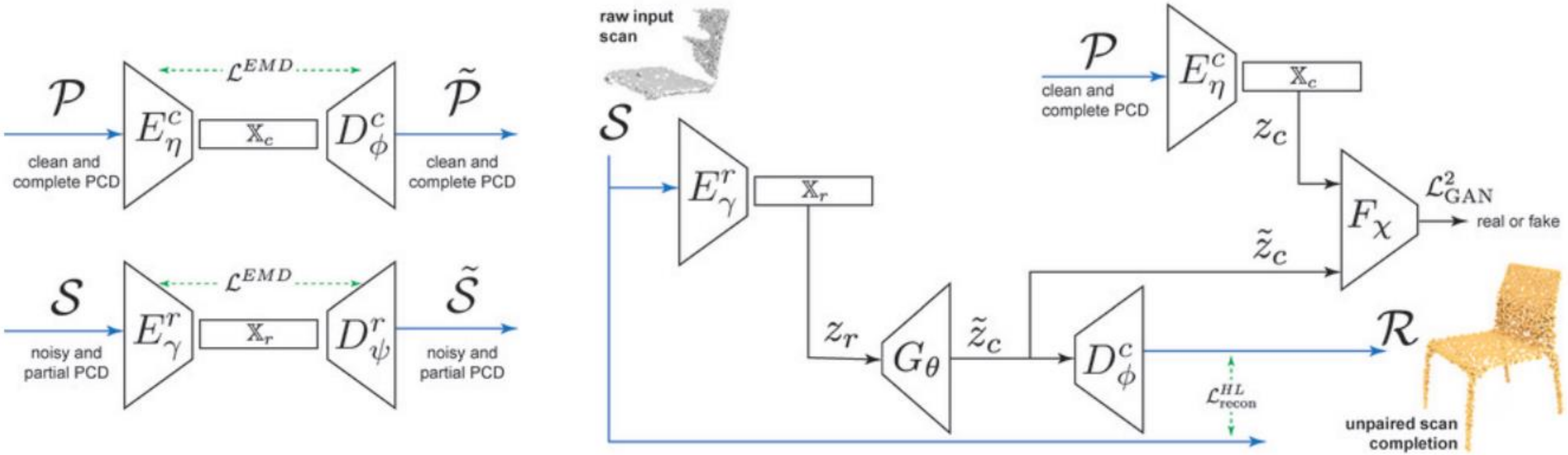


Figure 2: Unpaired Scan Completion Network.

$$\min_{\theta} \max_{\chi} \mathbb{E}_{x \sim p_{\text{clean-complete}}} \left[\log\left(F_\chi\left(E^c_\eta(x)\right)\right)\right] + \mathbb{E}_{y \sim p_{\text{noisy-partial}}} \left[\log\left(1 - F_\chi\left(G_\theta(E^r_\gamma(y))\right)\right)\right]. \quad (3)$$

$$\mathcal{L}_F(\chi) = \mathbb{E}_{x \sim p_{\text{clean-complete}}} \left[F_\chi\left(E^c_\eta(x)\right) - 1\right]^2 + \mathbb{E}_{y \sim p_{\text{noisy-partial}}} \left[F_\chi\left(G_\theta(E^r_\gamma(y))\right)\right]^2 \quad (4)$$

$$\cancel{\mathcal{L}_G(\theta) = \mathbb{E}_{y \sim p_{\text{noisy-partial}}} \left[F_\chi\left(G_\theta(E^r_\gamma(y))\right) - 1\right]^2.} \quad (5)$$

$$\mathcal{L}_G(\theta) = \alpha \mathbb{E}_{y \sim p_{\text{noisy-partial}}} \left[F_\chi\left(G_\theta(E^r_\gamma(y))\right) - 1\right]^2 + \beta \mathcal{L}^{\text{HL}}_{\text{recon}}\left(\mathcal{S}, D^c_\psi\left(G_\theta(E^r_\gamma(\mathcal{S}))\right)\right)$$

Table 1: **Completion plausibility on synthetic scans and real-world scans and effects of data distribution discrepancy.** (Left) Plausibility comparison on synthetic scans and real-world scans. Synthetic scans includes test data from 3D-EPN, real-world scans includes ScanNet and Matterport3D test data. 3D-EPN failed to produce good completions on real-world data. (Right) On our synthetic data, supervised methods trained on other simulated partial scans produce worse results on partial scans with different data distribution.

