PointLIE: Locally Invertible Embedding for Point Cloud Sampling and Recovery

Weibing Zhao[1,2†], Xu Yan[1,2†], Jiantao Gao[2,3], Ruimao Zhang[1,2], Jiayan Zhang[1], Zhen Li[1,2] *, Song Wu[4], Shuguang Cui[1,2]

[1] The Chinese University of Hong Kong, Shenzhen
[2] Shenzhen Research Institute of Big Data
[3] Shanghai University
[4] Shenzhen Luohu Hospital
{weibingzhao@link., xuyan1@link., lizhen@}cuhk.edu.cn,

Abstract

Point Cloud Sampling and Recovery (PCSR) is critical for massive real-time point cloud collection and processing since raw data usually requires large storage and computation. This paper addresses a fundamental problem in PCSR: How to downsample the dense point cloud with arbitrary scales while preserving the local topology of discarded points in a case-agnostic manner (*i.e.,* without additional storage for point relationships)? We propose a novel Locally Invertible Embedding (**PointLIE**) framework to unify the point cloud sampling and upsampling into one single framework through bi-directional learning. Specifically, PointLIE decouples the local geometric relationships between discarded points from the sampled points by progressively encoding the neighboring offsets to a latent variable. Once the latent variable is forced to obey a pre-defined distribution in the forward sampling path, the recovery can be achieved effectively through inverse operations. Taking the recover-pleasing sampled points and a latent embedding randomly drawn from the specified distribution as inputs, PointLIE can theoretically guarantee the fidelity of reconstruction and outperform state-of-the-arts quantitatively and qualitatively.

1 Introduction

Recently, as a fundamental representation of 3D data, point cloud collected by various depth scanners or LiDAR sensors has been applied to diverse domains, such as autonomous driving [Yan *et al.*, 2020a] and cultural heritage reconstruction [Xu *et al.*, 2014]. However, with the increasing capabilities of 3D data acquisition, gigabytes of raw point data can be generated per second, *e.g.,* Velodyne HDL-64E can collect up to 2.2 million points per second. Therefore, large-scale point clouds pose great challenges to the subsequent transmission, storage and interconnection. Thus, the Point Cloud Sampling and Recovery (PCSR) task becomes critical for massive real-time point cloud collection and processing, which aims to sample meaningful points from dense point cloud to compress the scale of the original point cloud while preserving the local topology of discarded points for reconstruction.

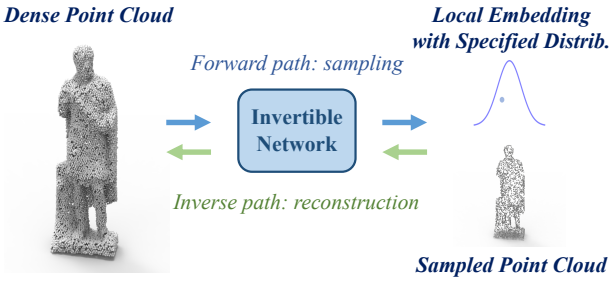

Figure 1: **Illustration of our PointLIE.** It incorporates the sampling and reconstruction processes into the same network, where the forward path transforms a dense point cloud to a sparse sub-point set and a case-agnostic latent variable made to follow a specific distribution. In the reverse path, a randomly drawn variable and the adaptively sampled point cloud are reconstructed to a dense one.

In point cloud research, significant progress has been made in the single-track task for better compressing or upsampling the input point clouds, *i.e.,* compressing point cloud with more pleasure surface approximations [Schwarz *et al.*, 2018] or upsampling sparse point cloud to dense point cloud [Yu *et al.*, 2018; Yifan *et al.*, 2019]. However, bi-directional PCSR remains challenging for several reasons: (1) Traditional methods compress point cloud through gradual sparseness. The global geometric configuration is lost in the intermediate process, making the intermediate results useless and invisible to downstream applications; (2) The relationship between data points requires huge storage. (3) The simple upsampling methods usually produce unsatisfactory recovery results due to the loss of local structure information.

To deal with the above issues, we propose a novel scheme, named Locally Invertible Embedding (PointLIE), to achieve adaptive sampling and faithful reconstruction via an invertible network for efficient storage, intermediate visibility and effective recovery. Inspired by the invertible neural network (INN), we design an INN-based framework to record the inter-points relationships without extra storage. As shown in Fig. 1, in the sampling path, PointLIE adaptively samples viewable and recovery-friendly sub-point clouds with

*Corresponding author. † Equal first authorship.

arbitrary scales while preserving the local offsets to the discarded points recursively. Furthermore, it explicitly embeds the above local topology of the discarded points into a latent variable constrained to follow a specified distribution. Therefore, when point cloud recovery is needed, PointLIE could reconstruct a faithful dense point cloud simply by sampling a **randomly-drawn** latent variable from the above distribution and traversing the inverse path of the network together with the sampled sub-points. In this way, the original point cloud can be better restored without introducing additional storage.

