Supplementary Material for "TearingNet: Point Cloud Autoencoder to Learn Topology-Friendly Representations"

Jiahao Pang Duanshun Li* Dong Tian
InterDigital, Princeton, NJ, USA
jiahao.pang@interdigital.com, duanshun@ualberta.ca, dong.tian@interdigital.com

I. Introduction

In this supplementary material, we first elaborate on the architecture of the Tearing network (T-Net) in Section II. Then additional experimentation alongside with a demonstration on *point cloud interpolation* are presented in Section III.

II. Architecture of the Tearing Network

The proposed Tearing network is composed of two sets of shared MLP layers. It has a similar model size as the FoldingNet [7] (or F-Net in our work). Its detailed structure is shown in Figure I.

First, the (replicated) codeword $\mathbf{C}$, the first F-Net output $\mathbf{X}^{(1)}$, and the initial 2D grid $\mathbf{U}^{(0)}$ are concatenated as a $45 \times 45 \times 517$ volume which is fed to the first series of shared MLP layers. This series of MLP layers have output dimensions of 512, 512, and 64, respectively.

Next, a second input volume is formed similar to the previous step. Additionally, the 64-dimension feature output from the previous step is further concatenated to form a $45 \times 45 \times 581$ volume. Finally, it is fed to the second series of shared MLP layers. This series of MLP layers have output dimensions of 512, 512, and 2, respectively. The final T-Net output of size $45 \times 45 \times 2$ (reshaped to 2025×2) is added to $\mathbf{U}^{(0)}$, leading to the modified 2D grid $\mathbf{U}^{(1)}$.

Unlike the AtlasNet [4], AtlasNetV2 [3], and the Point Capsule Network [8] which require multiple elementary encoders/decoders for reconstruction, the proposed TearingNet, advantageously, needs only one F-Net and one T-Net in the decoder to handle complex scene point clouds.

III. More on Experimentation

This section presents more details on our training process (Section III-A), followed by the generation of the multi-object datasets (Section III-B). We then present additional quantitative and qualitative evaluation (Section III-C). In the end, a demonstration of point cloud interpolation is provided to gain more insights about the TearingNet codewords (Section III-D).

III-A. More Training Details

Our TearingNet is trained with a two-step strategy to fully squeeze its advantages. The Encoder network (E-Net) and the Folding network (F-Net) are first pre-trained together using a modified version of Eq. (4), where we down-scale its second term, *i.e.*, $\frac{1}{m}\sum_{\widehat{\mathbf{x}}\in\widehat{\mathbf{X}}}\min_{\mathbf{x}\in\mathbf{X}}\|\mathbf{x}-\widehat{\mathbf{x}}\|_2$, by weighting it with a small factor such as 0.1. This is to let the first term dominates, so that the preliminary reconstruction roughly *encloses* the ground-truth surface. In the second step, we adopt a smaller learning rate and train the overall TearingNet with the intact augmented CD of Eq. (4)—this fine-tuning step lets the Tearing network (T-Net) gradually *carve* the details of the reconstructed point cloud.

This training strategy is well suited to the design of the pair of modules, T-Net and F-Net. For instance, it leads to an improvement in CD ($\times 10^{-2}$) from 6.66 to 6.43 on KIMO-4 compared to training from scratch. We also note that CD is observed to be inferior to EMD with respect to visual quality [1, 5] due to a phenomenon we quoted as *point-collapse*—points are over-populated in some regions of the reconstructed point clouds. For example, see the AtlasNet result in the last row of Table 1 in the paper, where the point distribution is unbalanced. With this two-step training strategy, we observe that the point-collapse phenomenon is greatly relieved. This finding holds for variants of CD using squared distance terms, and summing the two distance terms instead of taking the $\max\{\cdot,\cdot\}$. Deeper analysis is left for future investigation.

III-B. More Details on Multi-Object Datasets

Both the KITTI Multi-Object (KIMO) and the CAD Model Multi-Object (CAMO) datasets simulate driving scenes by putting together object point clouds. Different from the KIMO datasets coming from real LiDAR data, the CAMO datasets are assembled using CAD models. It is synthesized with point clouds labeled as `Person`, `Car`, `Cone`

*Work done while the author was an intern at InterDigital.

