APPENDIX OF SPAN-NET

Xiaofei Qin, Anluo Yi, Jie Zhang, Shiwei Tao

University of Shanghai for Science and Technology
Shanghai, 200093, China
xiaofei.qin@usst.edu.cn
{232260517, 233370860, 243370925}@st.usst.edu.cn

1. RELATED WORK

1.1. Point cloud completion

Point cloud completion is a generative and estimation problem derived from partial point clouds, playing a crucial role in the applications of 3D computer vision. According to the used different backbones, the existing point cloud completion methods could be roughly divided into two categories: convolution-based methods and Transformer-based methods.

The existing convolution-based [1, 2, 3, 4, 5, 6, 7, 8, 9] generally handle the point cloud completion task by extracting features from the input incomplete point clouds via either 3D or 2D convolutions. However, the convolution-based methods, are essentially based on Convolutional Neural Networks (CNNs). These networks are characterized by limited receptive fields and struggle with modeling variable-size inputs and long-range semantic relationships [10]. Consequently, they often exhibit suboptimal performance in point cloud completion tasks.

To address this problem, many Transformer-based methods [11, 12, 13, 14, 15] have been proposed recently, leveraging the capability of Transformers to effectively manage long-range dependencies in point clouds [10]. PionTr [16], presents an adaptive point cloud Transformer framework for point cloud generation. These two works demonstrate that Transformers are proficient in modeling long-range dependencies between local structures within point clouds. Subsequently, many Transformer-based methods have emerged [11, 17, 18, 19, 20, 21, 22]. However, as these methods have noted, Transformers inherently lack geometric priors (such as translation invariance) that are characteristic of CNNs. This deficiency may result in the loss of structural knowledge or detailed local information, thereby making it challenging to recover these details during decoding. Consequently, the contributions of previous Transformer-based methods have primarily focused on developing mechanisms to extract local and structural information from point cloud data. For example, PoinTr [16] devises a geometry-aware block that models the local geometric relationships explicitly. SeedFormer [13] introduces a novel shape representation named Patch Seed and develops an enhanced upsample Transformer for the generation process. ProxyFormer [15] strengthens structural knowledge through a proxy alignment mechanism. GSFormer [14] proposes a novel graph-structured representation for point cloud generation.

1.2. Noise processing based on latent space

In generative models, the latent space, serving as a high-order abstract representation of data, has emerged as a pivotal vehicle for noise suppression and anomaly detection [23, 24, 25]. Through mechanisms such as manipulating the latent distribution of features, constraining reconstruction errors, or adversarial training, it has demonstrated remarkable efficacy in image, time-series, and graph-structured data. DSCNet [26] integrates latent representations with subspace regularization, achieving automatic pruning of noise subspaces through low-rank constraints, thereby enabling precise segmentation of tubular structures (e.g., blood vessels, roads). For time-series data, OmniAnomaly [27] employs stochastic variable connection techniques to model multimodal latent trajectories, achieving state-of-the-art (SOTA) performance in server log anomaly detection. ADGAN [28] utilizes latent space interpolation to generate synthetic anomalies, addressing the challenge of missing negative samples and thereby facilitating adversarial training of the model. GLASS [29] employs latent space mapping to controllably synthesize a broader range of industrial anomaly samples.

For point cloud data, traditional methods typically employ various filtering techniques, such as Gaussian filtering and mean filtering, to smooth noise. However, these approaches are ineffective in handling sparse point clouds captured in real-world scenarios. Latent space methods, by transforming noise identification and removal into a decoupling problem in high-dimensional feature spaces, have validated their effectiveness in the field of 2D images. Therefore, in this paper, noise identification and removal are also conducted within the latent space.

This work was supported by the National Key R&D Program of China under Project [2021YFB2802300]