In practice, PointLIE contains several Decomposition Modules and Point Invertible Blocks. The former is adopted to decompose the input point features into sampled point features and offsets to the discarded ones. After this, the number of point features is halved, while the dimension of channels increases exponentially due to higher-order offsets (*i.e.,* subtracting the sampled point features from the discarded ones recursively). By applying an elaborately designed cross-connection architecture, Point Invertible Blocks could further characterize the mutual correlation between the sampled features and their offsets in each sampling scale. Following the aforementioned recursive process, PointLIE could encode all the lost information in each sampling scale into a locally invertible embedding. In the training phase, we force such an invertible embedding to conform to a pre-specified distribution (e.g. isotropic Gaussian) by using a distribution fitting loss. Due to the reversible nature of PointLIE, the recovery process can be conducted by passing through PointLIE inversely as illustrated by the green arrows in Fig. 1.

The main contributions of this paper are three folds. 1) To the best of our knowledge, this is the first work that adopts INN for the PCSR task. A novel **PointLIE** scheme is proposed to model the sampling and upsampling processes into the same network through bi-directional learning. 2) We propose a Decomposition Module to decouple the input point features into the sampled ones and offsets to their corresponding local neighbors, and Point Invertible Blocks to update them in each sampling scale. Meanwhile, a recursive invertible embedding is proposed to transform the offsets of local neighbors into a latent variable that obeys a specific distribution. 3) Extensive experiments demonstrate that PointLIE outperforms the state-of-the-art point cloud sampling and upsampling methods both quantitatively and qualitatively.

2 Related Work

2.1 Sampling Methods for Point Clouds

Traditional sampling methods, such as Farthest point sampling (FPS), have wide applications in various point cloud frameworks [Qi *et al.*, 2017; Wu *et al.*, 2019], since they can sample relatively uniformly distributed points. However, they do not consider the subsequent processing of the sampled points and may result in sub-optimal performance. Recently, there are some alternative sampling methods proposed to better capture the information of point clouds. [Nezhadarya *et al.*, 2020] introduced a critical points layer, which retains the critical points with the most active features to the next network layer. [Yan *et al.*, 2020b] adaptively shifted the sampled points to objects' surface and thus increased the robustness of the network in noisy point clouds. Other methods jointly consider sampling with downstream tasks. For example, [Dovrat *et al.*, 2019; Lang *et al.*, 2020] introduced a task-specific sampling, which can improve the results through training with task-specific loss. However, these methods improve the reconstruction mainly by joining the loss of specific tasks, while the geometric information lost in discarded points during sampling is not considered.

2.2 Upsampling Methods for Point Clouds

Point cloud upsampling aims to improve the point distribution density and uniformity. [Yu *et al.*, 2018] first proposed the neural network PU-Net, which learns point-wise features by PointNet++ [Qi *et al.*, 2017], expanding the point set in feature space, and reconstructs an upsampled point set from those features. 3PU [Yifan *et al.*, 2019] is a multi-step progressive network, which learns different levels of details in multiple steps. However, due to its progressive nature, it requires a large amount of computation and more data to supervise the intermediate output of the network. Recently, a Point Cloud Generative Adversarial Network (PU-GAN) [Li *et al.*, 2019] is designed to learn the distribution of the upsampled point set through adversarial learning. Upsampling is an ill-posed problem since a downsampled point set corresponds to multiple plausible dense point clouds. Existing deep-learning based methods directly model this ambiguous task by learning the mapping from a sparse point set to a dense one under the supervision of the ground truth dense point set. However, these methods fail to yield faithful complete reconstruction results, since the valuable information lost in the sampling process is ignored and irreversible.

2.3 Invertible Neural Network

Obtaining the measurable quantities (sampled points) from the given hidden parameters (sampling methods) is referred to as the forward process (*i.e.,* sampling). Correspondingly, the inverse process requires to infer the hidden states of a system from measurements (*i.e.,* reconstruction). The inverse process is often intractable and ill-posed because valuable information is lost in the forward process [Ardizzone *et al.*, 2018]. To fully assess the diversity of possible inverse solutions for a given measurement, invertible neural networks (INNs) are employed to estimate the complete posterior of the parameters conditioned by observation, which is widely employed in both generative models [Dinh *et al.*, 2014; Dinh *et al.*, 2016; Kingma and Dhariwal, 2018; Behrmann *et al.*, 2019; Chen *et al.*, 2019] and classification tasks [Jacobsen *et al.*, 2018]. Unlike traditional deep neural networks, which attempt to directly model the ambiguous problem of inferring the non-unique feasible result, INNs focus on learning the determinate forward process, using latent variables to capture the lost information. Due to the invertibility, the inverse process can be obtained for free by running through the network backwards.