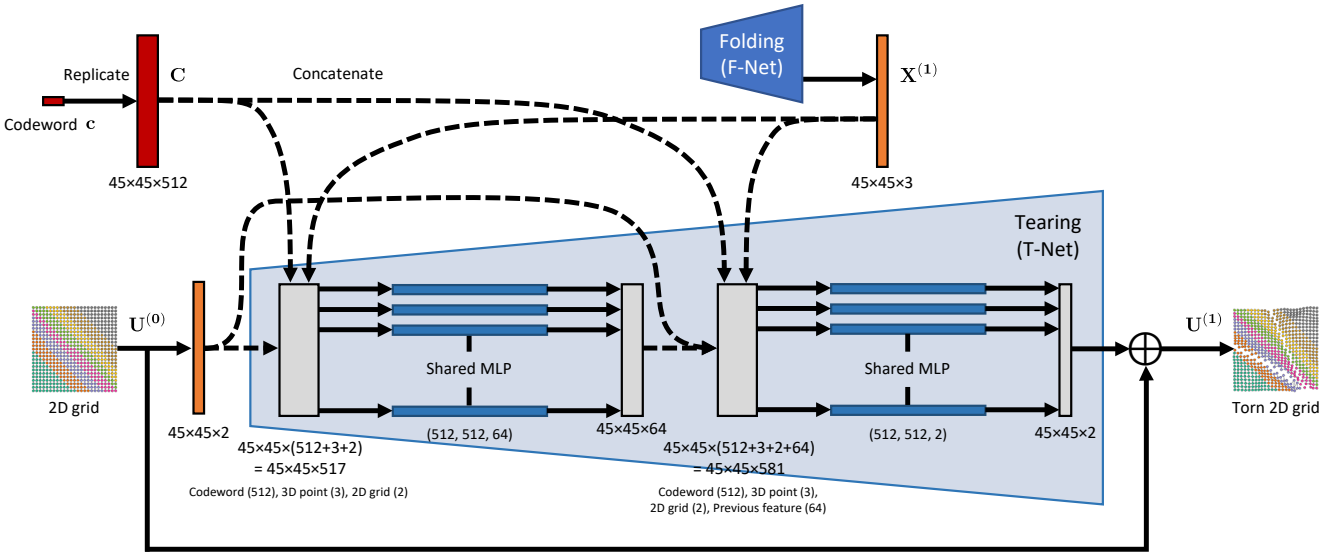


Figure I. Architecture of the Tearing network (T-Net), which employs two series of shared MLP layers to modify the input 2D grid.

and `Plant` from ModelNet40 [6] and point clouds labeled as `Motorbike` from ShapeNet [2].

In a generated multi-object dataset (CAMO or KIMO), the proportion of point clouds with k objects follows a *binomial* distribution. Particularly, we randomly let each of the grid on the $K \times K$ playground to be occupied by an object with a probability $p = 0.5$. Then the case with 0 object on the playground is excluded. When placing an object on a grid, we first normalize it within a sphere of diameter 1, then translate it vertically to let its bottom touch the plane of the "ground". In this way, we generate multi-object scenes with ample topological configurations.

III-C. More Performance Comparisons

Based on the same settings as introduced in Section 5 of the paper, more evaluation of the proposed TearingNet are performed and presented herein.

Quantitative: We first provide an additional quantitative evaluation of the proposed TearingNet with the CAMO datasets. Specifically, the 3D point cloud reconstruction results are reported in Table I; while the object counting and object detection results are presented in Table II. By inspecting Table I and Table II we observe similar trends as presented in Section 5.3 of the paper, and again see the superiority of the proposed TearingNet in terms of reconstructing and representing point clouds with ample topologies.

