"ZERO-SHOT" POINT CLOUD UPSAMPLING

Kaiyue Zhou[1,2], Ming Dong[1], and Suzan Arslanturk[1]*

[1]Department of Computer Science, Wayne State University, USA
[2]Department of Electronic Engineering, Tsinghua University, China
{kyzhou, mdong, suzan.arslanturk} @wayne.edu

ABSTRACT

Recent supervised point cloud upsampling methods are restricted by the size of training data and are limited in terms of covering all object shapes. Besides the challenges faced due to data acquisition, the networks also struggle to generalize on unseen records. In this paper, we present an internal point cloud upsampling approach at a holistic level referred to as "Zero-Shot" Point Cloud Upsampling (ZSPU). Our approach is data agnostic and relies solely on the internal information provided by a particular point cloud without patching in both self-training and testing phases. This single-stream design significantly reduces the training time by learning the relation between low resolution (LR) point clouds and their high (original) resolution (HR) counterparts. This association will then provide super resolution (SR) outputs when original point clouds are loaded as input. ZSPU achieves competitive/superior quantitative and qualitative performances on benchmark datasets when compared with other upsampling methods.

***Index Terms*—** Internal, holistic, zero-shot, upsampling

1. INTRODUCTION

In 3D computer vision domain, point cloud upsampling has recently achieved outstanding performance through supervised deep learning methods [1, 2, 3, 4]. However, these methods require tremendous amounts of training data to understand various structural shapes. While current 3D datasets for upsampling tasks from PU-Net [1], 3PU [3], PU-GAN [4], Dis-PU [5], and PU-GCN [6] covered a rich variety of objects, the absence of certain structures observed in real world scenarios (*e.g.*, trees, vehicles, animals, etc.) would limit the learning process. Particularly, although 3PU, PU-GAN, Dis-PU, and PU-GCN have proposed a patch-based scheme to improve the local uniformity of points for both training and testing phases, there still exists two potential limitations that restrict them from being applied to all sizes and structures of shapes. First, the parameters such as the number of points in each patch (NoPP) and the number of patches

---

* Corresponding Author. The source code is available at https://github.com/ky-zhou/ZSPU.
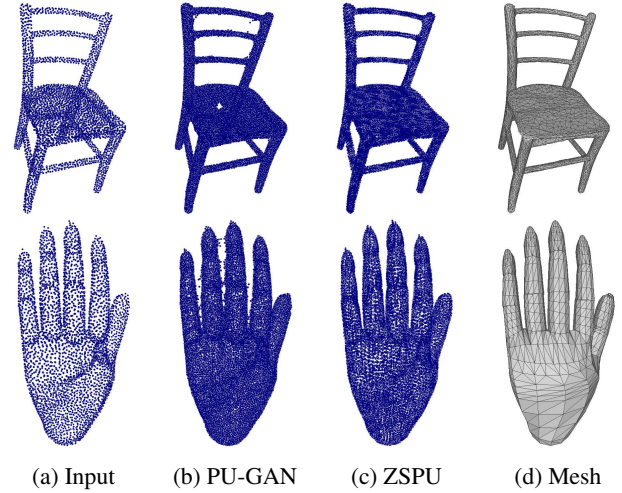

Fig. 1: (a) Input point cloud. (b) Patch based method sacrifices partial reconstruction capacity to generate more uniform distribution of points, while (c) our method tends to recover the original shape. (d) Ground truth mesh.

(NoP) need to be carefully set for various shapes to avoid producing problematic point clouds, (*e.g.*, in Fig. 1b (top), here the patches generated by farthest-point-sampling (FPS) algorithm are not sufficient to preserve the entire shape). Second, the nearby curvature might be disturbed by points in a patch with high geodesic distances *e.g.*, fingers, as shown in Fig. 1b (bottom). Finally, pre-processing such patches is not trivial, which requires to extract 3D coordinates (sometimes even the geodesic [7] or edge [2] information) from the existing mesh data.