3 Methods

3.1 Task Overview

Given a dense point set $\mathcal{Q} = \{q_i\}_{i=1}^{N}$, the goal of point cloud sampling and recovery (PCSR) with scale factor r is to adap-

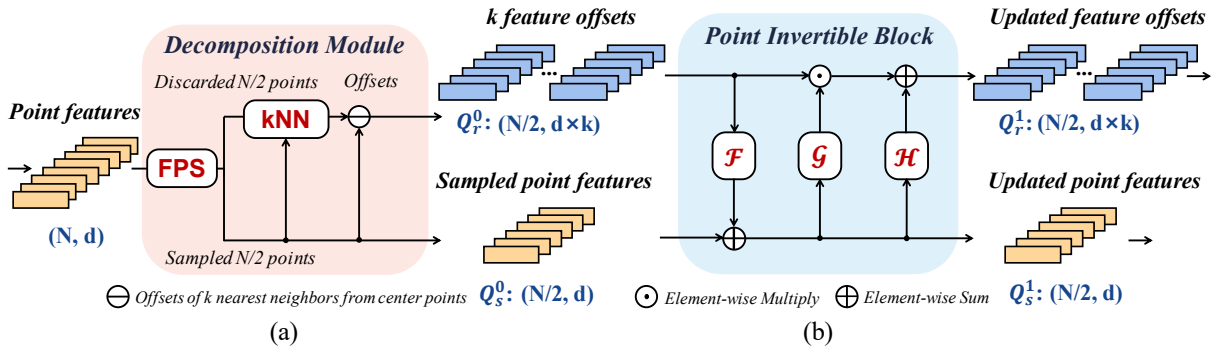

Figure 2: **The internal structure of the Decomposition Module and Point Invertible Block.** (a) illustrates the Decomposition Module, which decomposes the input point features into the sampled ones and their offsets to the k nearest neighboring features in the discarded points. (b) shows the Point Invertible (PI) Block, which updates the feature offsets and sampled point features into their new counterparts.

tively sample it into a sparse sub-point cloud $\hat{\mathcal{P}} = \{\hat{p}_j\}_{j=1}^{N/r}$ without any extra preservation, and then reconstruct a dense point cloud $\hat{\mathcal{Q}} = \{\hat{q}_i\}_{i=1}^{N}$ from $\hat{\mathcal{P}}$. To achieve the above goal, PointLIE is proposed as shown in Fig. 1. The forward path transforms the dense point cloud into sampled points $\hat{\mathcal{P}}$ and a local invertible embedding z containing the geometric information lost during sampling. Due to the reversible nature of PointLIE, the inverse path can reconstruct a faithful dense point cloud for free by running through the PointLIE backwards. The whole process can be formulated as,

$$f_\theta(\mathcal{Q}) = (\hat{\mathcal{P}}, z), \text{ s.t. } z \sim p(z), \tag{1}$$

$$f_\theta^{-1}(\hat{\mathcal{P}}, z^\star) = \hat{\mathcal{Q}}, \ \ z^\star \sim p(z). \tag{2}$$

where $f_\theta(\cdot)$ denotes the forward path of our model, and z is the locally invertible embedding generated in the forward path, which is made to follow a specific distribution $p(z)$. Note that here $z \sim p(z)$ is case-agnostic instead of case-specific (*i.e.,* $z \sim p(z|\hat{\mathcal{P}})$). Therefore, there is no need to store z after sampling, and we can just randomly draw an embedding $z^\star$ from the distribution $p(z)$ in the inverse path. $\hat{\mathcal{P}}$ and $z^\star$ are used to reconstruct a faithful $\hat{\mathcal{Q}}$ through the inverse process $f_\theta^{-1}(\cdot)$.

3.2 Invertible Architecture

To achieve the invertible operations, we firstly construct a *Rescale Layer* by stacking a *Decomposition Module* and M *Point Invertible Blocks* (PI Blocks) detailed in Fig. 2. When dealing with the PCSR with the scale factor r, we stack s *Rescale Layers* (s=$\lfloor \log_2 r \rfloor$) to obtain the entire framework of PointLIE as illustrated in Fig. 3.

Decomposition Module

As shown in Fig. 2 (a), during the sampling process, the decomposition module decouples the lost geometric information contained in discarded points from the sampled points. Specifically, for input point features with the shape of (N, d), we first conduct farthest point sampling (FPS) to select $N/2$ points while the remaining $N/2$ points are regarded as discarded points. To make the network preserve the information in discarded points, for each sampled point $q_i \in \mathcal{Q}_s$, we find its k nearest neighbors in the discarded points and denote the spatial offsets from q_i to its neighbors as $\mathcal{Q}_r$. Here we use offsets rather than spatial coordinates of its neighbors since deep neural networks are more capable of learning the residues, and it is also easier to make the residues follow the isotropic Gaussian distribution. The decomposition module outputs two branches of features, *i.e.,* offsets to k nearest features $\mathcal{Q}_r^0$ with shape $(N/2, d \times k)$ and sampled point features $\mathcal{Q}_s^0$ with shape $(N/2, d)$.