Qualitative: More visual comparisons between the TearingNet and other competing methods are presented in Table III, where we zoom in the objects/regions in the red boxes and show them in the blue boxes for better visualization. Again, the multi-object reconstructions provided by AtlasNet have unbalanced point distributions. For instance, some of its reconstructed objects are sparser than the ground-truths. Moreover, most results of FoldingNet (and some results of AtlasNet) appear to be noisy. In contrast, TearingNet consistently provides reconstructions close to the inputs, with clean and neat appearances.

Table I. Evaluation of reconstruction on the CAMO datasets.

Metrics	Methods	Datasets			
		CA.-3	CA.-4	CA.-5	CA.-6
CD ($\times 10^{-2}$)	LatentGAN	8.05	11.59	15.98	20.07
	AtlasNet	**6.83**	8.76	11.15	13.74
	FoldingNet	**6.79**	8.65	11.06	13.76
	Cascaded F-Net	6.90	8.81	11.20	13.89
	TearingNet$_{\text{TF}}$	6.99	8.76	11.29	13.66]
	TearingNet$_{\overline{\text{GF}}}$	6.88	8.61	10.95	13.25
	TearingNet (Ours)	6.88	**8.59**	**10.86**	**13.15**
	TearingNet$_3$ (Ours)	6.85	**8.56**	**10.78**	**13.09**
EMD	LatentGAN	1.909	2.971	3.371	4.726
	AtlasNet	1.359	2.450	2.449	2.949
	FoldingNet	0.951	1.354	1.966	2.669
	Cascaded F-Net	1.192	1.442	2.077	2.581
	TearingNet$_{\text{TF}}$	0.858	1.212	2.006	2.430
	TearingNet$_{\overline{\text{GF}}}$	**0.774**	1.111	1.679	2.055
	TearingNet (Ours)	**0.780**	**1.074**	**1.651**	**2.049**
	TearingNet$_3$ (Ours)	0.781	**1.103**	**1.610**	**1.994**

III-D. Point Cloud Interpolation

To further understand how the codewords of TearingNet naturally embed the point cloud topology, we inspect if the TearingNet can novelly interpolate between two point clouds [7]. Given two point clouds, their codewords $\mathbf{c}_1$ and $\mathbf{c}_2$ are first computed by our encoder (E-Net). The codewords are then weighted averaged as $(1-w)\,\mathbf{c}_1 + w\,\mathbf{c}_2$ with different weights w ranging from 0 to 1. The averaged codewords are fed to the TearingNet decoder to reconstruct the interpolated point clouds.

Table IV provides four examples of point cloud interpolation (with the torn 2D grids) on different datasets. For both the point clouds and the 2D grids, we draw the edges of the

Table II. Evaluation of object counting and object detection on the CAMO datasets.

Tasks	Methods	Datasets			
		CA.-3	CA.-4	CA.-5	CA.-6
Counting (MAE, $\times 10^{-1}$)	LatentGAN	0.426	4.057	8.764	10.654
	AtlasNet	0.095	2.430	5.601	7.618
	FoldingNet	0.068	1.161	4.225	7.178
	Cascaded F-Net	0.070	1.195	4.456	7.266
	TearingNet$_{\text{TF}}$	0.067	0.734	4.284	7.076
	TearingNet$_{\overline{\text{GF}}}$	0.065	0.663	4.250	7.073
	TearingNet (Ours)	**0.064**	**0.656**	**4.199**	**7.044**
	TearingNet$_3$ (Ours)	**0.064**	**0.645**	**4.203**	**6.988**
Detection (Accuracy, %)	LatentGAN	93.17	63.78	65.65	78.80
	AtlasNet	88.84	73.79	73.58	83.42
	FoldingNet	92.71	80.12	77.10	82.92
	Cascaded F-Net	93.15	82.85	78.81	82.43
	TearingNet$_{\text{TF}}$	93.33	83.35	79.44	83.52
	TearingNet$_{\overline{\text{GF}}}$	**93.44**	83.42	79.72	**84.55**
	TearingNet (Ours	93.42	**83.44**	**79.70**	**84.55**
	TearingNet$_3$ (Ours)	**93.48**	**83.42**	**79.74**	84.54