In this paper, we present a "zero-shot" point cloud upsampling (ZSPU) framework to internally learn the self-aware representation of a holistic point cloud. This design skips the setup of parameters like NoPP and NoP to avoid potentially fallacious topological mappings and effectively preserves the shape of the target point cloud. We summarize our **contributions** as follows: (i) To the best of our knowledge, ZSPU is the first holistic and internal point cloud upsampling method. Our holistic design adapts the target shape complexity automatically and avoids the intricate settings and

patch pre-processing in patch-based methods. Additionally, the internal learning only requires one input point cloud, which significantly reduces the amount of training samples and time required by other supervised methods. (ii) Quantitatively, we achieve competitive performances on a benchmark dataset when compared with existing point cloud upsampling methods, and our method outperforms those methods on another benchmark dataset with unseen categories. (iii) Qualitatively, ZSPU preserves better local detail and curvature without losing much uniformity performance when compared with patch-based upsampling methods and thus is well suited to handle complex scenes obtained from real-world scans.

2. RELATED WORK

In this section, we briefly review the methods of point cloud upsampling and single image super resolution.

2.1. Point Cloud Upsampling

PointNet [8] and PointNet++ [9] have motivated multiple point cloud networks targeting upsampling in the past few years. Taking advantage of multi-scale grouping (MSG) and feature interpolation, PU-Net [1] first constructed a deep learning network that hierarchically upsamples the point clouds. Thereafter, EC-Net [2] took edge and surface information as prior knowledge, such that the upsampled point cloud forces the points to consolidate along the edges and surfaces. By leveraging geodesic distance as an additional feature, GeoNet [7] learned surface topology around each point. 3PU [3] first introduced the progressive trend with skip connections that constructs features from previous layers. DisPU [5] split the upsampling framework to a coarse point generator and a refiner that regressed offset to the points. AR-GCN [10], PU-GAN [4], and PU-GCN [6] applied adversarial learning to point cloud upsampling. PU-GCN designed a novel NodeShuffle module to rearrange the points and their neighbor features.

In general, the aforementioned external learning methods perform well to shapes with smooth surfaces or small curvatures. However, they are all patch-based approaches and suffer from the potential mismatch between the patching scheme and the target shape, especially for complex shapes with high curvatures. Additionally, they may require different hyperparameters (*e.g.*, NoPP and NoP) for different shapes.

2.2. Single Image Super Resolution

External learning is the most common manner in supervised learning trained with massive data. In contrast, internal learning has a pivotal role on problems that are restricted to relatively smaller data size.

Single image super resolution (SISR) aims to reconstruct an HR image from a degraded LR query. Most recent studies adopted internal examples [11, 12] for this task. Others utilized external information including zoomed camera image pairs [13] and texture transfer learning [14] to gain the extra knowledge for the network. A common characteristic among these studies is that they are not focusing on generating higher (super) resolution than the ground truth or the HR image.

"Zero-shot" super resolution (ZSSR) [15] was initially proposed to perform super resolution (SR) on a single image. This approach synchronously trains and tests the network with augmented pairs of LR-HR images extracted from the test image such that the synthesized image has a higher resolution than the HR image. By exploiting the internal recurrence of information within a single image and training an image-specific model at test time, ZSSR achieved competitive/superior results when compared with the external learning methods. Inspired by ZSSR, we use the term "zero-shot" in this paper to indicate holistic and internal upsampling of point clouds.

3. METHOD

Given a point set $\chi = \{x_i\}$, where $x_i \in \mathbb{R}^d$ and $i = 1, 2, ..., N$, we aim at generating a denser point set $\chi^r = \{x_j\}$, where r is the upsampling ratio and $j = 1, 2, ..., rN$. Here, we use $d = 3$, *i.e.*, xyz coordinate in Euclidean space. As the only available training data is χ, we apply either a random (non-ideal) or FPS (ideal) kernel to downsample χ to several LR versions, *i.e.*, $\chi^{LR_i}, i = 1, 2, ..., B$, where B is configured by the user during self-training. ZSPU provides an internal, holistic learning framework for point cloud upsampling. The key difference between ZSPU and existing external, patch-based methods is illustrated in Fig. 2, where a darker object color represents a higher resolution.

3.1. Architecture

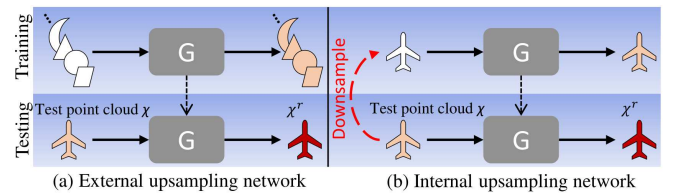(a) External upsampling network   (b) Internal upsampling network

Fig. 2: "Zero-Shot" point cloud upsampling. (a) External methods are first trained on a large number of patches. The test point cloud (whole or patched) χ is then fed into the network. (b) ZSPU is trained internally on the test point cloud itself. The resulting point cloud specific network is then applied on the original point cloud to produce a denser output.