Point Invertible Block

To further characterize the representation of the two branches during the forward path, we design a point invertible block to update features, inspired by the coupling layer in generative models [Dinh *et al.*, 2014; Dinh *et al.*, 2016; Xiao *et al.*, 2020]. As shown in Fig. 2 (b), each PI block takes two branches as input (*i.e.,* the $1/2$ sampled point features $\mathcal{Q}_s^l$ and their kNN offsets $\mathcal{Q}_r^l$) and generates updated features $\mathcal{Q}_s^{l+1}$ and offsets $\mathcal{Q}_r^{l+1}$ by Eq. (3) (4),

$$\mathcal{Q}_s^{l+1} = \mathcal{Q}_s^l \odot \exp(\mathcal{Q}_r^l) + \mathcal{F}(\mathcal{Q}_r^l), \tag{3}$$

$$\mathcal{Q}_r^{l+1} = \mathcal{Q}_r^l \odot \exp(\mathcal{G}(\mathcal{Q}_s^{l+1})) + \mathcal{H}(\mathcal{Q}_s^{l+1}), \tag{4}$$

where l denotes passing through the l-th PI block, and $\mathcal{F}$, $\mathcal{G}$, $\mathcal{H}$ are three independent nonlinear transformations. We use several stacked `conv1d` with the nonlinear activation for $\mathcal{F}$, $\mathcal{G}$, and the dense feature extractor in [Yifan *et al.*, 2019] for $\mathcal{H}$. PI blocks only enhance the representation of sampled features and neighboring offsets gradually, while the shapes of inputs and outputs of each PI block remain unchanged.

Recursive Offset Residue Embedding

Fig. 3 illustrates the overall bi-directional pipeline of PointLIE for the PCSR task. By stacking s rescale layers, where each of them contains a decomposition module and M PI blocks, we construct a hierarchical structure for PCSR with arbitrary scales. For each rescale layer, taking point features with shape (N, d) as input, it will generate sampled features with shape $(N/2, d)$ and feature offsets to k nearest neighbors with shape $(N/2, k \times d)$. Then a channel-wise concatenation is conducted to merge the sampled features and their neighboring offsets to generate new point features. These 'higher-order' point features will continue to be input to the

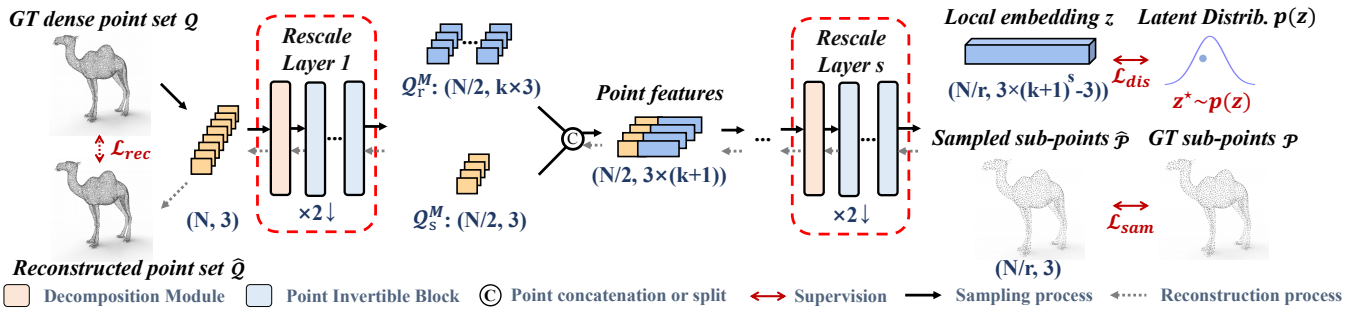

Figure 3: **Illustration of the overall pipeline of PointLIE.** The black solid and grey dashed arrows indicate the flow of sampling and reconstruction respectively. During the sampling path, the lost information contained in the offsets to neighboring features are encoded into a latent variable z following the distribution $p(z)$ by passing through s Rescale Layers. Composed of a Decomposition Module and multiple Point Invertible Blocks, each Rescale Layer samples the point set by half. The reconstruction can be achieved through reverse operations.

next rescale layer. Therefore, the final embedding can be expanded to a series of high-order offsets recursively. $\mathcal{Q}_s^M$ and $\mathcal{Q}_r^M$ generated by the last Rescale Layer are treated as the adaptively sampled sub-point cloud $\hat{\mathcal{P}}$ and the embedding z.

Inverse Reconstruction Process

To reconstruct the original dense point set, we use the adaptively sampled point set $\hat{\mathcal{P}}$ and a randomly-drawn embedding $z^\star \sim p(z)$ as two branches of input to the reverse path of PointLIE (*i.e.,* rescale layer s, $s-1$, ..., 1) as indicated by the gray dashed arrows in Fig. 3. In each rescale layer, they will also flow in a reverse direction (*i.e.,* PI block M, $M-1$, ..., 1, decomposition module).

