

PU-EVA: An Edge-Vector based Approximation Solution for Flexible-scale Point Cloud Upsampling

Luqing Luo¹, Lulu Tang^{1*}, Wanyi Zhou², Shizheng Wang³, Zhi-Xin Yang^{1†}

¹State Key Laboratory of Internet of Things for Smart City, University of Macau

²South China University of Technology

³Institute of Microelectronics Chinese Academy of Sciences

{gabrielle_tse, lulutang}{outlook.com, 202021044254@mail.scut.edu.cn,

shizheng.wang@foxmail.com, zxyang@um.edu.mo

Abstract

High-quality point clouds have practical significance for point-based rendering, semantic understanding, and surface reconstruction. Upsampling sparse, noisy and non-uniform point clouds for a denser and more regular approximation of target objects is a desirable but challenging task. Most existing methods duplicate point features for upsampling, constraining the upsampling scales at a fixed rate. In this work, the arbitrary point clouds upsampling rates are achieved via edge-vector based affine combinations, and a novel design of Edge-Vector based Approximation for Flexible-scale Point clouds Upsampling (PU-EVA) is proposed. The edge-vector based approximation encodes neighboring connectivity via affine combinations based on edge vectors, and restricts the approximation error within a second-order term of Taylor’s Expansion. Moreover, the EVA upsampling decouples the upsampling scales with network architecture, achieving the arbitrary upsampling rates in one-time training. Qualitative and quantitative evaluations demonstrate that the proposed PU-EVA outperforms the state-of-the-arts in terms of proximity-to-surface, distribution uniformity, and geometric details preservation.

1. Introduction

With increasing capability of acquiring point clouds as a direct approximation of an object or scene surface from 3D scanning sensors, typical applications processing raw points are prevailing, such as point-based semantic understanding [3, 9, 24, 25, 27], point clouds rendering [4, 13] and surface reconstruction [19, 29], etc. Hence, the quality of input points is critical for the digital designs. However, a couple of varying artifacts, including occlusion, light

*Equal contributions as co-first author

†Corresponding author

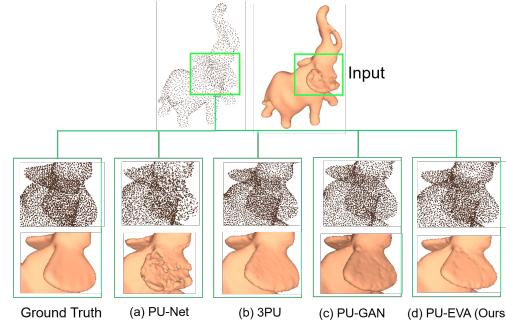


Figure 1. Point cloud upsampling results of an elephant by using (a) PU-Net [34], (b) 3PU [32], (c) PU-GAN [16] and (d) our PU-EVA. Note that our PU-EVA with edge-vector based approximation generates points more uniform with fine-grained details, compared to other state-of-the-arts.

reflection, surface materials, sensor resolutions and viewing angles, hinder high-quality practical acquisition of point clouds. It is desirable to upsample raw point clouds for a better description of an underlying surface, by converting the sparse, noisy, and non-uniform points into the dense, regular and uniform ones.

Intuitively, as a similar task in 2D computer vision, image super-resolution [15, 18, 23, 26, 31] is supposedly to lend well-developed techniques for point clouds upsampling. Nevertheless, the intrinsic irregularity of point clouds makes effective implementations for image interpolations not applicable in 3D world. Based on this, different criteria should also be considered for point clouds upsampling. First, the upsampled points should lie on or be close to the underlying surfaces of target objects. Next, the upsampled results should distribute uniformly, instead of being clustered together. Last, the generated points should preserve geometric details of the target objects.

The previous works often upsample points by simply du-

plicating point features and disentangling them in place via a set of MLPs [16, 32, 34]. The strategy may lead the generated points cluster around original ones and loss geometric details of the target objects. To overcome the issues, we are motivated to develop an upsampling strategy by formulating it as a problem of surface approximation based on edge-vector affine combinations. According to linear approximation theorem, the new points are upsampled in each small local neighborhood of the underlying surfaces. Here, edges as a concept inspired by EdgeConv [28], connect each center point to its neighboring points within the local neighborhood. The edge vectors encode feature of the oriented edges pointing from center points. The affine combinations interpolate new points based on the edge vectors via coefficients determined by a similarity matrix. Different from straightforward feature duplications for point upsampling, the edge-vector based approximation interpolates new points with endowing neighboring geometric information, and restricts the approximation error within a second-order term. To implement the motivation, we propose a novel design of Edge-Vector based Approximation for Flexible-scale Point clouds Upsampling (PU-EVA). The proposed PU-EVA decouples the upsampling scales with network architecture, making the upsampling rate flexible in one-time end-to-end training. To sum up, the contributions of proposed PU-EVA is listed as below.

1. The proposed PU-EVA achieves arbitrary upsampling scales via edge-vectors based affine combinations, making the upsampling rate decouple with network architectures;
2. The edge-vector based affine combinations upsample new points by endowing geometric information of the target objects, which benefits upsampling performance in sharp regions and areas with fine-grained details;
3. The proposed upsampling unit restricts the approximation error in a second-order term within each local neighborhood in light of linear approximation theorem.