		Raw input	3D-EPN	PCN	Ours
Synthetic	chair	73.1	77.3	85.0	**91.5**
	table	52.5	71.2	72.0	**80.6**
Real-world	chair	71.4	7.1	78.6	**94.3**
	table	47.8	4.4	69.6	**81.2**

	3D-EPN			PCN			Ours		
model	acc.	comp.	F1	acc.	comp.	F1	acc.	comp.	F1
chair	39.6	61.8	48.2	49.3	76.0	59.8	80.7	80.8	**80.8**
car	43.8	62.3	51.4	63.2	81.4	71.2	82.6	80.7	**81.7**
table	36.6	61.0	45.8	62.3	80.6	70.3	83.1	84.5	**83.8**
plane	17.1	57.6	26.3	67.1	85.4	75.1	94.4	92.7	**93.6**

Table 2: **Comparison with baselines on the 3D-EPN dataset**. Note that 3D-EPN and PCN require paired supervision data, while ours does not. Ours outperforms 3D-EPN and achieves comparable results to PCN. Furthermore, after adapted to leverage the ground truth data as well, our method achieves similar performance to PCN.

	AE			EPN (**fully supervised**)			PCN (**fully supervised**)			Ours (**unsupervised**)			Ours+ (supervised)		
model	acc.	comp.	F1	acc.	comp.	F1	acc.	comp.	F1	acc.	comp.	F1	acc.	comp.	F1
boat	89.6	81.4	85.3	82.4	81.4	81.9	92.6	93.4	**93.0**	86.6	84.7	85.6	89.8	92.0	90.9
car	81.3	71.1	75.9	69.8	81.7	75.3	97.3	96.1	**96.7**	88.9	87.6	88.2	93.5	92.8	93.1
chair	79.9	68.5	73.8	61.7	76.9	68.5	91.1	90.6	**90.9**	78.7	77.4	78.0	82.3	83.3	82.8
dresser	68.9	64.2	66.5	58.4	72.7	64.8	93.5	91.5	**92.5**	75.8	76.5	76.2	87.4	91.5	89.4
lamp	75.9	79.6	77.7	60.8	67.8	64.1	82.9	88.3	**85.5**	71.3	80.2	75.5	76.6	86.3	81.2
plane	97.6	95.1	96.3	78.1	93.5	85.1	98.3	98.2	**98.2**	97.2	95.9	96.5	95.6	94.8	95.2
sofa	80.3	64.0	71.2	65.0	72.6	68.6	91.5	90.8	**91.1**	68.2	72.3	70.2	81.0	87.0	83.9
table	82.8	72.5	77.3	56.8	75.1	64.7	93.4	89.2	**91.2**	82.2	77.8	80.0	81.2	81.4	81.3

$$\mathcal{L}_G(\theta) = \alpha \mathbb{E}_{y\sim p_{\text{noisy-partial}}}\left[F_\chi(G_\theta(E_\gamma^r(y))) - 1\right]^2 + \beta \mathcal{L}_{\text{recon}}^{\text{HL}}(\mathcal{S}, D_\psi^c(G_\theta(E_\gamma^r(\mathcal{S}))))$$

$\alpha = 0$ and use EMD loss as L_{recon}

PC2PC 模型

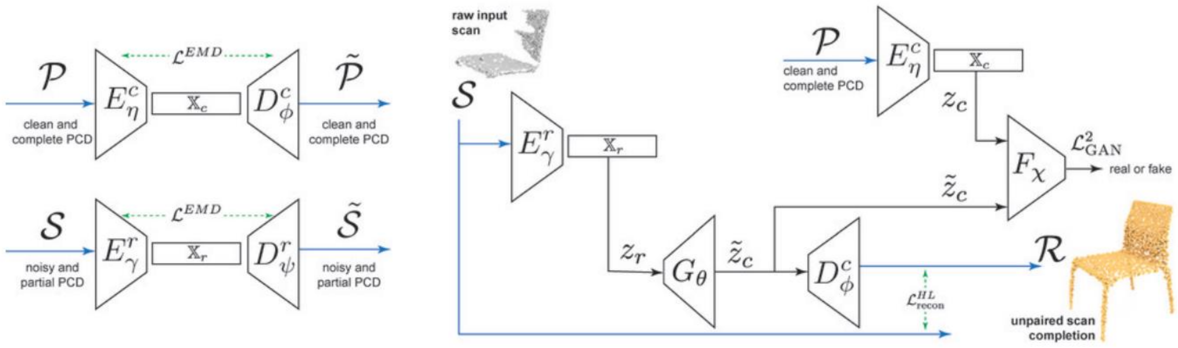


Figure 2: Unpaired Scan Completion Network.