The inverse operations of the PI Blocks and the decomposition module are shown in Fig. 2 of supp. In the reverse path, the $(l+1)$-th PI block aims to recover the neighboring offsets $\mathcal{Q}_r^l$ and the sampled features $\mathcal{Q}_s^l$ from $\mathcal{Q}_r^{l+1}$ and $\mathcal{Q}_s^{l+1}$. Considering the inputs $\mathcal{Q}_r^{l+1}$, $\mathcal{Q}_s^{l+1}$ with shapes $(N, k \times d)$ and (N, d), the reverse process of Eq. (3)(4) can be expressed as,

$$\mathcal{Q}_r^l = (\mathcal{Q}_r^{l+1} - \mathcal{H}(\mathcal{Q}_s^{l+1})) \odot \exp(-\mathcal{G}(\mathcal{Q}_s^{l+1})), \quad (5)$$

$$\mathcal{Q}_s^l = (\mathcal{Q}_s^{l+1} - \mathcal{F}(\mathcal{Q}_r^l)) \odot \exp(-\mathcal{Q}_r^l). \quad (6)$$

After reversely passing through M PI blocks, the output $\mathcal{Q}_r^0$, $\mathcal{Q}_s^0$ will flow into the decomposition module reversely. In detail, $\mathcal{Q}_r^0$ will be evenly split into k offset matrices $\{Q_r^{(i)}\}_{i=1}^k$ along the channel dimension, where $Q_r^{(i)}$ with shape (N, d) represents the offsets to the i-th nearest neighbour for each point in $\mathcal{Q}_s^0$ in the discarded points. Then element-wise addition will be conducted between each $Q_r^{(i)}$ and $\mathcal{Q}_s^0$ respectively, obtaining features of recovered discarded points $\mathcal{Q}_d$ with shape (kN, d). $\mathcal{Q}_d$ will be concatenated with $\mathcal{Q}_s^0$ in a point-wise manner to form a candidate recovered point set Q_c with shape $((k+1) \times N, d)$, whose first three dimensions in d record the spatial coordinates. To guarantee the uniformity of the reconstructed points, we use FPS to select $2N$ point features from Q_c based on their coordinates.

Analogously, these $\times 2$ reconstructed point features will be evenly split into $(k+1)$ parts, where the first part and the remaining k parts are taken as the sampled point features $\mathcal{Q}_s^M$ and the neighboring offsets $\mathcal{Q}_r^M$ respectively. Then, they will be fed into the next reversed rescale layer to conduct another $\times 2$ reconstruction. Supported by Theorem 1, a faithful dense point cloud can be reconstructed progressively. The proof is provided in the supp.

Theorem 1. *Suppose the generated invertible local embedding z is subject to a latent distribution $p(z)$. In the recovery process, by randomly sampling a $z^\star$ from $p(z)$ and passing it through the reverse path, the reconstructed dense point cloud $\hat{\mathcal{Q}}$ will necessarily conform to the distribution of the real point cloud $p(\mathcal{Q})$.*

3.3 Training Objectives

To improve the reconstruction result, PointLIE models the bi-directional transformation between the dense point cloud $\mathcal{Q}$ and the sampled point cloud $\hat{\mathcal{P}}$ with a latent distribution $p(z)$. Therefore, the total loss contains the following parts.

Sparse Point Sampling Loss

Since the generated sampled point cloud $\hat{\mathcal{P}}$ is not the subset of $\mathcal{Q}$, we adopt the Earth Mover's distance loss (EMD) [Fan *et al.*, 2017] $\mathcal{L}_{\text{sam}}$ to restrict $\hat{\mathcal{P}}$ to approach $\mathcal{P}$.

$$\mathcal{L}_{\text{sam}} = \mathcal{L}_{\text{emd}}(\hat{\mathcal{P}}, \mathcal{P}) = \min_{\phi:\hat{\mathcal{P}}\to\mathcal{P}} \sum_{\hat{p}_j \in \hat{\mathcal{P}}} \|\hat{p}_j - \phi(\hat{p}_j)\|_2, \quad (7)$$

where $\mathcal{P}$ denotes the ground truth sub-point set uniformly sampled from the original dense point set $\mathcal{Q}$ by FPS, and $\phi : \hat{\mathcal{P}} \to \mathcal{P}$ is a bijective mapping.

Dense Point Reconstruction Loss

To reconstruct finer results, besides using EMD loss to restrict the geometric details of the predicted sub-point set $\hat{\mathcal{P}}$, the reconstructed point set $\hat{\mathcal{Q}}$ should also be uniformly distributed on the surface of objects, thus repulsion loss $\mathcal{L}_{\text{rep}}$ [Yu *et al.*, 2018] and uniform loss $\mathcal{L}_{\text{uni}}$ [Li *et al.*, 2019] are used to distribute the recovered point set $\hat{\mathcal{Q}}$ uniformly. So the total loss for reconstruction is formulated as,

$$\mathcal{L}_{\text{rec}} = \lambda_{\text{emd}}\mathcal{L}_{\text{emd}}(\hat{\mathcal{Q}}, \mathcal{Q}) + \lambda_{\text{rep}}\mathcal{L}_{\text{rep}}(\hat{\mathcal{Q}}) + \lambda_{\text{uni}}\mathcal{L}_{\text{uni}}(\hat{\mathcal{Q}}). \quad (8)$$