2. Related work

2.1. Optimization-based upsampling methods

One of the earliest optimization-based methods generates a set of triangles for the sample points by using three-dimensional Voronoi diagram and Delaunay triangulation [2]. A parametric-free method of point resampling and surface approximation is developed by using a locally optimal projection operator (LOP) in [17]. However, the LOP-based methods require surface smoothness, giving rise to performance struggling around sharp edges and corners. To this end, an edge-aware point clouds resampling

(EAR) method uses implicit models to preserve sharp features, by resampling points away from edges and upsampling them progressively to approach the edge singularities [11], whereas the performance of EAR heavily depends on the given normal values and parameter tuning. For point clouds completion and consolidation, a new representation of associating surface points with inner points to reside on the extracted meso-skeletons is proposed, and the optimization is conducted under global geometry constraints from the meso-skeletons [30]. The method recovers regions with holes successfully though, it is sensitive to noise and outliers. Overall, the piece-wise smoothness assumption of optimization-based upsampling methods makes the fine-grained patterns missing, causing the performance of this category of methods limited.

2.2. Learning-based upsampling methods

Inspired by data-driven approaches and their promising results, taking advantages of deep learning techniques to model the complex geometries has been brought to attention thus far. PU-Net [34], as the first learning-based point upsampling network, learns geometry semantics of point patches based on the framework of PointNet++ [20], and expands the learned multi-scale features to upsample a given point cloud. EC-Net [33] utilizes an edge-aware technique to process point clouds and simultaneously recovers 3D point coordinates and point-to-edge distances by a joint loss. 3PU [32] proposes a progressive upsampling network using different number of neighbors in subsequent upsampling units. PU-GAN [16] is a generative adversarial network (GAN) [8] based point cloud upsampling network. It focuses on boosting the quantitative performance of upsampled point clouds by constructing an up-down-up expansion unit, a self-attention unit as well as a compound loss. PU-GCN [21] is built upon Graph Convolutional Networks (GCNs), and multiple designed upsampling modules can be incorporated into other upsampling pipelines. One of the main limitations of these learning-based upsampling methods is that the upsampling rate is fixed during each training, limiting their applications to the real-world upsampling tasks. Our proposed upsampling strategy decouples the upsampling rate with network architecture, with an edge-vector based approximation solution to generate new points by encoding neighboring connectivity, achieving arbitrary the upsampling rates in one-time training.

3. Theoretical formulation

Given an input sparse point cloud $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3 | (p_i^x, p_i^y, p_i^z)\}_{i=1}^N$, where N is the number of points. The objective is to generate more points $\mathcal{P}_R = \{\mathbf{p}_r \in \mathbb{R}^3 | (p_r^x, p_r^y, p_r^z)\}_{r=1}^{RN}$ approximating the

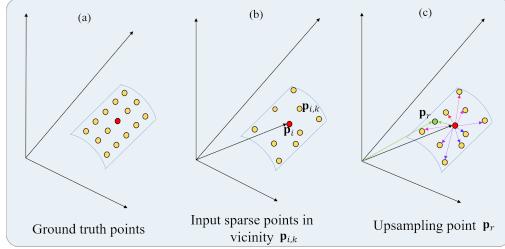


Figure 2. Edge-vector based approximation for point clouds upsampling. (a) Ground truth points. (b) Input sparse point in vicinity $\mathcal{N}_K(\mathbf{p}_i)$, centered at red point \mathbf{p}_i and surrounded by k nearest yellow neighbor points $\mathbf{p}_{i,k}$. (c) Points upsampling via edge-vector based approximation. Arrows with different colors represent the edge-vectors with different significance to the green upsampled point \mathbf{p}_r .

underlying surface, where R is the upsampling rate¹.

As shown in Figure 2, consider a small vicinity $\mathcal{N}_K(\mathbf{p}_i) \subseteq \mathcal{P}$ centered at \mathbf{p}_i (red point) and enclosed by K nearest neighbors $\mathbf{p}_{i,k}$ (yellow points), we assume that $\mathcal{N}_K(\mathbf{p}_i)$ is local smoothness at point \mathbf{p}_i . According to the implicit function theorem [10], the vicinity satisfies $f(\mathbf{p}_{i,k}) = 0, \forall \mathbf{p}_{i,k} \in \mathcal{N}_K(\mathbf{p}_i)$, where $f(\cdot)$ is a smooth implicit function. If the partial derivative $\frac{\partial f}{\partial p^z}$ exists, the local surface can be expressed explicitly by a height function $F : \mathbb{R}^2 \rightarrow \mathbb{R} : p^z = F(p^x, p^y)$.

We start from constructing p_r^x and p_r^y of the upsampling points in local region, which is,

$$(p_r^x, p_r^y) = \sum_{k=1}^K w_k (p_{i,k}^x, p_{i,k}^y), \quad (1)$$

with non-negative coefficients w_k . Theoretically, given p_r^x and p_r^y , p_r^z can be obtained via the height function $F(\cdot)$. However, the analytical solution of a continuous function is too computational expensive to acquire. The corresponding numerical solution can be approximated via Taylor's expansion at point \mathbf{p}_i as, $p^z = F(p_i^x, p_i^y) + \nabla F(p_i^x, p_i^y)^\top \cdot (p^x - p_i^x, p^y - p_i^y) + O((p^x - p_i^x, p^y - p_i^y)^2) = F_a(p^x, p^y) + O((p^x - p_i^x, p^y - p_i^y)^2)$. A linear function $F_a(\cdot)$ can be defined as $F_a(p^x, p^y) \triangleq F(p_i^x, p_i^y) + \nabla F(p_i^x, p_i^y)^\top \cdot (p^x - p_i^x, p^y - p_i^y)$. By omitting the second-order error term in Taylor's expansion, p_r^z on the surface can be approximated as,

$$\begin{aligned} p_r^z &\approx F_a(p_r^x, p_r^y) = F_a \left(\sum_{k=1}^K w_k (p_{i,k}^x, p_{i,k}^y) \right) \\ &= \sum_{k=1}^K w_k F_a(p_{i,k}^x, p_{i,k}^y) \approx \sum_{k=1}^K w_k p_{i,k}^z. \end{aligned} \quad (2)$$

¹In our formulation, cursive letter \mathcal{P} represents point clouds, and bold letter \mathbf{p} represents point coordinates.