- Ours with partial AE, uses encoder E^r_γ and decoder D^r_ψ that are trained to reconstruct partial point sets for the latent space of partial input.
- Ours with EMD loss, uses EMD as the reconstruction loss.
- Ours without GAN, "switch off" the GAN module by simply setting $\alpha = 0$ and $\beta = 1$, to verify the effectiveness of using adversarial training in our network.
- Ours with reconstruction loss, removes the reconstruction loss term by simply setting $\alpha = 1$ and $\beta = 0$, to verify the effectiveness of the reconstruction loss term in generator loss.

$$\mathcal{L}_G(\theta) = \alpha \mathbb{E}_{y\sim p_{\text{noisy-partial}}}\left[F_\chi\left(G_\theta(E^r_\gamma(y))\right) - 1\right]^2 + \beta \mathcal{L}^{\text{HL}}_{\text{recon}}(\mathcal{S}, D^c_\psi(G_\theta(E^r_\gamma(\mathcal{S}))))$$

Table 4: Ablation study showing the importance of various design choices in our proposed network. On 3D-EPN.

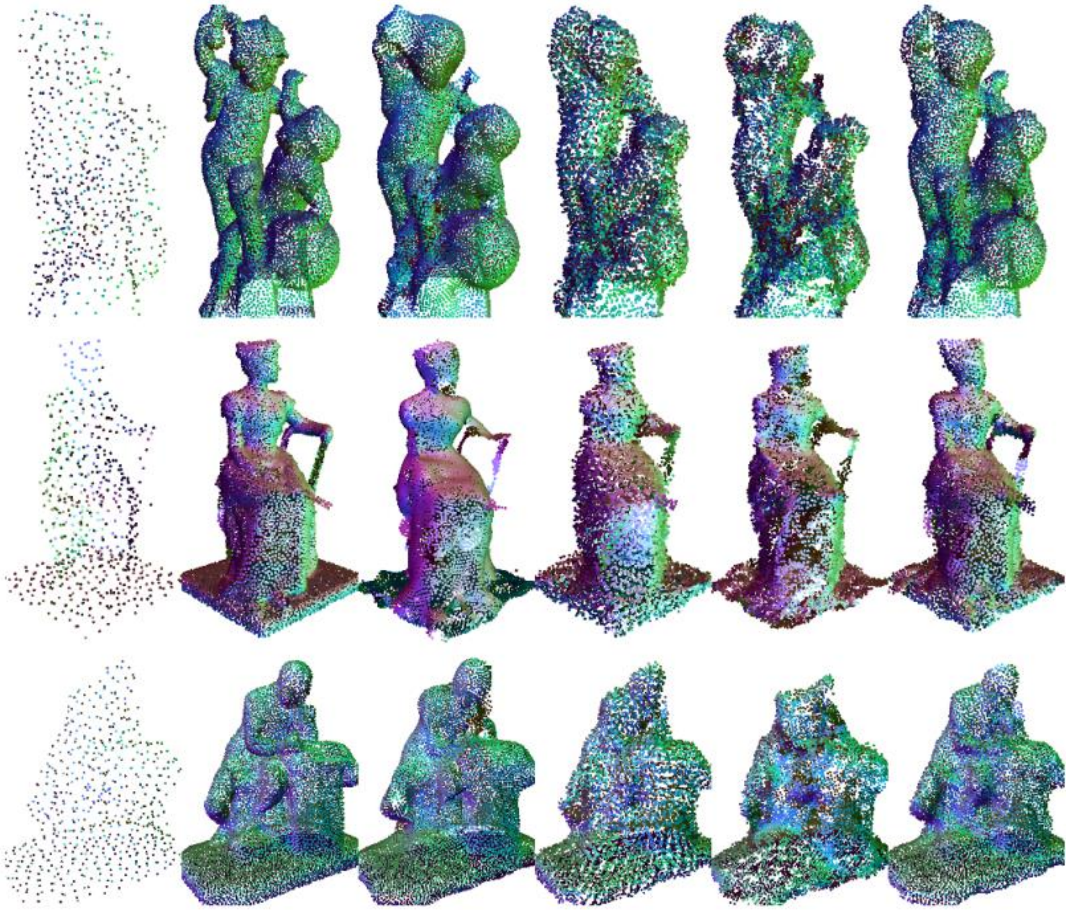
	Ours w/ partial AE			Ours w/ EMD			Ours w/o GAN			Ours w/o Recon.			Ours		
	acc.	comp.	F1	acc.	comp.	F1	acc.	comp.	F1	acc.	comp.	F1	acc.	comp.	F1
boat	75.1	75.4	75.2	82.0	84.8	83.4	47.4	93.1	62.8	44.4	38.1	41.0	86.6	84.7	**85.6**
car	88.9	87.6	88.2	76.0	76.8	76.4	46.2	88.3	60.7	72.2	72.7	72.5	88.9	87.7	**88.3**
chair	64.1	66.7	65.4	78.6	76.4	77.5	41.3	79.8	54.4	75.6	75.1	75.3	78.7	77.4	**78.0**
dresser	67.4	68.6	68.0	71.4	72.3	71.9	44.2	74.4	55.4	20.9	21.9	21.4	75.8	76.5	**76.2**
lamp	64.0	74.8	69.0	69.9	79.0	74.2	28.6	84.7	42.8	15.6	22.2	18.3	71.3	80.2	**75.5**
plane	94.3	94.9	94.6	96.8	95.4	96.1	41.2	98.3	58.1	87.1	84.7	85.9	97.2	95.9	**96.5**
sofa	64.8	67.3	66.0	68.6	69.8	69.2	38.6	75.6	51.1	55.1	58.0	56.5	68.2	72.3	**70.2**
table	76.0	77.6	76.8	81.5	75.1	78.2	23.0	59.3	33.1	27.4	23.4	25.2	82.2	77.8	**80.0**