ZSPU is a generative adversarial network (GAN) conditioning on a specific point cloud. More details of the architecture are provided in supplementary materials.
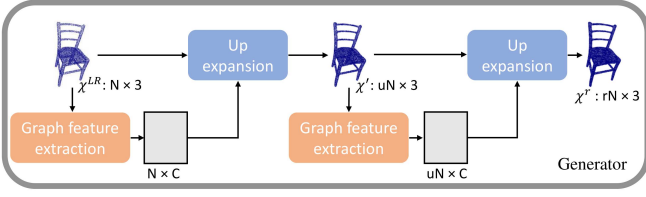
2

Fig. 3: The input point cloud has size of N. After progressive up expansions, the output χ^r has rN points. u and r are the progressive and overall upsampling ratios.

Generator. As shown in Fig. 3, the generator takes LR children χ^{LR_i} as input, and progressively upsamples with ratio 2 until the upsampling ratio is reached. This procedure repeats t times, where $t = log(r)$. This progressive learning approach supports changing the overall upsampling ratio r without modifying the network. We use χ^r to denote the output (and hereafter).

Discriminator. We adopt the module in [4] to distinguish the generated χ^r from the real target χ^{gt}. The global and point-wise features are first concatenated for better understanding the overall shape and details of the object. An attention module thereafter captures the point-wise similarities given the flattened features. Finally, a fidelity score is predicted via an aggregation of multilayer perceptron, max pooling, and dense layers.

3.2. Loss Function

Our joint loss function of the generator can be written as:

$$L = \alpha L_G + \beta L_{EMD}(\chi^r, \chi^{gt}) + \upsilon L_{uni} + \lambda L_{rep}(\chi^r) + \omega||\theta||^2, \quad (1)$$

in where, $||\theta||$ is the regularization term, and α, β, υ, λ, and ω are the weights, L_G, L_{EMD}, L_{uni}, and L_{rep} are the adversarial, reconstruction, uniform, and repulsion losses, respectively. The details of loss functions are described in the supplementary material.

3.3. Holistic vs. Patching

Unlike patch-based methods (*i.e.*, subdividing a point cloud into patches during testing and then merging upsampled patches later using FPS), our approach tends to learn and predict in a holistic style. In patch-based methods, the motivation is to map the testing patches to the similar pairs in the training set. However, the formation of patches restricts the accuracy of reconstruction of the entire shape, *i.e.*, a set formed by kNN of a centroid might have extremely large geodesic distance while having a small euclidean distance. This requires patch based methods to select parameters (*e.g.*, NoPP and NoP) carefully. In the case of a patch containing points from different parts of the object (*e.g.*, two fingers), the net-
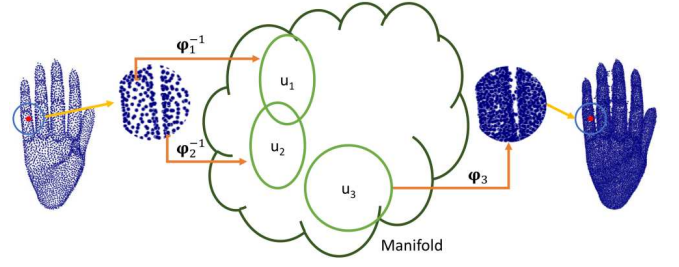

Fig. 4: In this example, the little finger and the ring finger are considered as a single shape since the manifold fails to map the two parts (u_1 and u_2) simultaneously. Instead, u_3 is selected as the corresponding subset in topology T_M, such that the mixture is occurred by $\varphi_3(u_3)$.

work might fail seeking for the correct subsets in the manifold for these parts.