To this end, \mathcal{P}_R is approximated via affine combinations in each $\mathcal{N}_K(\mathbf{p}_i)$ with avoiding gradient calculations:

$$\mathbf{p}_r = \sum_{k=1}^K w_k \mathbf{p}_{i,k}, \quad \text{s.t. } \sum_{k=1}^K w_k = 1, \forall w_k > 0. \quad (3)$$

The coefficients of the affine combination w_k are obtained by a similarity matrix S based on K neighboring points and their R randomly selected anchor points,

$$w_k = f_{\text{softmax}}(g \cdot h^\top), \quad (4)$$

where f_{softmax} denotes softmax function, $g = \{\phi_1(\Delta \mathcal{F}_{i,r}) | g \in \mathbb{R}^{N \times R \times C_1}\}$ and $h = \{\phi_2(\Delta \mathcal{F}_{i,k}) | h \in \mathbb{R}^{N \times K \times C_1}\}$ are high dimensional features of edge vectors mapped by convolutional operators $\phi_1(\cdot)$ and $\phi_2(\cdot)$ in dimension channel C_1 . The edge vectors $\Delta \mathcal{F}_{i,r}$ and $\Delta \mathcal{F}_{i,k}$ are features of oriented edges pointing to K neighboring points and R anchor points, respectively. The similarity matrix S consists of the significance of each neighboring point to the upsampled point, which are showed by different colors of arrows in Figure 2 (c). The weighted sum of neighboring connectivities is then used to interpolate new points, as shown in green point of Figure 2 (c).

4. Proposed method

Motivated by the theoretical formulation in Section 3, a novel data-driven framework for point clouds upsampling in an end-to-end fashion is proposed, dubbed edge-vector based approximation for flexible-scale point clouds upsampling (PU-EVA), which consists three elements: dense feature extraction, edge-vector based approximation upsampling, and coordinate reconstruction, as shown in Figure 3.

4.1. Dense feature extraction

In feature extraction unit, the structure-aware features $V = \{v_i\}_{i=1}^N$ of $N \times C$ are extracted from input sparse point cloud \mathcal{P} . A dense dynamic feature extraction method in [32] is used, where EdgeConv [28] with skip-connections are adopted as basic blocks. The local neighborhood of feature space is defined and assembled by KNN in terms of feature similarity. A series of densely connected MLPs and max-pool are then utilized to update the non-local order invariant point features dynamically. To leverage features extracted from different layers, skip-connections are applied both within and between those dense blocks. Within a dense block, the output of each MLP is passed to all subsequent MLPs; and between the dense blocks, each block's output is fed as input to the following blocks. This mechanism of skip-connections takes full advantage of information across different levels, improving the learned geometric features from multiple scales.

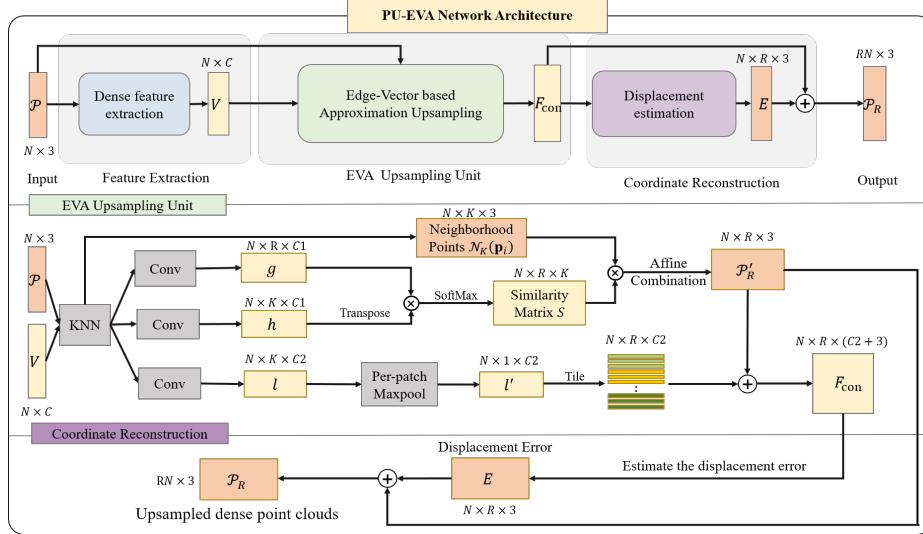


Figure 3. Architecture of PU-EVA network, consisting of three components: feature extraction, points upsampling and coordinate reconstruction. It takes spare input (N points) and produces denser point clouds (RN points) by a carefully designed edge-vector based approximation, where R is the upsampling rate. The Chamfer Distance (CD loss) and the uniform loss are adopted jointly in the end-to-end training.