graph $\widehat{\mathcal{G}}$ (as presented in Section 3.1) to explicitly check how tearing happens. We see the topologies of the point clouds change as the 2D grids are deformed and the graph edges are broken. Especially, for the multi-object point clouds, the objects are split/merged, reshaped, and translated to form new point clouds. We also see that the interpolated point clouds manifest geometric characteristics of both the two input point clouds, *e.g.*, the torus at step $4/7$. It affirms that the learned feature space is highly expressive in terms of geometry, which facilitates the network to generate novel point clouds that have never been seen during training.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In *International Conference on Machine Learning*, pages 40–49, 2018. 1

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2

[3] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. Learning elementary structures for 3D shape generation and matching. In *Adv. Neural Inform. Process. Syst.*, pages 7433–7443, 2019. 1

[4] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3D surface generation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 216–224, 2018. 1

[5] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10130–10139, 2019. 1

[6] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1912–1920, 2015. 2

[7] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. FoldingNet: Point cloud auto-encoder via deep grid deformation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 206–215, 2018. 1, 2

[8] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3D point capsule networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1009–1018, 2019. 1

Table III. Visual comparisons of point cloud reconstructions. Points are colored according to their indices. S: ShapeNet; T: Torus; C: CAMO-5; K: KIMO-5. Objects/regions in the red boxes are zoomed in and shown in the blue boxes.