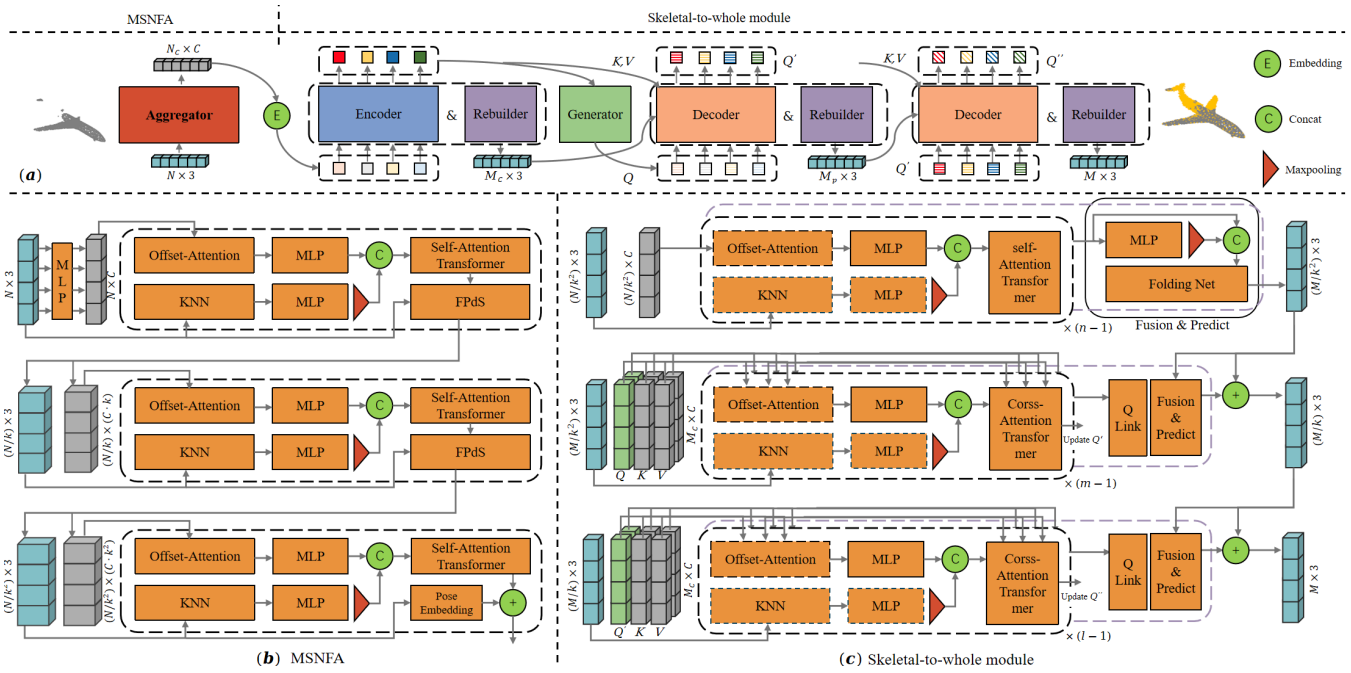

Fig. 1. Architecture of SPA-Net and its two modules.(**a**) SPA-Net: it takes partial point cloud (the gray part) as input, and predicts the unknown part of point cloud (the yellow part). SPA-Net mainly consists of Multi-scale neighbor feature aggregator (MSNFA) and the skeletal-to-whole rebuilding module. (**b**) MSNFA: it provides three scales of features through three stages of feature extraction and two feature aggregation processes. In feature extraction stages, MSNFA utilizes an offset-attention layer to model global features and a KNN-based graph convolution module to capture structure knowledge and detailed local information. In feature aggregation processes, MSNFA aggregates and assigns the previous-level features to the next level during the Furthest Point down-Sampling (FPdS) process. (**c**) skeletal-to-whole rebuilding module: it employs three rebuilders to predict three scales of point clouds. The dimension of the features input to the three rebuilders is kept consistent and relatively small to ensure that the computational complexity of attention matrix multiplications is acceptable. Three rebuilders are progressively fused together to provide close geometric and semantic connections between adjacent scales of point clouds.

2. SUPPLEMENTARY METHODS

2.1. Architectures of previous work

As shown in Fig. 1, SPA-Net mainly consists of two parts, namely the MSNFA and the skeletal-to-whole rebuilding module. Specifically, the MSNFA provides three scales of features through three stages of feature extraction and two feature aggregation processes.

In each feature extraction stage, the MSNFA utilizes an offset-attention layer to model global features and a KNN-based graph convolution module to capture structure knowledge and detailed local information. Offset-attention has been proven to be effective in handling the invariance of point clouds to rigid transformations [30], thereby ensuring the stability of global features. The KNN-based graph convolution methods capture the features of nearest-neighbor points in the point cloud, which constituting local information and structure knowledge.