4.2. Edge-vector based approximation upsampling

The core of point clouds upsampling is the upsampling unit, which normally is treated as feature expansion. Different from the straightforward feature duplication, the proposed Edge-Vector based Approximation (EVA) upsampling unit interpolates new points by encoding geometric information around local neighborhood, as shown in the middle part Figure 3. Specifically, the edge vectors $\Delta\mathcal{F}_{i,k}$ based on K neighboring points are computed as,

$$\begin{aligned} \Delta\mathcal{F}_{i,k} = & (\mathbf{v}_{i,k} - \mathbf{v}_i) \oplus \mathbf{v}_{i,k} \oplus \mathbf{v}_i \\ & \oplus (\mathbf{p}_{i,k} - \mathbf{p}_i) \oplus \mathbf{p}_{i,k} \oplus \mathbf{p}_i \oplus d_{i,k}, \end{aligned} \quad (5)$$

where \mathbf{p}_i and \mathbf{v}_i are point coordinates and point features extracted from dense feature extraction (see Section 4.1), and $d_{i,k}$ are Euclidean distances between center point and the surrounding points. Similarly, the edge vectors $\Delta\mathcal{F}_{i,r}$ are obtained based on R anchor points. The R anchor points are randomly selected from K neighboring points in each iteration, the correlations between them are calculated as elements of similarity matrix S . Followed by the 1×1 convolutional operators ϕ and ψ , the edge vectors are mapped into high dimensional feature space as $h = \{\phi_2(\Delta\mathcal{F}_{i,k}) | h \in \mathbb{R}^{N \times K \times C_1}\}$, $g = \{\phi_1(\Delta\mathcal{F}_{i,r}) | g \in \mathbb{R}^{N \times R \times C_1}\}$, and $l = \{\psi(\Delta\mathcal{F}_{i,k}) | l \in \mathbb{R}^{N \times K \times C_2}\}$. By obtaining coefficients of similarity matrix S based on g and h , the affine combinations around the neighborhood points $\mathcal{N}_K(\mathbf{p}_i)$ are implemented to determine new points. Furthermore, the local feature l introduces displacement error of neighborhood and tiles up to encourage new points describing sharp edges

or tiny structures of the latent target objects better. Since R anchor points are randomly selected in each iteration, making all possible combinations of neighboring connectivities are captured during training, the generated points can be R times of original one in inference stage. In this way, the dimension of similarity matrix $N \times R \times K$ for affine combinations of feature expansion decouples with the network architecture, the upsampling rate R can be set as an arbitrary positive integer smaller than K .

4.3. Coordinate reconstruction

The expanded features in (Sec. 4.2) are then reconstructed to output dense point clouds \mathcal{P}_R , see bottom of Figure 3. In our implementation, we adopt regression of displacement error E by the learned neighborhood feature. The concatenated feature F_{con} from EVA upsampling unit contains full neighborhood information. It is then fed as input of coordinate reconstructor to estimate displacement error E via a chain of MLPs. Here, E can be considered as the second-order error term of Taylor's expansion, whose theoretical explanation can be found in section 3. The final coordinates of output points are then computed by adding the generated new points \mathcal{P}'_R to the estimated error E , resulting in the desired dense point clouds \mathcal{P}_R .

5. Experiments

In this section, several experiments are conducted to compare our method with state-of-the-art point upsampling methods quantitatively and qualitatively, and various as-

Methods (10^{-3})	Sparse (256) input			Medium (2,048) input			Dense (4,096) input		
	CD	HD	P2F	CD	HD	P2F	CD	HD	P2F
EAR [12]	-	-	-	0.520	7.370	5.820	-	-	-
PU-Net [34]	2.461	15.370	13.099	0.720	8.940	6.840	0.247	2.802	12.033
3PU [32]	2.177	12.672	10.328	0.490	6.110	3.960	0.446	4.225	4.281
PU-GAN [16]	2.072	16.592	8.055	0.280	4.640	2.330	0.131	1.284	1.687
Ours	1.784	13.939	8.727	0.266	3.070	2.362	0.123	1.394	1.416

Table 1. Quantitative comparison of proximity-to-surface with various input point resolutions.