PC2PC 模型

Figure 1: We present a point-based shape completion network that can be directly used on raw scans without requiring paired training data. Here we show a sampling of results from the ScanNet, Matterport3D, 3D-EPN, and KITTI datasets.

Figure 4: Qualitative comparisons on real-world data, which includes partial scans of ScanNet chairs and tables, Matterport3D chairs and tables, and KITTI cars. We show the partial input in grey and the corresponding completion in gold on the right.

点云的表示其他经典方法



Sketchfab: 90 (training) + 13 (testing)
highly detailed 3D models

PU-Net: Point Cloud Upsampling Network

CVPR 18

Lequan Yu* [1,3] Xianzhi Li* [1] Chi-Wing Fu [1,3] Daniel Cohen-Or [2] Pheng-Ann Heng [1,3]
[1] The Chinese University of Hong Kong [2] Tel Aviv University
[3] Guangdong Provincial Key Laboratory of Computer Vision and Virtual Reality Technology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
{lqyu,xzli,cwfu,pheng}@cse.cuhk.edu.hk dcor@mail.tau.ac.il

Patch-based Progressive 3D Point Set Upsampling

CVPR 19

Wang Yifan [1] Shihao Wu [1] Hui Huang [2*]
Daniel Cohen-Or [2,3] Olga Sorkine-Hornung [1]

[1] ETH Zurich [2] Shenzhen University [3] Tel Aviv University

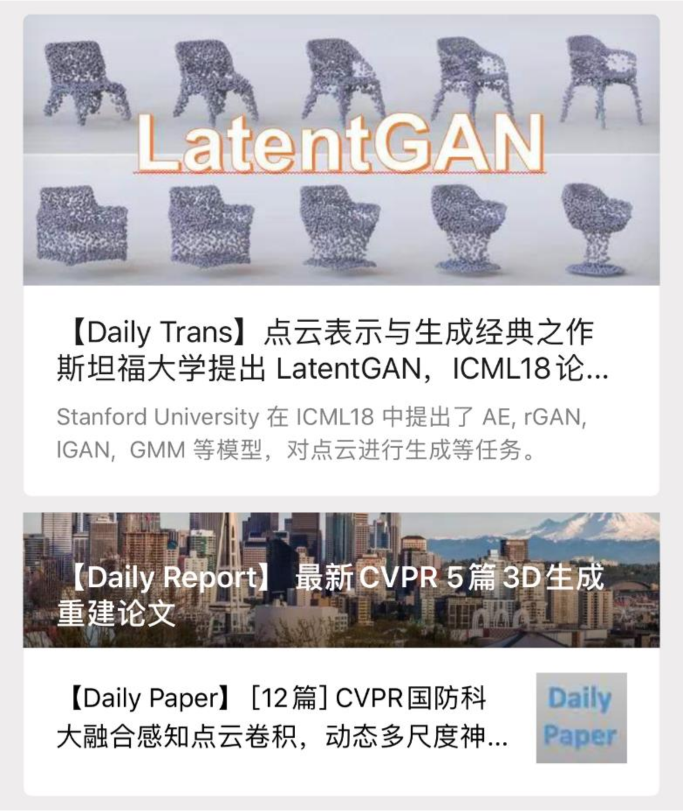
感谢聆听

Thanks for Listening

欢迎关注

公众号：3D Daily

后台回复“LatentGAN”，获取论文讲解、翻译和相关文章。



加三弟微信
进入讨论群

欢迎关注3D视觉工坊

我们这里有3D视觉算法、VSLAM算法、计算机视觉、深度学习、自动驾驶、图像处理等干货分享！

如果你也想成为主讲人，欢迎加入我们。

➢ 报名方式：请发送邮件至vision3d@yeah.net

公众号

交流群请添加客服