In each feature aggregation process, the MSNFA aggregates and assigns the previous-level features to the next level during the Furthest Point down-Sampling (FPdS) process. This method facilitates the gradual aggregation of points and their corresponding features while establishing a direct link between the point cloud features across different scales. Without this aggregation process, the three scales

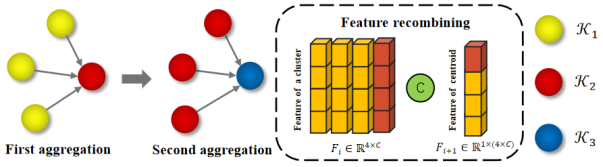

Fig. 2. Illustration of point feature aggregation. The point cloud data is partitioned into multiple clusters, assuming each cluster contains four points for sake of illustration. The center point of each cluster is selected as a representative of the entire cluster by FPdS in the aggregation process (i.e., only retain the centroid). The feature of the centroid is concatenated from the features of all the points that make up the cluster.

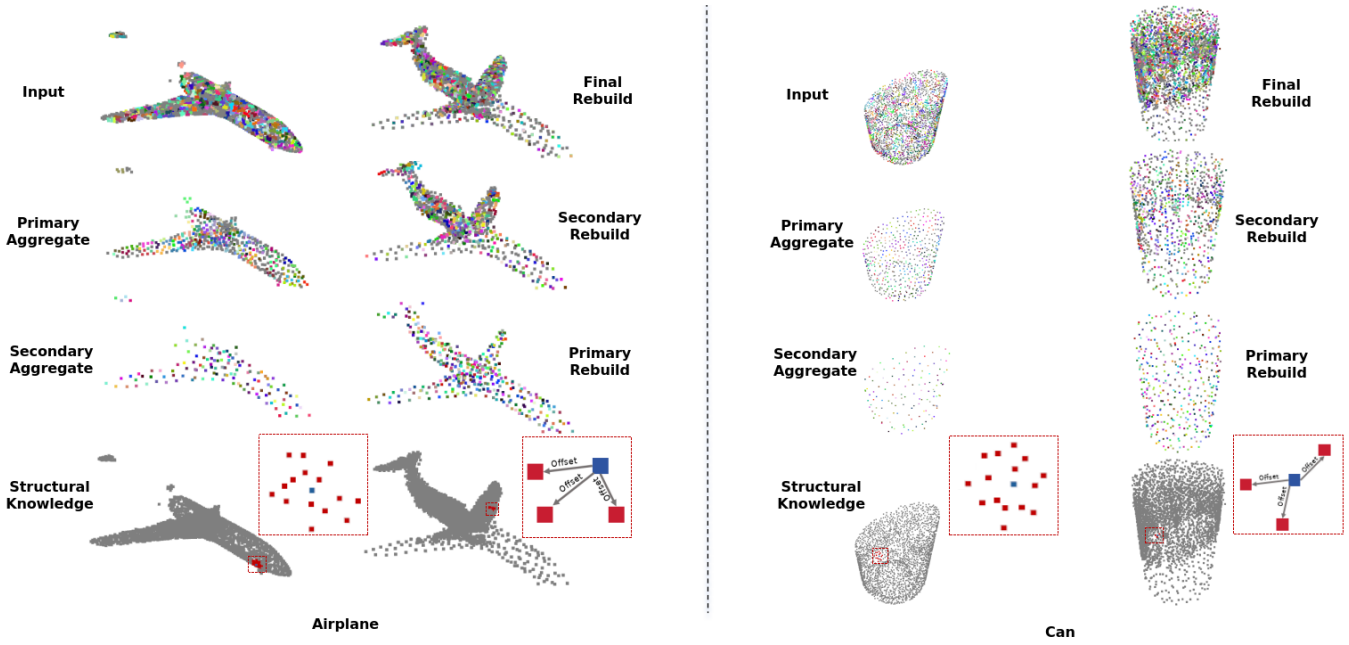

Fig. 3. Inference examples on Airplane and Can. First column (aggregating process): Points in the same color indicating membership in the same cluster. The fourth image highlights a local structure where graph convolution extracts structural knowledge and detailed local information, aggregating these to the blue cluster center. Second column (rebuilding process): Points in the first two images originate from the same lower-level point. The fourth image shows the higher-level point cloud generated by adding offsets to the lower-level blue points, resulting in a combined red and blue point cloud.

of feature extraction stage would process point clouds separately, leading to a lack of dynamic updates of features and fusion of cross scale information. This deficiency can significantly limit the completeness and consistency of feature representations across different scales. By contrast, the proposed aggregation process avoids these drawbacks by unifying feature extraction across multiple scales, ensuring that the feature representation maintains both cohesion and completeness and preserves critical structural and contextual relationships within the point cloud data.