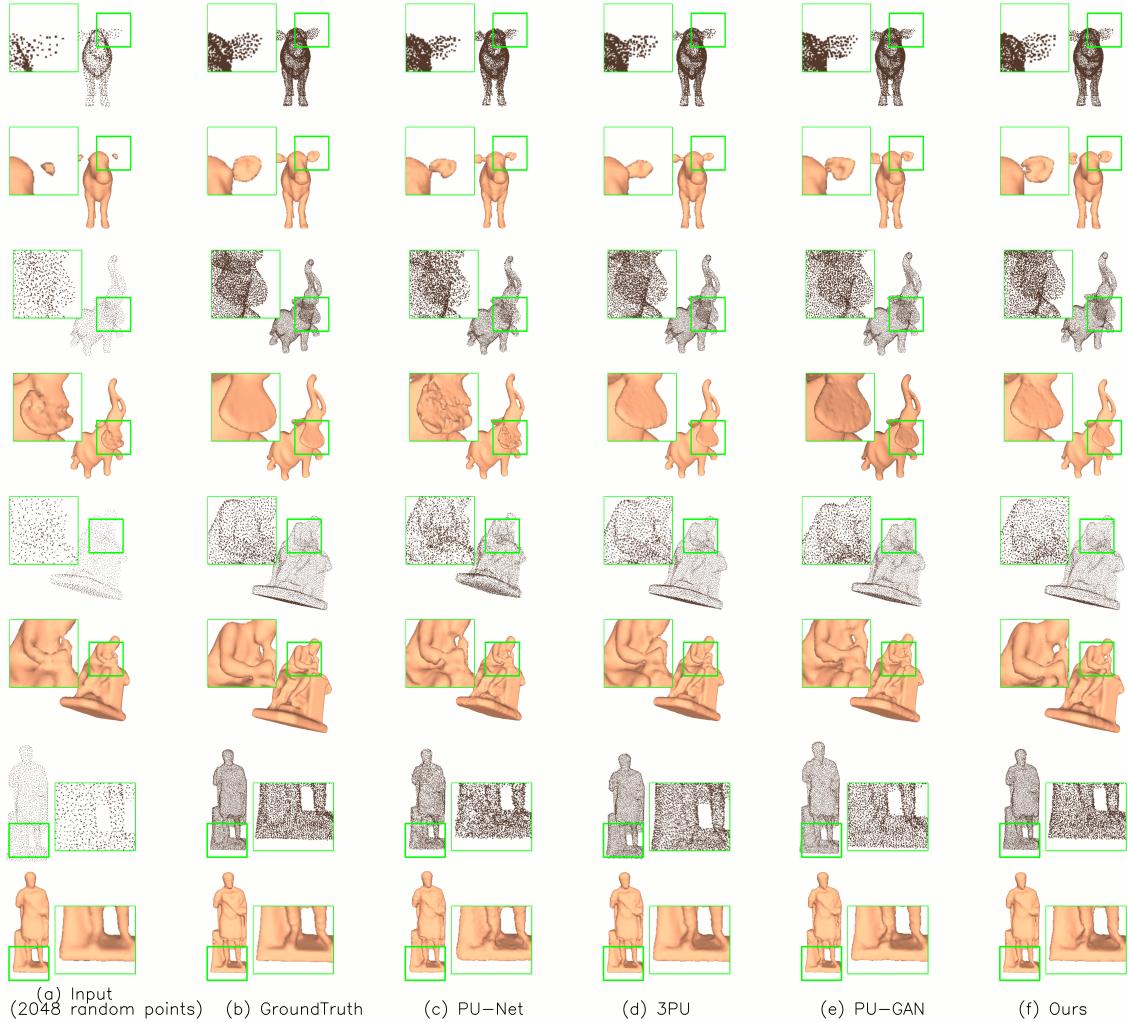


Figure 4. Comparison of point clouds upsampling ($4\times$) and surface reconstruction with state-of-the-art methods (c-f) from inputs (a).

pects of our model are evaluated.

5.1. Experimental setup

We train all these upsampling methods based on a benchmark dataset mentioned in [16] for fair comparison, in which 147 objects covering a rich variety of shapes are collected. The same 120 objects are used as training data, and the remaining are used for testing by following [16]. For

quantitative comparison, the referenced ground truth point distribution \mathcal{Q} of 8,192 points on the patch is sampled from original meshes by Poisson disk sampling [6]. During the training, the input spare point clouds \mathcal{P}_i is selected randomly from \mathcal{Q} on-the-fly. In testing, $6\times$ points are randomly selected from ground truth first, then $4\times$ points are further sampled by Farthest Point Sampling (FPS) [7]. K and R are set as 12 and 6 respectively to obtain the neigh-

Methods (10^{-3})	Uniformity for different p				
	0.4%	0.6%	0.8%	1.0%	1.2%
EAR [12]	16.84	20.27	23.98	26.15	29.18
PU-Net [34]	29.74	31.33	33.86	36.94	40.43
3PU [32]	7.51	7.41	8.35	9.62	11.13
PU-GAN [16]	3.38	3.49	3.44	3.91	4.64
Ours	2.26	2.10	2.51	3.16	3.94

Table 2. Quantitative comparison of uniformity for different p .

borhood vicinity $\mathcal{N}_K(\mathbf{p}_i)$ and the anchor points. Data augmentation techniques are also applied to rotate, shift and scale data randomly to avoid over-fitting. Throughout the experiments, we train PU-EVA for 120 epochs with a mini batch size 32 on GPU NVIDIA 2080Ti. All models converge before the maximum epochs. The models from the last epoch are used to evaluate the performance.

5.2. Loss function and evaluation metrics

5.2.1 Loss function

CD loss in [1] measures the similarity between upsampled point clouds $\mathcal{P}_r \subseteq \mathbb{R}^3$ and the ground truth $\mathcal{Q} \subseteq \mathbb{R}^3$, which encourages the upsampled points lie on or be closer to the underlying surface: $\mathcal{L}_{\text{CD}}(\mathcal{P}_r, \mathcal{Q}) = \frac{1}{|\mathcal{P}_r|} \sum_{p \in \mathcal{P}_r} \min_{q \in \mathcal{Q}} \|p - q\|_2^2 + \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \min_{p \in \mathcal{P}_r} \|q - p\|_2^2$.

Uniform loss adopted in [16] are defined as $U_{\text{imbalance}}(S_j) = \frac{(|S_j| - \hat{n})^2}{\hat{n}}$, and $U_{\text{clutter}}(S_j) = \sum_{k=1}^{|S_j|} \frac{(d_{j,k} - \hat{d})^2}{\hat{d}}$, where $\{S_j | j = 1 \dots M\}$ is acquired by cropping with a ball query of radius r_d around a seed point on a patch of \mathcal{P}_r , \hat{n} represents the expected number of points in S_j , $d_{j,k}$ indicates the distance between each point in S_j and its nearest neighbor $\mathbf{p}_{j,k}$ and \hat{d} is the expected point-to-neighbor distance. The uniform loss is then formulated as $\mathcal{L}_{\text{uni}} = \sum_{j=1}^M U_{\text{imbalance}}(S_j) \cdot U_{\text{clutter}}(S_j)$.

Overall, our PU-EVA network is trained end-to-end by minimizing the joint loss function:

$$\mathcal{L}(\theta) = \alpha \mathcal{L}_{\text{CD}} + \beta \mathcal{L}_{\text{uni}} + \gamma \|\theta\|^2. \quad (6)$$

where θ denotes the parameters in our PU-EVA network, α and β balance the CD loss and the uniform loss, and γ indicates the multiplier of weight decay.