Distribution Fitting Loss

Distribution fitting loss is used to encourage the distribution of the generated local embedding $f_\theta^z(\mathcal{Q})$ to approach the latent distribution $p(z)$, which is the sufficient condition for the

Method	Sampling mode	Network size (mean)	Scale factor 4 (10^{-3})			Scale factor 8 (10^{-3})			Scale factor 16 (10^{-3})		
			CD	HD	P2F	CD	HD	P2F	CD	HD	P2F
PU-Net	FPS	16.5 MB	0.49	4.78	8.81	0.92	10.21	14.92	1.05	12.22	17.38
3PU	FPS	92.5 MB	0.41	4.86	2.72	0.54	8.91	3.68	1.02	14.89	**5.94**
PU-GAN	FPS	16.7 MB	0.24	3.16	1.97	0.75	9.02	4.57	0.84	14.29	8.15
PU-Net	SampleNet	21.2 MB	0.49	4.95	9.02	0.89	10.19	14.50	1.04	12.55	18.88
PU-GAN	SampleNet	21.4 MB	0.23	2.89	**1.93**	0.71	8.03	4.57	0.80	14.96	8.06
PointLIE	-	24.6 MB	**0.21**	**1.71**	2.20	**0.35**	**4.68**	**3.37**	**0.61**	**9.20**	6.80

Table 1: Performance comparison of PointLIE with the state-of-the-art point cloud sampling and reconstruction methods. **Bold** denotes the best performance.

reconstructed point set $\hat{\mathcal{Q}}$ to follow the real distribution of the original dense point set Q as proved in the Theorem 1. In practice, the cross-entropy loss (CE) is employed to measure the difference between the distributions of the generated embedding $f_\theta^z(\mathcal{Q})$ and $p(z)$. Here $p(z)$ is set as an isotropic Gaussian distribution,

$$\begin{aligned}\mathcal{L}_{\text{dis}} &= \text{CE}[f_\theta^z[p(Q)], p(z)] = -\mathbb{E}_{f_\theta^z[p(Q)]}[\log p(z)] \\ &= -\mathbb{E}_{p(Q)}[\log p(z = f_\theta^z(Q))].\end{aligned} \tag{9}$$

Compound Loss

Overall, we train our PointLIE in an end-to-end manner by minimizing the total loss $\mathcal{L}$ as follows,

$$\mathcal{L} = \lambda_{\text{sam}}\mathcal{L}_{\text{sam}} + \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{dis}}\mathcal{L}_{\text{dis}}. \tag{10}$$

4 Experiments

4.1 Dataset and Metrics

To fully evaluate the proposed PointLIE, we compared our method with the state-of-the-art methods on PU-147 [Li *et al.*, 2019] dataset, which follows the official split of 120/27 for our training and testing sets. We first used the Poisson disk sampling (PDS) to uniformly sample 8192 points from each original mesh as our ground truth dense point set (GT). Then we used different scale factors (*i.e.,* 4, 8 and 16), and sampling modes (*i.e.,* FPS and learnable sampling methods) to sample sub-point cloud from GT, and compared the reconstruction results with GT. To quantitatively evaluate the performance of different methods, three commonly-used metrics are adopted, *i.e.,* Chamfer distance (CD), Hausdorff distance (HD) and point-to-surface distance (P2F). The lower the metric values are, the better the reconstruction results are.

4.2 Implementation Details

Under the premise of balancing efficiency and effectiveness, we set PI block number M as 8 in the $4\times$ scale task, and M as 4 in the rest $8\times$ and $16\times$ tasks. Furthermore, we set k as 3 to ensure that the information in the discarded points can be sufficiently preserved. More details are shown in the supp.

4.3 Quantitative Results

Point Cloud Sampling and Recovery. In Tab. 1 we compared the results of PCSR with recent state-of-the-art methods: PU-Net [Yu *et al.*, 2018], 3PU [Yifan *et al.*, 2019] and PU-GAN [Li *et al.*, 2019]. To fairly compare with previous methods, we used different sampling modes to sample the sub-point clouds (*i.e.,* FPS and adaptive sampling [Lang *et al.*, 2020]). The upper and lower part of Tab. 1 show the results by using FPS and the learnable sampling respectively. Among all, our PointLIE achieved the best results for most of the evaluation metrics, especially for large scale PCSR tasks ($\times 8$ and $\times 16$). Furthermore, we compared with the state-of-the-art adaptive sampling methods (*i.e.,* SampleNet) combined with the most appealing upsampling methods. We used the official codes of SampleNet [Lang *et al.*, 2020] and jointly trained it with the downstream upsampling methods (i.e., PU-Net and PU-GAN). The results show that SampleNet cannot effectively improve the reconstruction performance.