The skeletal-to-whole rebuilding module employs three rebuilders to predict three scales of point clouds. The first rebuilder utilizes the features from the encoder's last layer to predict the skeletal point cloud. The second rebuilder utilizes the features from the decoder's middle layer to predict offsets relative to the skeletal point cloud. By adding these predicted offsets to the skeletal point cloud, the second-scale point cloud is obtained. The third rebuilder utilizes the features from the decoder's last layer to predict offsets relative to the second-scale point cloud, thereby the final-scale point cloud can be generated similarly.

The dimension of the features input to the three rebuilders is kept consistent and relatively small to ensure that the computational complexity of attention matrix multiplications is acceptable. As mentioned earlier, reconstructing high-precision point clouds from features of relatively small

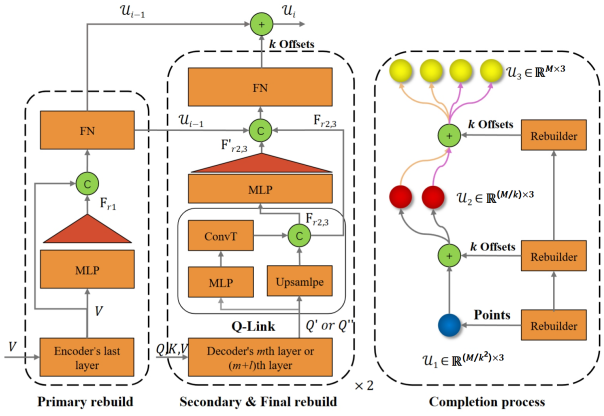

Fig. 4. Architecture of the skeletal-to-whole rebuilding module, consisting of three rebuilders. The first rebuilder includes the encoder's last layer and a Fusion & Prediction module. The second and third rebuilder includes the decoder's middle or last layer, a Q-Link, and a Fusion & Prediction module. The point clouds are completed in a level-by-level manner, by accumulating the predicted offsets to the previous level point clouds.

dimension is a significant challenge. To address this issue, we establish close geometric and semantic connections between adjacent scales of point cloud. Specifically, we utilize

the offset addition method to update the lower-level point cloud, thereby generating the higher-level point cloud and establishing close geometric connections. For the semantic connections, firstly, we feed the lower-level point clouds to the first m layers and the last l layers of the decoder, and employ a KNN to learn structure knowledge and detailed local information of the lower-level point cloud. Secondly, we feed the lower-level point clouds to the Fusion & Prediction modules to assist the generation of the higher-level point clouds. This module enhances the continuity and consistency of information across different scales, and reduces the loss of details caused by level-by-level rebuilding.

The inference examples of SPA-Net. In Fig.3, each object is represented by two columns of images. In the first column, the aggregating process is depicted, with points of the same color indicating membership in the same cluster. The fourth image in this column illustrates a local structure, where the model employs graph convolution to acquire structural knowledge and detailed local information, subsequently aggregating these to the cluster center (blue point). In the second column, the rebuilding process is shown. In the first two images in this column, points of the same color originate from the same lower-level point. The fourth image shows the higher-level point cloud being generated by adding multiple offsets to the lower-level point cloud (blue points), thereby generating the higher-level point cloud composed of both red and blue points.

2.2. Motivation of this work

In the Transformer-based point cloud completion algorithm paradigm, the primary task of the Encoder is to transform the input point cloud data into high-dimensional semantic feature representations. The Decoder then generates a complete point cloud based on the semantic features extracted by the Encoder. The point cloud completion algorithm designed in previous work of this paper incorporates a multi-scale neighboring feature aggregator before the Encoder, which progressively extracts local and global features from the point cloud. The input point cloud is converted into point embeddings through the multi-scale neighboring feature aggregator, equivalent to projecting the point cloud into a high-dimensional latent semantic space. In other words, the role of the multi-scale neighboring feature aggregator is to fit this projection process. In this space, each dimension represents a certain semantic feature. Consequently, the self-attention mechanism in the Encoder allows each point embedding to interact with all other point embeddings, thereby capturing their relationships and mutual influences. However, when noise points are present in the point cloud, the multi-scale neighboring feature aggregator will generate incorrect point embeddings, which in turn affect subsequent completion. This constitutes the core motivation for the technical improvements in this paper and necessitates algorithm