5.2.2 Evaluation metrics

Four evaluation metrics are employed for quantitative comparison of proximity-to-surface and distribution uniformity. To measure the deviations between upsampled points and ground truth, three commonly used metrics Chamfer distance (CD), Hausdorff distance (HD) [5, 22], and point-to-surface distance (P2F) are adopted to compute the distances between an upsampled point \mathbf{p}_r and its closet points from

Methods (10^{-3})	$\sigma = 0$		$\sigma = 0.01$		$\sigma = 0.02$	
	CD	HD	CD	HD	CD	HD
PU-GAN [16]	0.280	4.640	0.512	6.496	0.912	11.326
Ours	0.266	3.070	0.464	5.501	0.864	9.178

Table 3. Quantitative comparison of upsampling results with different additive Gaussian noise levels.

ground truth. To compare the uniformity, \mathcal{L}_{uni} is utilized with α , β and γ set as 150, 10 and 1 empirically. According to $r_d = \sqrt{p}$, p is set as 0.4%, 0.6%, 0.8%, 1.0%, and 1.2%, respectively. In the implementation, M is chosen as 1,000, and the actual mesh of testing model is used to find S_j geodesically instead of cropping S_j by ball query. All these metrics are compared over the whole point cloud, and the smaller values indicate the better upsampling quality.

5.3. Quantitative and qualitative comparison

5.3.1 Quantitative results

Table 1 summarizes quantitative comparison of the proposed PU-EVA and other state-of-the-arts, i.e., EAR [12], PU-Net [32], 3PU [32] and PU-GAN [16], in terms of proximity-to-surface. The comparison is based on $\times 4$ upsampling rate, various input resolutions are evaluated, containing sparse points (256), medium points (2,048), and dense points (4,096), respectively. PU-EVA achieves the best performance with the lowest deviations from surface consistently according to all three metrics across the input resolutions, which means points generated from our method are closer to the ground truth. Besides, the upsampling rates of PU-Net [32] and PU-GAN [16] are tangled with the network architectures, with features expanded by replication and rearrangement, neglecting the complex geometric information contained in the latent object surface. Although 3PU [32] shows a competitive capability to deal with flexible upsampling rates, the training process is complicated and more subsets are required for a higher upsampling rate, which is still should be in powers of 2.

Table 2 reports quantitative comparison of points distribution uniformity for different p . We compare with optimization based EAR [12] and three deep learning based methods, i.e., PU-Net [32], 3PU [32] and PU-GAN [16]. The uniformity of our results stays the lowest for all different p , indicating that PU-EVA obtains the best distribution uniformity compared to state-of-the-arts over varying scales. Furthermore, the performance of EAR [12] heavily depends on normal values and parameter tuning.

5.3.2 Qualitative results

The visual comparison of point clouds upsampling and surface reconstruction (using [14]) is shown in Figure 4, in which (a) is 2,048 random input points, (b) is the ground

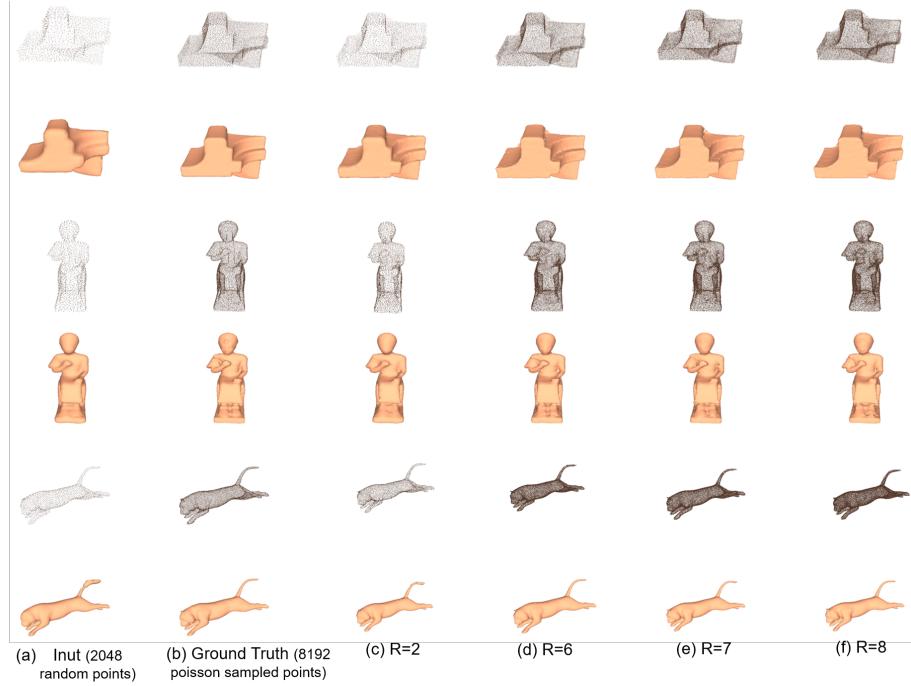


Figure 5. Point clouds upsampling of PU-EVA with various upsampling rate.