We define M as manifold, T_M as the topology for M, and Λ as the transition functions between Euclidean and topological spaces. Specifically, in Λ, we denote φ^{-1} and φ as the transition functions into and out of the topological space, respectively. Given (M, T_M, Λ), this procedure can be formally written as

$$u \in T_M : \exists \varphi : u \to \varphi(u) \subset \mathbb{R}^d. \quad (2)$$

In Fig. 4, we illustrate an example when an input patch P_{in} is mapped into the topological space, its corresponding subsets $\{u_1, u_2\} = \varphi^{-1}(P_{in})$ cannot be accurately located since $\{u_1 + u_2\} \geqslant \{u_1 \cup u_2\}$. Alternatively, the network could select u_3 to represent $\{u_1 \cup u_2\}$ but $\varphi(\{u_1 \cup u_2\}) \neq \varphi(u_3)$. Empirically, this phenomenon can be alleviated by using more training samples. However, it is hard to cover all shapes in real-world applications.

In our holistic setting, P_{in} denotes the entire point cloud, such that $u_h = \varphi^{-1}(P_{in})$ can represent the topology T_M of P_{in}. The transition $\varphi(u_h)$ to Euclidean space is then directly conducted without any potential mapping problem in the topological space.

4. EXPERIMENTS

4.1. Datasets

We use the testing set (20 objects) in PU-GAN [4] and select 20 representative objects in Princeton Shape Benchmark [16] (denoted as $Data_{PU}$ and $Data_{PS}$, respectively) for our quantitative experiments. Specifically, due to $Data_{PU}$ being non-uniform, we generate 4,096 points (HR) using the meshes in $Data_{PU}$ through Poisson sampling algorithm. If not specified, we apply the non-ideal kernel to downsample 4,096 points (HR) to 1,024 (LR) through a random selection in all experiments. The ideal kernel uses FPS to downsam-

3

Methods	CD (10⁻³)	HD	P2F (10⁻³) μ	σ	Uniformity 0.4% (10⁻³)	Train Holistic	Test	Epoch	Time	# of param
PU-Net	0.38	3.67	8.19	6.65	6.36	✗	✔	120	4.5h	814k
AR-GCN	0.23	1.78	3.02	3.52	1.29	✗	✔	120	6.2h	822k
3PU	0.21	1.90	1.72	2.21	1.32	✗	✗	400	27h	304k
PU-GAN	0.17	1.76	1.05	1.92	0.55	✗	✗	100	25h	684k
Dis-PU	0.15	1.40	1.19	1.86	0.58	✗	✗	400	80h	1047k
PU-GCN	0.26	2.62	2.15	3.01	1.75	✗	✗	100	9h	542k
ZSPU	0.19	1.11	2.12	2.21	2.24	✔	✔	50	96s	310k
ZSPU-I	0.20	1.10	1.89	1.79	1.89	✔	✔	50	98s	310k

Table 1: Quantitative comparisons with supervised upsampling networks on $Data_{PU}$.

Methods	CD (10⁻³)	HD	P2F (10⁻³) μ	σ	Uniformity for different p (10⁻²) 0.4%	0.6%	0.8%	1.0%	1.2%
PU-Net	0.83	10.58	9.52	7.98	52.38	29.59	22.51	19.86	19.08
AR-GCN	0.42	4.75	2.73	3.22	51.20	22.17	13.89	11.18	10.57
3PU	0.44	3.04	0.94	1.32	47.37	16.44	8.30	6.06	5.89
PU-GAN	0.52	5.66	1.45	2.61	73.55	28.63	15.35	10.69	9.20
Dis-PU	0.47	4.44	1.45	2.47	80.53	32.32	17.69	12.36	10.52
PU-GCN	0.69	8.65	1.89	3.03	64.55	26.22	14.94	10.98	9.73
ZSPU	0.33	2.40	1.56	1.62	38.89	19.50	14.29	12.80	12.76
ZSPU-I	0.32	2.35	1.45	1.40	44.68	20.79	14.32	12.43	12.27

Table 2: Quantitative comparisons with supervised upsampling networks on $Data_{PS}$.

ple the point sets, which is denoted by ZSPU-I in our experiments. There are B pairs of LR and HR point clouds, which in our test is set as 12 for the purpose of performance and computational efficiency. In our experiments, the patched training data from other works are not taken into consideration.

4.2. Implementation Detail

To train ZSPU, we use Adam optimizer with learning rate 0.001 for the generator and 0.0001 for the discriminator, and batch size 12 for 50 epochs. Data augmentation is applied to all LR inputs, $i.e.$, rotation, jittering, shifting, and scaling. The weights in Eq. 1 are empirically set at 0.005, 1, 0.1, 0.01, and 0.01, respectively. Our network is implemented on TensorFlow 1.15 and trained by NVIDIA Titan RTX GPU.