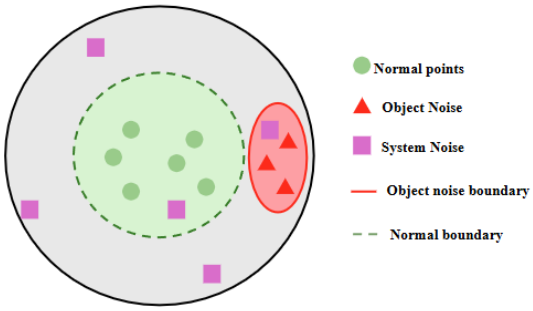

Fig. 5. Representation of latent space based on hypersphere hypothesis. In the latent space, normal point embeddings and object noise point embeddings follow different distributions, while system noise embeddings are completely random.

enhancements to address two main issues: (1) In real-world environments, there are multiple factors causing noise in point clouds. How can noise be effectively identified? (2) Due to systematic errors in devices, the point clouds of an object may all exhibit varying degrees of disturbance. Simply removing noise points may result in the loss of significant information.

To address issue (1), according to latent space theory, if the multi-scale neighboring feature aggregator fits the projection process accurately enough, the positions of noise points in the latent semantic space will be entirely different from those of normal points. This is because all normal points originate from the same object, and their distribution in the latent semantic space follows certain patterns. Inspired by this, we utilizes the differences in semantic features between noise points and normal points for noise identification, distinguishing the two based on the differences in their distributions in the latent semantic space. Specifically, we assumes that the normal point cloud of the same object follows a hyperspherical model in the latent semantic space. That is, all normal point cloud embeddings of each object in the latent semantic space are enveloped by a sphere, referred to as the "Normal boundary" in this paper.

In this paper, a first discriminator is designed to fit the normal boundary, thereby distinguishing between normal and noisy point clouds. To make the fitted normal boundary more accurate, we employs an adversarial learning strategy and designs a Gradient Affine Transformation to convert normal point embeddings into pseudo-noise point embeddings. This adversarial training mechanism enables the discriminator to precisely capture the distribution differences between noise points and normal points in the latent space. As shown in Figure 7, normal point clouds are distributed within the hypersphere, while noise points are scattered outside the sphere. Notably, this chapter classifies noise into two types based on their semantic features: random noise (which can be distributed anywhere in the latent space) and object noise (point

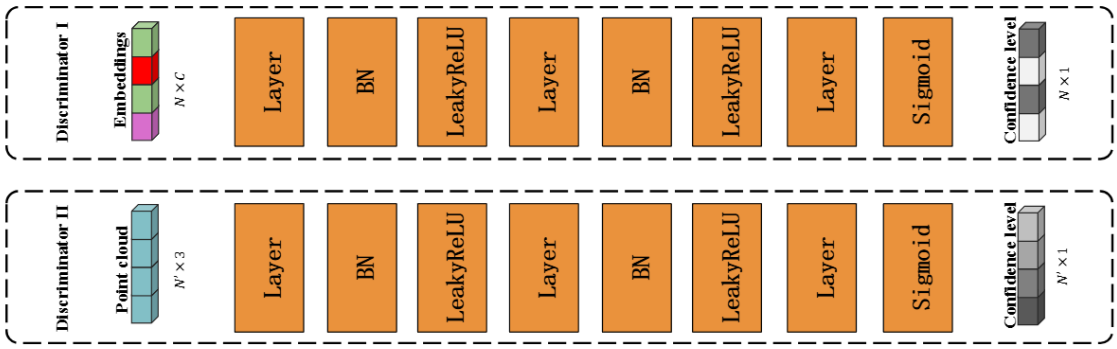

Fig. 6. Architectures of discriminators.

clouds from other objects, which also follow certain patterns in the latent space).