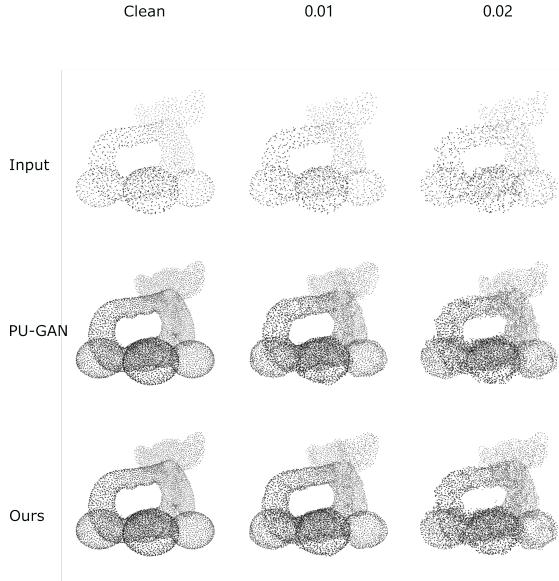


Figure 6. Qualitative comparison of upsampling results with different additive Gaussian noise levels: 0, 0.01 and 0.02 from left to right, respectively.

truth points uniformly-sampled from the original meshes of testing models, (c-f) are state-of-the-art methods PU-Net [32], 3PU [32] and PU-GAN [16], and (d) is our PU-EVA. The expanded features of PU-Net [32] are too simi-

lar to the inputs, which affects the upsampling quality, as shown in Figure 4 (c). Comparing with (c-e), our PU-EVA produces more uniform point clouds with less noise. According to the blown-up views of upsampled results, PU-EVA renders fine-grained details better, *e.g.* ox’s ear (top) and statue’s leg (bottom). The reconstructed surfaces are smoother with less wrinkles or bulges while maintain the complex structures of the original shape. To sum up, the proposed PU-EVA generates points with less noise and containing more details.

Figure 5 illustrates the upsampling results of the proposed PU-EVA with various upsampling rate, in which (a) is 2,048 random input points, (b) is 8,192 Poisson-sampled ground truth points, (c-f) are upsampling results in four rates, $R=2$, $R=6$, $R=7$ and $R=8$. Note that all the upsampling results are obtained in one-time training with $R=4$, and the upsampling results of rates bigger than the training rate still create point clouds fidelity to ground truth. Our PU-EVA interpolates new points based on edge-vectors to decouple the network architecture with feature expansion, making it flexible to achieve arbitrary upsampling rates. More upsampling results of real-scanned data and some failure cases are given in the supplementary material.

5.4. Robustness test

Robustness against varying levels of input noise. We evaluate the robustness of proposed PU-EVA by comparing with the state-of-the-art method, PU-GAN [16], against

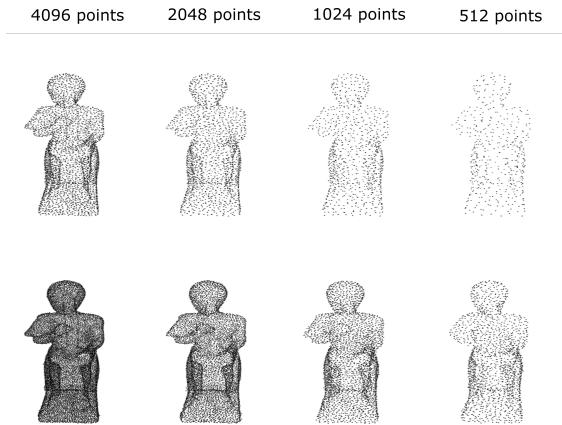


Figure 7. Upsampling results with varying sizes of input.

varying levels of input noise. Figure 6 and Table 3 demonstrate the comparison of upsampling results with additive Gaussian noises, the different additive noise levels of input point clouds are $\sigma = 0, 0.01, 0.02$, from left to right, respectively. Compared with PU-GAN [16] upsampled point clouds, the proposed PU-EVA generates points approximating surfaces better, especially in the cases with heavier noise contamination. Furthermore, with the additive Gaussian noise levels increasing, the proposed PU-EVA shows a stable performance, indicating robustness of the upsampling method. **Robustness against varying sizes of input.** Figure 7 illustrates the upsampling results with respect to varying sizes of the input point clouds. We sample four sets of test samples, which are 4,096, 2,048, 1,024, and 512 points, respectively. The bottom row of Figure 7 shows the corresponding $\times 4$ upsampled outputs. We can witness that even for the sparsest input points in the rightmost column, our PU-EVA acquires competitive upsampling results.

5.5. Ablation study

To justify the approximation accuracy of the second-order error term, the performances of PU-EVA with and without it are compared. The comparison is based on $\times 4$ upsampling rate, various input resolutions are evaluated, containing sparse points (256), medium points (2,048), and dense points (4,096), respectively. The quantitative results are shown in Table 4. The upsampling results of PU-EVA with the second-order error term declines both CD and HD distances, indicating its ability to drive upsampling points approaching the underlying surface.

To explore the efficacy of proposed EVA upsampling unit, we conduct ablation study to compare with different upsampling units. Specifically, direct point feature duplication upsampling, and the best upsampling unit in PUGCN [21], named NodeShuffle, are adopted for comparison. The direct point feature duplication follows the net-

2nd-order Error Term (10^{-3})	Sparse (256)		Medium (2,048)		Dense (4,096)	
	CD	HD	CD	HD	CD	HD
without	2.318	18.713	0.312	4.622	0.157	5.535
with	1.784	13.939	0.266	3.070	0.123	1.394

Table 4. Quantitative comparison for PU-EVA with and without the second-order error term from sparse to dense input point resolutions.