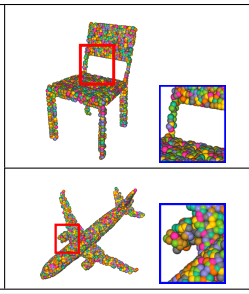
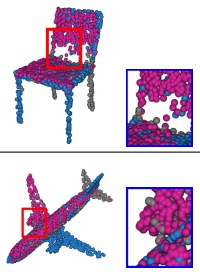
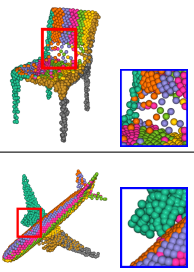
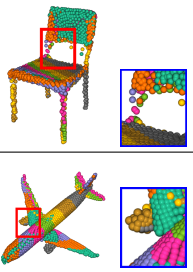
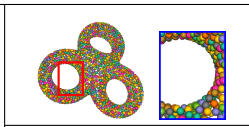
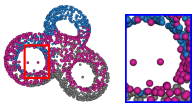
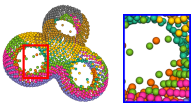
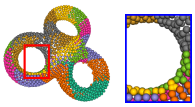
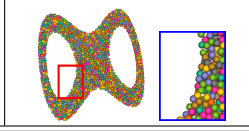
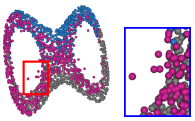
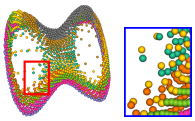
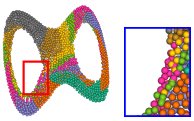
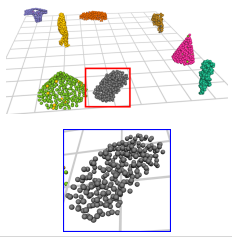
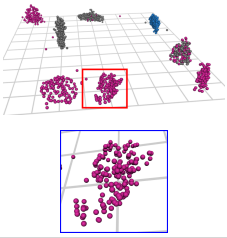
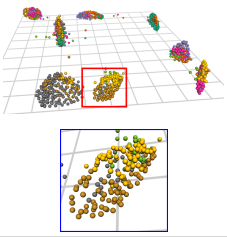
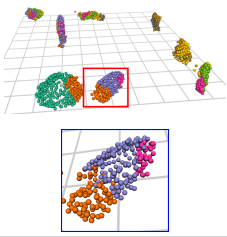
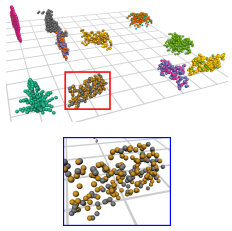
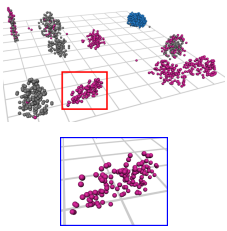
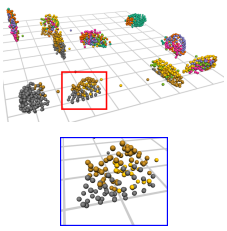
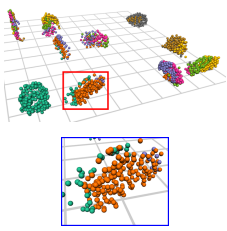
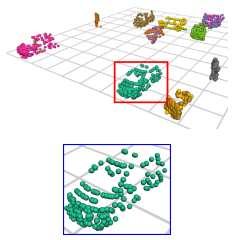
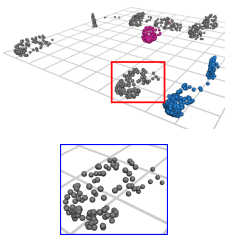
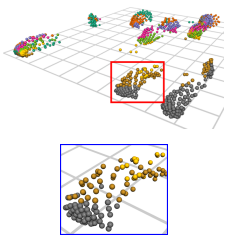
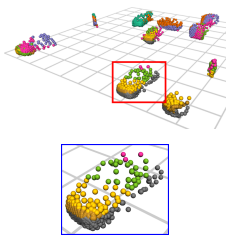
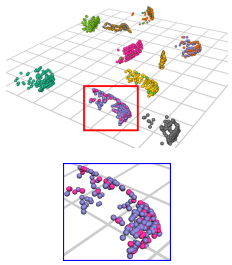
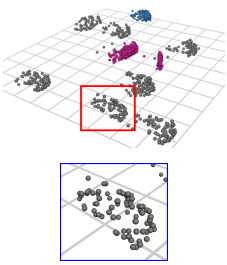
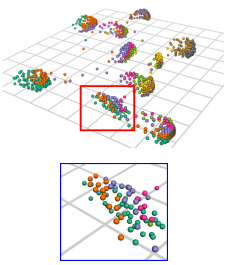
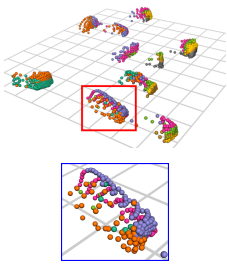
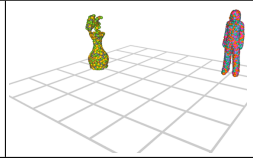
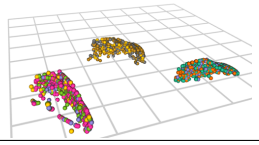
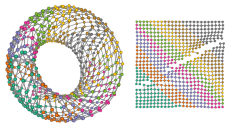
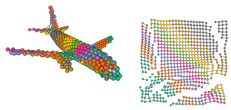
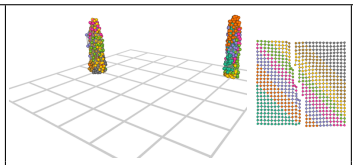
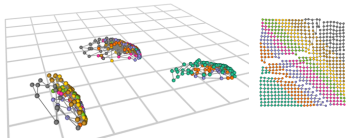
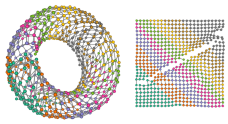
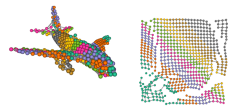
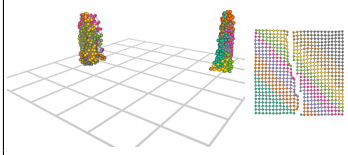