To address issue (2), instead of simply removing noise embeddings identified by the first discriminator, they are mapped back inside the hypersphere through an Inverse Gradient Affine Transformation to generate pseudo-embeddings with normal point semantic features. This correction mechanism synergizes with the self-attention mechanism of the Encoder: the corrected pseudo-embeddings retain the positional information of the original noise points while avoiding outlier interference through regularization constraints in the latent space. To further enhance reconstruction robustness, the algorithm employs an oversaturation point generation strategy during the first reconstruction process, generating an oversaturated number of skeleton points. A second discriminator is then used to implement confidence screening, achieving a dynamic balance between feature preservation and noise suppression.

2.3. Architectures of discriminators

As illustrated in Figure 6, the architectures $\mathcal{D}_1(\cdot)$ and $\mathcal{D}_2(\cdot)$ of both discriminators are identical in structure, differing only in the number of convolutional kernels within their fully connected layers.

3. EXPERIMENT DETAILS

3.1. Datasets

ProjectShapeNet-55 and ProjectShapeNet-34 [12] are alternative versions of the ShapeNet-55 and ShapeNet-34 [16], which more closely resemble real-world scenarios with missing point cloud data (characterized by noise and greater sparsity). These datasets share similar settings and possess identical ground truth labels.

In the ShapeNet-55 and ShapeNet-34, incomplete point clouds are generated by directly cropping complete point clouds. In contrast, ProjectShapeNet-55 and ProjectShapeNet-34 are generated through noisy back-projection of depth images from a single viewpoint. For each sample, 16 viewpoints are randomly selected to render depth images, and noisy back-projection is performed to obtain 16 different incomplete point clouds as training inputs. ProjectShapeNet-55 uses all object categories for both training and testing. In contrast, ProjectShapeNet-34 utilizes only 34 visible object categories for training and evaluates the model's performance on both visible and invisible categories.

KITTI [31] ranks among the most authoritative computer vision algorithm evaluation datasets in the autonomous driving domain. It primarily addresses scene tasks in the in-vehicle environment, including stereo vision, optical flow estimation, visual odometry, 3D object detection, and tracking. The data is collected from real-world urban, rural, and highway scenarios, encompassing dynamic objects (such as vehicles, walkers, and cyclists) under complex traffic conditions, along with multi-sensor fusion data. We evaluate the model's performance on the KITTI dataset, utilizing incomplete point clouds of cars in real-world scenarios scanned by LiDAR.

3.2. Evaluation Metrics

For the ProjectedShapeNet-55 and ProjectedShapeNet-34 datasets, the evaluation metrics are similar to those used for ShapeNet-55/34. Specifically, the L1 chamfer distance (CD-ℓ_1) and F-score@1% are used for evaluation. Specifically, the CD-ℓ_1 between the predicted point set $\mathcal{P}$ and the ground-truth point set $\mathcal{G}$ is defined as follows:

$$d_{CD}(\mathcal{P},\mathcal{G}) = \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}}\min_{g\in\mathcal{G}}\|p-g\| + \frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}}\min_{p\in\mathcal{P}}\|g-p\| \tag{1}$$

where $\|\cdot\|$ represents the L1-norm. We define the precision and recall of the point cloud completion results at threshold d

Table 1. Comparison on the ProjectedShapeNet-55. CD-ℓ_1-Avg denotes the average CD-$\ell_1 \times 10^3$ across all categories. And additional results for certain categories were displayed.

Method	Table	Airplane	Car	Sofa	Birdcage	Remote	Keyboard	Rocket	**CD-ℓ_1-Avg↓**	**F-score@1%↑**
PCN[32]	14.79	9.07	12.85	17.12	20.38	14.62	13.69	10.98	16.64	0.403
TopNet[33]	14.40	9.85	13.61	16.93	22.00	13.52	11.05	10.45	16.35	0.337
GRNet[1]	12.01	8.30	12.13	14.36	16.52	12.18	9.71	8.58	12.81	0.491
PoinTr[16]	9.97	6.02	10.58	12.11	14.60	9.55	7.61	6.86	10.68	0.615
Snowflake[11]	10.49	6.35	11.20	12.59	15.24	10.07	8.12	7.49	11.34	0.594
AdaPoinTr[12]	8.81	5.18	9.77	10.89	13.27	8.81	6.79	5.58	9.58	**0.701**
Ours	**8.70**	**5.16**	**8.61**	**10.55**	**11.88**	**8.27**	**5.90**	**5.37**	**8.85**	0.689

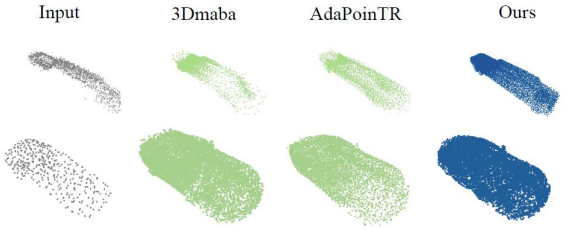

Fig. 7. More qualitative results on point cloud completion at KITTI. The pre-trained model weights are also made available in our open-source code.