Upsampling Unit (10^{-3})	Sparse (256) input		Medium (2,048) input		Dense (4,096) input	
	CD	HD	CD	HD	CD	HD
Feature Duplication	2.94	24.67	0.38	3.99	0.13	4.47
NodeShuffle [21]	1.86	15.56	0.27	4.66	0.12	3.11
EVA Upsampling	1.78	13.93	0.26	3.07	0.12	1.39

Table 5. Quantitative comparison of different upsampling units from sparse to dense input point resolutions.

work architecture of Feature Expansion module in PU-Net [34], while keeps rest of network the same as PU-EVA. Table 5 summarizes quantitative comparison in terms of surface deviations. According to all three metrics, the proposed EVA upsampling unit achieves the best performance with the lowest deviation from surface consistently across the input point resolutions, convincing our upsampling unit encourages points to approximate the underlying surface better. The qualitative comparison is given in the supplementary material.

6. Conclusion

This work is motivated to approximate underlying surface for point clouds upsampling. To this end, we first analyze the linear approximation theorem by Taylor’s expansion, and identify edge-vector based approximation as an important controlling factor that determines the quality of upsampled results. Based on the analysis, we have proposed a novel network design of PU-EVA. Qualitative and quantitative evaluations demonstrate that the proposed PU-EVA outperforms the state-of-the-arts by generating smoother and more uniform dense point clouds maintaining more fine-grained details. In future research, we are interested in acquiring better upsampling results adaptive to the underlying geometric properties.

Acknowledgments. This work was funded in part by the Science and Technology Development Fund, Macau SAR (Grant No. 0018/2019/AKP, 0008/2019/AGJ, and SKL-IOTSC-2021-2023), the University of Macau (Grant No. MYRG2018-00248-FST and MYRG2019-0137-FST), the Beijing Natural Science Foundation (Grant No. 4194095), the National Natural Science Foundation of China (Grant No. 61802390), and the Hundred Talents Program of Chinese Academy of Sciences.

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. PMLR, 2018.
- [2] Nina Amenta, Marshall Bern, and Manolis Kamvysselis. A new voronoi-based surface reconstruction algorithm. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 415–421, 1998.
- [3] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016.
- [4] Rowel Atienza. A conditional generative adversarial network for rendering point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 10–17, 2019.
- [5] T Birsan and Dan Tiba. One hundred years since the introduction of the set distance by dimitrie pompeiu. In *IFIP Conference on System Modeling and Optimization*, pages 35–39. Springer, 2005.
- [6] Massimiliano Corsini, Paolo Cignoni, and Roberto Scopigno. Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE transactions on visualization and computer graphics*, 18(6):914–924, 2012.
- [7] Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing*, 6(9):1305–1315, 1997.
- [8] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.
- [9] Tong He, Haibin Huang, Li Yi, Yuqian Zhou, Chihao Wu, Jue Wang, and Stefano Soatto. Geonet: Deep geodesic networks for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6888–6897, 2019.
- [10] Francis B Hildebrand. Advanced calculus for applications. 1962.
- [11] Hui Huang, Dan Li, Hao Zhang, Uri Ascher, and Daniel Cohen-Or. Consolidation of unorganized point clouds for surface reconstruction. *ACM transactions on graphics (TOG)*, 28(5):1–7, 2009.
- [12] Hui Huang, Shihao Wu, Minglun Gong, Daniel Cohen-Or, Uri Ascher, and Hao Zhang. Edge-aware point set resampling. *ACM transactions on graphics (TOG)*, 32(1):1–12, 2013.
- [13] Alireza Javaheri, Catarina Brites, Fernando Manuel Bernardo Pereira, and Joao M Ascenso. Point cloud rendering after coding: Impacts on subjective and objective quality. *IEEE Transactions on Multimedia*, 2020.
- [14] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013.
- [15] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.
- [16] Ruihui Li, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-gan: a point cloud upsampling adversarial network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7203–7212, 2019.
- [17] Yaron Lipman, Daniel Cohen-Or, David Levin, and Hillel Tal-Ezer. Parameterization-free projection for geometry reconstruction. *ACM Transactions on Graphics (TOG)*, 26(3):22–es, 2007.
- [18] Peyman Milanfar. *Super-resolution imaging*. CRC press, 2017.
- [19] Christian Mostegel, Rudolf Prettenthaler, Friedrich Fraundorfer, and Horst Bischof. Scalable surface reconstruction from point clouds with extreme scale and density diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 904–913, 2017.
- [20] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.
- [21] Guocheng Qian, Abdulellah Abualshour, Guohao Li, Ali Thabet, and Bernard Ghanem. Pu-gcn: Point cloud upsampling using graph convolutional networks. *arXiv preprint arXiv:1912.03264*, 2019.
- [22] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.
- [23] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.
- [24] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.
- [25] Lulu Tang, Ke Chen, Chaozheng Wu, Yu Hong, Kui Jia, and Zhi-Xin Yang. Improving semantic analysis on point clouds via auxiliary supervision of local geometric priors. *IEEE Transactions on Cybernetics*, 2020.
- [26] JD Van Ouwerkerk. Image super-resolution survey. *Image and vision Computing*, 24(10):1039–1052, 2006.
- [27] Kaiqi Wang, Ke Chen, and Kui Jia. Deep cascade generation on point sets. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3726–3732, 2019.
- [28] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

- [29] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10130–10139, 2019.
- [30] Shihao Wu, Hui Huang, Minglun Gong, Matthias Zwicker, and Daniel Cohen-Or. Deep points consolidation. *ACM Transactions on Graphics (ToG)*, 34(6):1–13, 2015.
- [31] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.
- [32] Wang Yifan, Shihao Wu, Hui Huang, Daniel Cohen-Or, and Olga Sorkine-Hornung. Patch-based progressive 3d point set upsampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5958–5967, 2019.
- [33] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Ec-net: an edge-aware point set consolidation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 386–402, 2018.
- [34] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2799, 2018.