4.3. Evaluation Metric

For the quantitative evaluation, we report the following widely-adopted metrics [4]: Chamfer distance (CD), Hausdorff distance (HD), point-to-surface (P2F) distance in average and its standard deviation, and uniformity on different radii in a unit area. To compare distances with ground truth, we uniformly sample 16,384 points from the ground truth mesh using the Poisson sampling algorithm. Lower values in the evaluation metrics represent better performances.

4.4. Quantitative Comparison

Although unfair, we quantitatively compare ZSPU with state-of-the-art external learning networks. We use their pre-trained models to test on $Data_{PU}$ and $Data_{PS}$. Here, we manually set up NoPP and NoP during testing to obtain visually complete point clouds for patch-based methods ($i.e.$, 3PU, PU-GAN, Dis-PU, and PU-GCN) as they significantly affect the reconstruction performance. Empirically, these hyperparameters fluctuate exceedingly with the shape complexity of the object or the number of objects in a target scene, making it nontrivial to apply patch-based methods straightly on all target categories. We highlight the best results with red color and the second best with blue color in all quantitative results.

Table 1 shows the quantitative comparisons on $Data_{PU}$. The epoch and time are reported by the corresponding articles without retraining on our end. In terms of HD, our ZSPU outperforms all other supervised methods. Results show that our method has competitive performance with supervised methods regarding CD, P2F, and uniformity metrics. Note that our training time is significantly less than the others, and our method avoids selecting hyper-parameters ($i.e.$, NoPP and NoP) for miscellaneous scenarios. Also in Table 1, we report the number of trainable parameters for the upsampling methods with 4096 points as the input. ZSPU generally has a smaller size than other models.

Table 2 lists the quantitative results with other supervised learning methods on unseen shape categories from $Data_{PS}$. Our approach achieves the best global and local results. Globally, the reconstruction is outperformed by our method by a large margin (21% in CD and HD). Locally, our method achieves the best local uniformity in the smaller disk (0.4%), indicating that ZSPU is compatible to numerous shapes.

4.5. Qualitative Comparison

In this section, we qualitatively compare our model with the top performers based on the quantitative results on mesh reconstructions, complex shapes, and real-world scenarios. More results are provided in the supplementary material.

Despite 3PU and PU-GAN achieving best performance regarding uniformity in quantitative results on $Data_{PU}$, through visual inspection, we find that they both sacrifice the smoothness of close surfaces in the patch-based prediction phase, which may cause severe mistakes in classification and remeshing tasks. Fig. 5 shows that the uniformity metric does not necessarily lead to better surface reconstruction, partly due to the interference between nearby patches. Furthermore, we show some upsampled examples in Fig. 6 for complex shapes in $Data_{PS}$. Our model produces significantly better results, particularly in fine detailed areas. These examples demonstrate that ZSPU is more suitable than patch-based upsampling methods on shapes with complex details or rapid changing curvatures.

External vs. Internal. To further illustrate the upsampling preferences of the external and internal methods on local regions of a point cloud, in Fig. 7 we provide the error
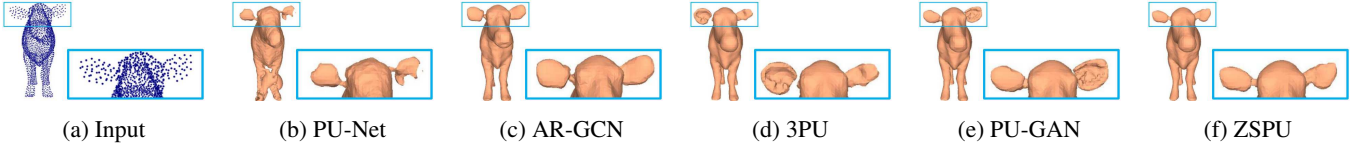
4