(set to 1% by following previous works) as:

$$P(d) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \left[\min_{g \in \mathcal{G}} \|p - g\| < d \right] \tag{2}$$

$$R(d) = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \left[\min_{p \in \mathcal{P}} \|g - p\| < d \right] \tag{3}$$

$$F\text{-}score(d) = \frac{2P(d) \cdot R(d)}{P(d) + R(d)} \tag{4}$$

where $P(d)$ and $R(d)$ denote the precision and recall.

For the KITTI dataset, the evaluation metrics [11] include minimum matching distance (MMD) and fidelity. MMD represents the chamfer distance between the closest output point cloud and the ground truth point cloud, which measures the similarity between the former and the latter. Fidelity refers to the average distance from each point in the input point cloud to its nearest neighbor in the output point cloud. This metric is used to evaluate the extent to which the input information is preserved.

3.3. Implementation Detail

In the model training process, two NVIDIA RTX 3090 graphics cards are used to train the model separately on the ProjectedShapeNet-55 and ProjectedShapeNet-34 datasets for evaluation on these two datasets. For the KITTI dataset, a model pre-trained on ProjectedShapeNet-55 is fine-tuned on the KITTI dataset for 20 epochs for evaluation. The initial learning rate is set to 10^{-4}. An AdamW optimizer with a weight decay of 5×10^{-3} is selected. The weight coefficient is initially set to 10^{-2} and is multiplied by ten every 100 training epochs until it reaches 1. The total number of training epochs is 400. The initial number of point embeddings in the SPA-net encoder is $M_c = 512$. If the saturation rate a is set to $\alpha = 0.25$, then the number of point embeddings in the encoder is $512 \times (1 + 0.25) = 640$.

Table 2. Comparison on the ProjectedShapeNet-34.

Method	**34 seen categories**		**21 unseen categories**	
	CD-ℓ_1-Avg↓	F-Score@1%↑	CD-ℓ_1-Avg↓	F-Score@1%↑
PCN[32]	15.53	0.432	21.44	0.307
TopNet[33]	12.96	0.464	15.98	0.358
GRNet[1]	12.41	0.506	15.03	0.439
PoinTr[16]	10.21	0.634	12.43	0.551
Snowflake[11]	10.69	0.616	12.82	0.551
AdaPoinTr[12]	9.12	0.721	11.37	0.642
Ours	**8.71**	**0.723**	**10.49**	**0.665**

3.4. More results

Table 1 and Table 2 present the comparison results between our method and other state-of-the-art approaches on the ProjectedShapeNet-55 and ProjectedShapeNet-34 datasets. It should be noted that the results for PoinTr [16], AdaPoinTr [12], and Snowflake [11] are taken from their original papers. Since PCN [32], TopNet [33], and GRNet [1] do not perform experiments on these datasets in their original papers, we use an existing paper [12] to reproduce their methods and obtain results on the above datasets. Table 3 in manuscript shows the comparison results between our method and other state-of-the-art approaches on the KITTI dataset (all results are from the original papers).

3.5. More ablation experiments

In addition to the hypersphere hypothesis, as shown in Figure 8, the manifold hypothesis is also a common method for representing potential spaces. Therefore, in this section, the