Method	Scale 4 (10^{-3})		Scale 16 (10^{-3})	
	CD	HD	CD	HD
PU-Net	0.52	7.37	2.46	14.37
3PU	0.72	8.94	2.17	12.67
PU-GAN (-)	0.57	7.25	2.20	18.82
PU-GAN	**0.28**	**4.64**	2.07	16.59
PointLIE	0.32	4.93	**1.98**	**12.08**

Table 2: Performance comparison of PointLIE with the state-of-the-arts for upsampling, where PU-GAN (-) denotes the results of PU-GAN without the discriminator. **Bold** denotes the best performance.

Point Cloud Upsampling. Our PointLIE can also be used as a general point cloud upsampling framework by feeding a sparse point cloud into the inverse stream of the trained model. For a fair comparison, we followed the experiment setting of [Li *et al.*, 2019], feeding *randomly* sampled 2048 or 512 points to predict 8192 points. As shown in Tab. 2, our PointLIE achieved comparable results in all evaluation metrics. Particularly, our results by far exceed all previous methods without adversarial learning (*e.g.,* PU-GAN (-)), and even outperforms the complete PU-GAN in $\times 16$ task. This result confirms that the performance improvement of PU-GAN mainly comes from the discriminator rather than the model architecture itself, while our architecture design can achieve superior upsampling results for both dense and sparse input.

4.4 Qualitative Results

We also compared our qualitative results with PU-Net, 3PU and PU-GAN for PCSR on different scales. Here PU-Net and PU-GAN took the points sampled by SampleNet as in-

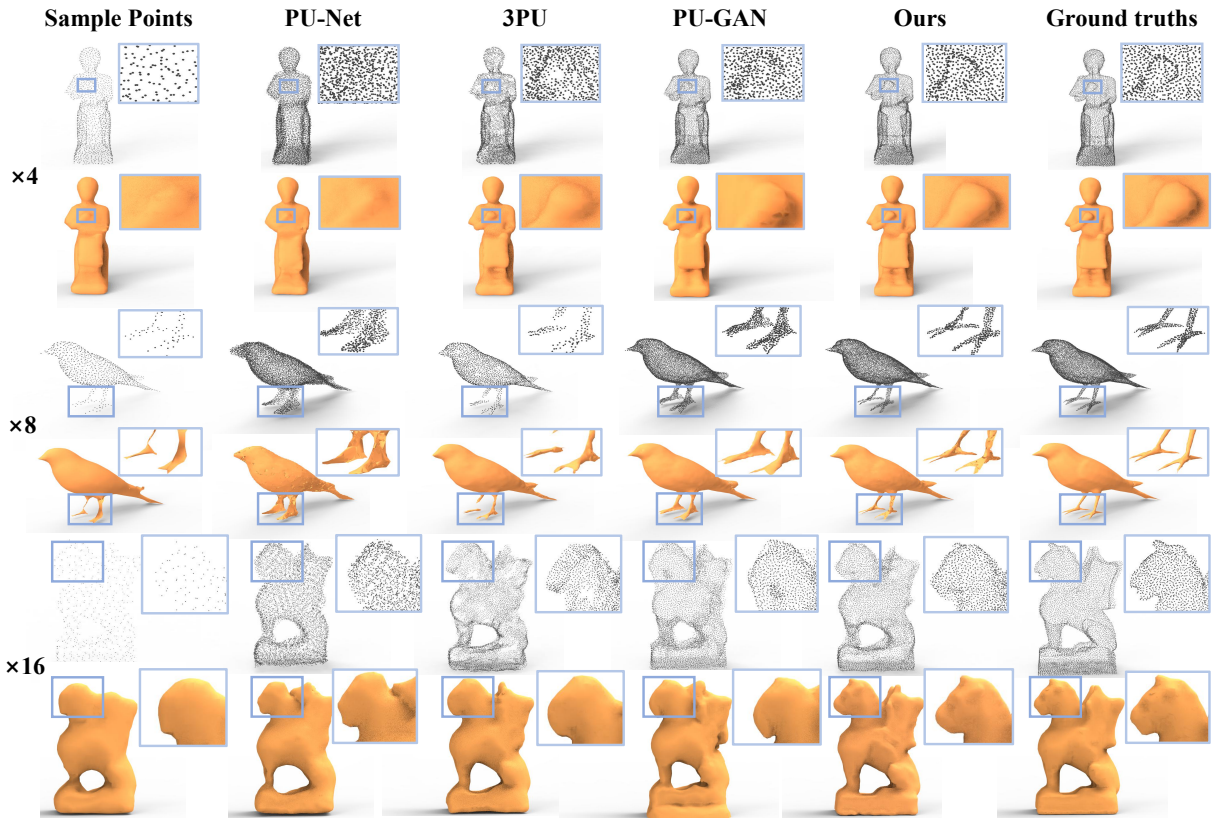

Figure 4: Comparing the point cloud sampling and recovery (PCSR) and surface reconstruction results produced by different methods.

puts. Fig. 4 shows the visual results of PCSR and surface reconstruction by [Kazhdan and Hoppe, 2013]. As shown in Fig. 4, PointLIE can reconstruct more fine-grained details, especially for local geometric shapes (*e.g.,* human hands, bird claws and dragon horns), while other methods tend to produce more noisy and nonuniform point sets, resulting in more artifacts and ambiguities on the reconstructed surfaces.