(a) Input   (b) PU-Net   (c) AR-GCN   (d) 3PU   (e) PU-GAN   (f) ZSPU

Fig. 5: Comparison of surface reconstruction results. Our method has better local reconstruction than the patch-based methods even for shapes with smooth surfaces.
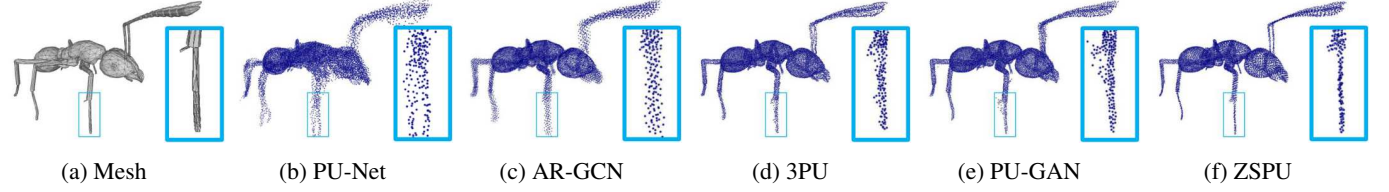

(a) Mesh   (b) PU-Net   (c) AR-GCN   (d) 3PU   (e) PU-GAN   (f) ZSPU

Fig. 6: Comparison of synthesis for complex shapes. Visually, more local details are preserved by our method.

(a) Input & GT   (b) PU-Net   (c) AR-GCN   (d) 3PU   (e) PU-GAN   (f) ZSPU
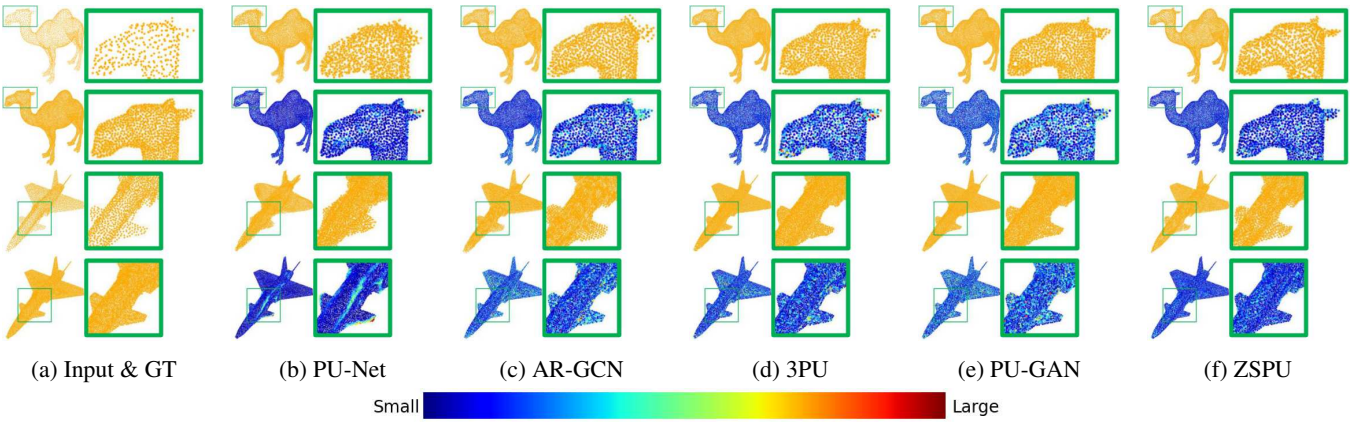
Small ▬▬▬▬▬▬▬ Large

Fig. 7: Comparison of error maps. The colors reveal the nearest distance from each target point to the generated point. ZSPU tends to preserve more local curvatures while maintaining competitive performance on smooth regions.

map comparison for two examples selected from $Data_{PU}$ and $Data_{PS}$, respectively. The colors in error maps reveal the nearest distance from each target point to the generated point. Visually, ZSPU generates the least error on local curvatures while maintaining competitive performance on smooth regions.

Upsampling Real-Scanned Data. PU-GAN achieved the best overall performance on the most common upsampling dataset $Data_{PU}$. In Fig. 8, we further compare our model with PU-GAN on LiDAR-scanned point clouds from the KITTI dataset [17], which contains mixed shapes of car, bike, scrub, and ground. NoP is carefully set up for PU-GAN to ensure that all points can be covered. Albeit PU-GAN fills the tiny gaps through the optimization of the uniformity metric, in practice it leads to the potential interference between different parts of an object (*e.g.*, the zoomed frames in Fig. 8). These examples clearly demonstrate the advantages of our "zero-shot" model over supervised patch-based methods in real-world scenarios.

5. CONCLUSION

In this paper, we present a "zero-shot" point cloud upsampling framework, which takes the entire point cloud as input and trains the network internally to produce a point cloud with a higher resolution. Through the holistic and internal learning design, our model avoids the intricate settings used in patch-based approaches and achieves competitive/superior results on benchmark datasets. It is well suited to handle high curvature regions or complex scenes obtained from real-world scans.