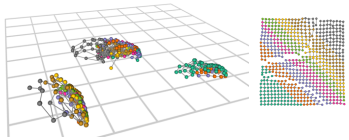
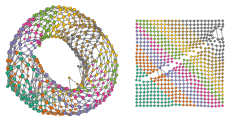
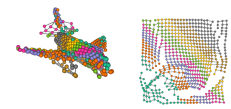
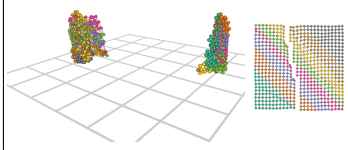
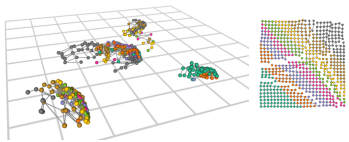
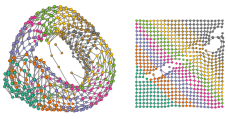
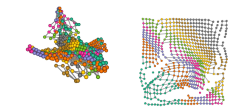
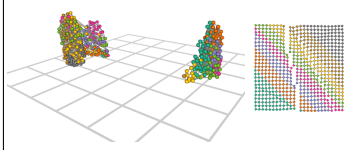
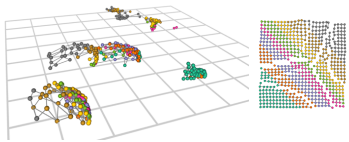
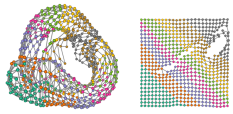
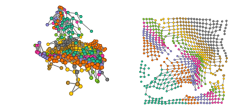
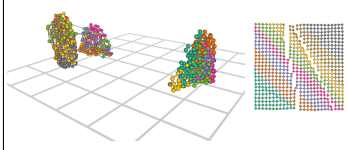
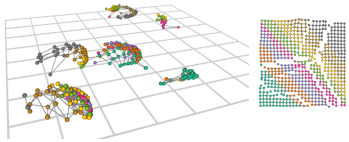
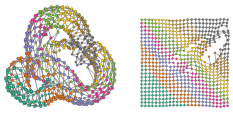
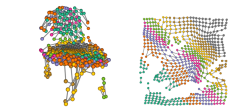
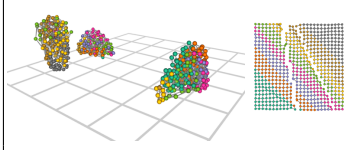
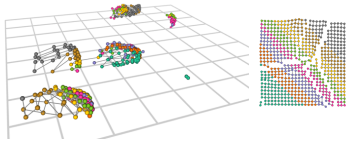
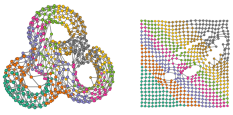
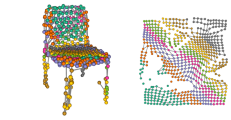
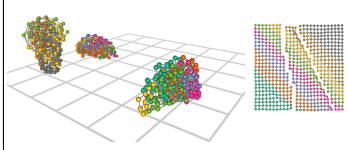
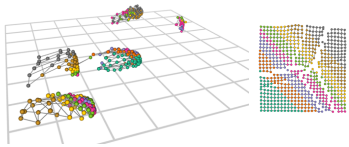
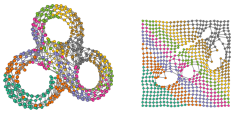
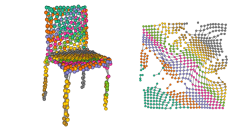
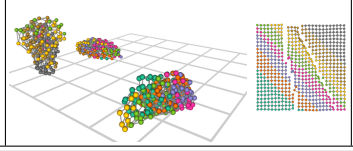
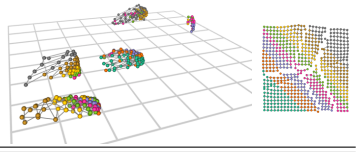
	Ground-truth	AtlasNet	FoldingNet	TearingNet	Torn Grid
S					
					
T					
					
C					
					
K					
					

Table IV. Point cloud interpolation with TearingNet. Points are colored according to their indices. G1, G2 - Ground-truths of the two point clouds. Other rows show interpolations with different weights. Non-zero graph edges are also drawn.

-	Torus	ShapeNet	CAMO-3	KIMO-4
G1				
0/7				
1/7				
2/7				
3/7				
4/7				
5/7				
6/7				
7/7				
G2