4.5 Ablation Study

To further demonstrate the effectiveness of our proposed method, we design an ablation study for different training modes and data feeding. In Tab. 3, we first show the result produced without bi-directional learning (only training the inverse process) in the first two rows. These results show that only using the inverse process during training cannot make the model learn the distribution of the reconstructed point cloud. Then, we used the proposed bi-directional training strategy mentioned above, which made a remarkable improvement dealing with randomly or uniformly sampled point clouds. Finally, when we used the sub-point set adaptively sampled by our network, further improvement is achieved.

Model	Sample mode	Bi-direction	CD	HD
PointLIE-R	Random	✗	4.93	16.59
PointLIE-R	FPS	✗	2.32	7.58
PointLIE-R	Random	✓	0.32	4.93
PointLIE-R	FPS	✓	0.27	2.73
PointLIE-R	PointLIE-S	✓	**0.21**	**1.71**

Table 3: Ablation study for PointLIE by using different training modes and inputs, where R and S refer to reconstruction and sampling process respectively. **Bold** denotes the best performance.

5 Conclusion

We are the first to adopt the INN for the PCSR task and propose a novel framework PointLIE, which models the sampling and upsampling into the same network through bi-directional learning. By using decomposition modules and point invertible blocks to decouple the discarded points from the sampled points and update the features, our PointLIE can finely restore the original point cloud with a recursive invertible embedding using the reversed operations.

Acknowledgments

The work was supported in part by NSFC-Youth 61902335, by Key Area R&D Program of Guangdong Province with grant No. 2018B030338001, by the National Key R&D Program of China with grant No. 2018YFB1800800, by Shenzhen Outstanding Talents Training Fund, by Guangdong Research Project No. 2017ZT07X152, by Guangdong Regional Joint Fund-Key Projects 2019B1515120039, by the NSFC 61931024 & 81922046, by helix0n biotechnology company Fund and CCF-Tencent Open Fund.

References

[Ardizzone *et al.*, 2018] Lynton Ardizzone, Jakob Kruse, Sebastian Wirkert, Daniel Rahner, Eric W Pellegrini, Ralf S Klessen, Lena Maier-Hein, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. *arXiv preprint arXiv:1808.04730*, 2018.

[Behrmann *et al.*, 2019] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, pages 573–582, 2019.

[Chen *et al.*, 2019] Ricky TQ Chen, Jens Behrmann, David K Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. In *Advances in Neural Information Processing Systems*, pages 9916–9926, 2019.

[Dinh *et al.*, 2014] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[Dinh *et al.*, 2016] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[Dovrat *et al.*, 2019] Oren Dovrat, Itai Lang, and Shai Avidan. Learning to sample. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2760–2769, 2019.

[Fan *et al.*, 2017] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[Jacobsen *et al.*, 2018] Jörn-Henrik Jacobsen, Arnold Smeulders, and Edouard Oyallon. i-revnet: Deep invertible networks. *arXiv preprint arXiv:1802.07088*, 2018.

[Kazhdan and Hoppe, 2013] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013.

[Kingma and Dhariwal, 2018] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in neural information processing systems*, pages 10215–10224, 2018.

[Lang *et al.*, 2020] Itai Lang, Asaf Manor, and Shai Avidan. Samplenet: Differentiable point cloud sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7578–7588, 2020.

[Li *et al.*, 2019] 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.

[Nezhadarya *et al.*, 2020] Ehsan Nezhadarya, Ehsan Taghavi, Ryan Razani, Bingbing Liu, and Jun Luo. Adaptive hierarchical down-sampling for point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12956–12964, 2020.

[Qi *et al.*, 2017] 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.

[Schwarz *et al.*, 2018] Sebastian Schwarz, Marius Preda, Vittorio Baroncini, Madhukar Budagavi, Pablo Cesar, Philip A Chou, Robert A Cohen, Maja Krivokuća, Sébastien Lasserre, Zhu Li, et al. Emerging mpeg standards for point cloud compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(1):133–148, 2018.

[Wu *et al.*, 2019] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019.

[Xiao *et al.*, 2020] Mingqing Xiao, Shuxin Zheng, Chang Liu, Yaolong Wang, Di He, Guolin Ke, Jiang Bian, Zhouchen Lin, and Tie-Yan Liu. Invertible image rescaling. In *European Conference on Computer Vision*, pages 126–144. Springer, 2020.

[Xu *et al.*, 2014] Zhihua Xu, Lixin Wu, Yonglin Shen, Fashuai Li, Qiuling Wang, and Ran Wang. Tridimensional reconstruction applied to cultural heritage with the use of camera-equipped uav and terrestrial laser scanner. *Remote Sensing*, 6(11):10413–10434, 2014.

[Yan *et al.*, 2020a] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[Yan *et al.*, 2020b] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5589–5598, 2020.

[Yifan *et al.*, 2019] 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.

[Yu *et al.*, 2018] 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.