6. REFERENCES

[1] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng, "PU-Net: Point cloud up-
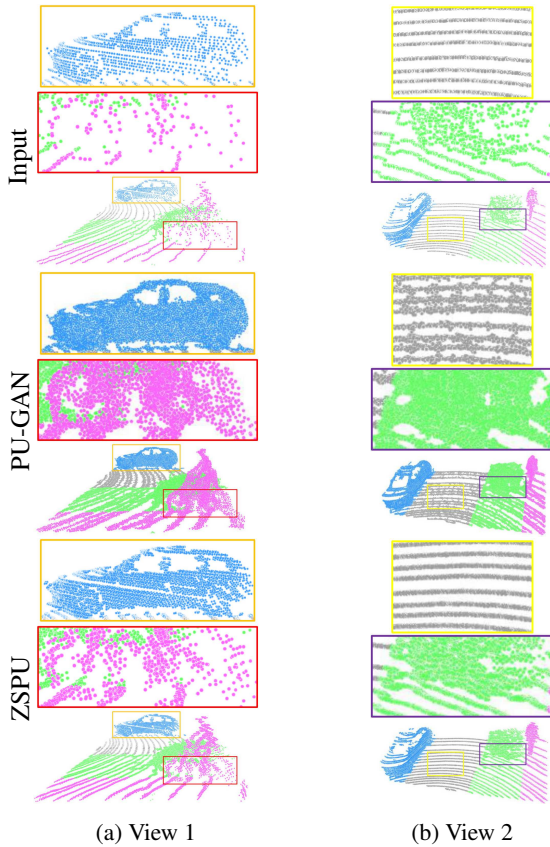
(a) View 1   (b) View 2

Fig. 8: Comparison Between PU-GAN and ZSPU using a real scene containing multiple objects.

sampling network," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2790–2799.

[2] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng, "EC-Net: an edge-aware point set consolidation network," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018, pp. 386–402.

[3] Wang Yifan, Shihao Wu, Hui Huang, Daniel Cohen-Or, and Olga Sorkine-Hornung, "Patch-based progressive 3d point set upsampling," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5958–5967.

[4] Ruihui Li, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng, "PU-GAN: a point cloud upsampling adversarial network," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2019, pp. 7203–7212.

[5] Ruihui Li, Xianzhi Li, Pheng-Ann Heng, and Chi-Wing Fu, "Point cloud upsampling via disentangled refinement," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 344–353.

[6] Guocheng Qian, Abdulellah Abualshour, Guohao Li,

Ali Thabet, and Bernard Ghanem, "PU-GCN: Point cloud upsampling using graph convolutional networks," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11683–11692.

[7] Tong He, Haibin Huang, Li Yi, Yuqian Zhou, Chihao Wu, Jue Wang, and Stefano Soatto, "GeoNet: Deep geodesic networks for point cloud analysis," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 6888–6897.

[8] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas, "PointNet: Deep learning on point sets for 3d classification and segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 652–660.

[9] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[10] Huikai Wu, Junge Zhang, and Kaiqi Huang, "Point cloud super resolution with adversarial residual graph networks," in *British Machine Vision Conf. (BMVC)*, 2020.

[11] Cheng Ma, Yongming Rao, Yean Cheng, Ce Chen, Jiwen Lu, and Jie Zhou, "Structure-preserving super resolution with gradient guidance," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7769–7778.

[12] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen, "Single image super-resolution via a holistic attention network," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020, pp. 191–207.

[13] Xuaner Zhang, Qifeng Chen, Ren Ng, and Vladlen Koltun, "Zoom to learn, learn to zoom," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3762–3770.

[14] Zhifei Zhang, Zhaowen Wang, Zhe Lin, and Hairong Qi, "Image super-resolution by neural texture transfer," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7982–7991.

[15] Assaf Shocher, Nadav Cohen, and Michal Irani, ""Zero-Shot" super-resolution using deep internal learning," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3118–3126.

[16] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser, "The Princeton shape benchmark," in *Proc. of Shape Modeling Applications*, 2004, pp. 167–178.

[17] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun, "Vision meets robotics: The KITTI dataset," *The Int. Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

6