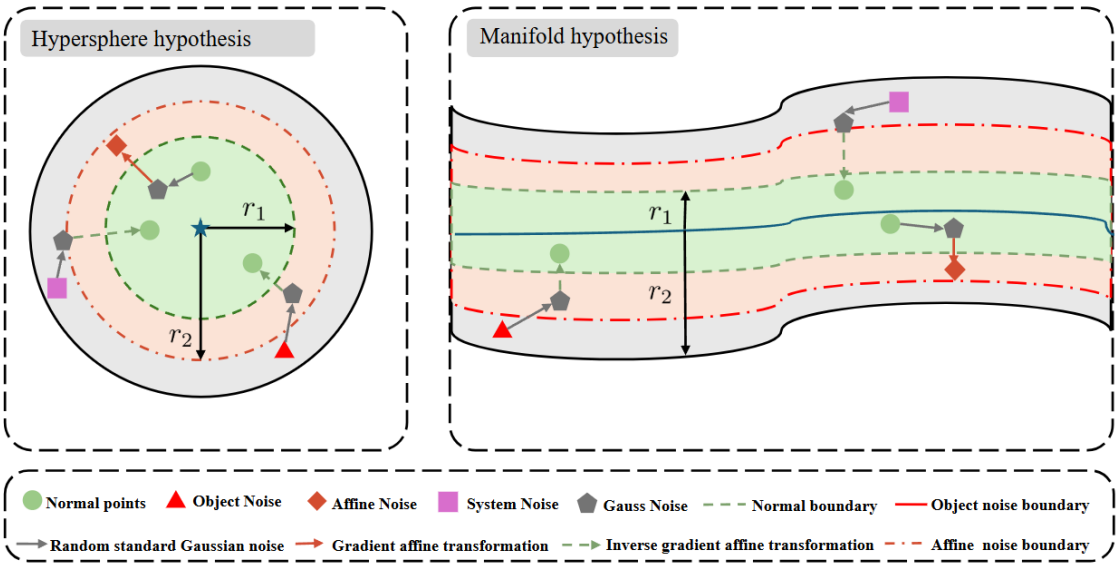

Fig. 8. Potential spatial representations of hypersphere hypothesis and manifold hypothesis. Hypersphere hypothesis. **Hypersphere hypothesis**: Assuming the data distribution is symmetrical and uniform, normal samples are strictly confined within the hypersphere, forming a convex envelope. **Manifold hypothesis**: Assuming high-dimensional data is distributed on a low dimensional manifold, the manifold locally approximates Euclidean space, but the overall structure may be nonlinear, non convex, or fragmented.

Table 3. The hypothesis of latent spatial model is set on ProjectShapeNet-55 to melt experimental results.

r_2/r_1	Hypothesis	CD-L_1-Avg	F-score@1%
1.5	Hypersphere	9.12	0.667
2	Hypersphere	**8.85**	0.689
3	Hypersphere	8.97	0.654
1.5	Manifold	9.37	0.686
2	Manifold	9.19	0.695
3	Manifold	9.10	**0.701**

manifold hypothesis and the ratio of the distance from the boundary of the potential space to the normal boundary r_2/r_1 were studied using the CD-ℓ_1-Avg($\times 10^3$) and F-score@1% metrics, with the results shown in Table 3. Compared to the two hypotheses, the CD-ℓ_1-Avg metric of the hypersphere hypothesis is generally better than that of the manifold hypothesis, but the F-score@1% is slightly better for the manifold hypothesis. This may be because the hypersphere hypothesis has a simpler structure, providing a clear constraint for normal data, making the representation of normal point clouds in the potential space more concentrated and compact. This helps to reduce reconstruction errors, thus performing better on the CD-ℓ_1-Avg metric.

The boundary of the manifold is not as clear as that of the hypersphere. It may be more challenging to distinguish between normal and abnormal data, requiring more complex models to learn and recognize the boundaries of the manifold. However, this more complex boundary, under the premise of proper training, can better capture the complex structure and data distribution within it. The F-score@1% is often used to measure the model's precision at a low recall rate, reflecting the model's ability to correctly classify fine or edge points. Therefore, the manifold hypothesis is superior in recognizing fine details or edge points (F-score@1%). For the ratio of the distance from the affine boundary to the normal boundary r_2/r_1 , its effect on the model's performance is also significant. When this ratio is too small, the boundary may be smaller than the actual normal boundary, leading to an overly sensitive detector that easily misclassifies normal data as abnormal. Conversely, when this ratio is too large, the boundary becomes blurred, which may be beneficial for the approximation of the manifold boundary, as it reduces the restriction on the boundary range, providing more space for the complex structure of the manifold, but is not conducive to the approximation of the spherical boundary. The hypersphere hypothesis relies on the clear boundary to distinguish between normal and abnormal data, and a blurred boundary makes this distinction difficult, affecting the model's performance under the hypersphere hypothesis, especially in scenarios requiring precise boundary definition.

In summary, when using the hypersphere hypothesis and $r_2/r_1 = 2$, the best CD-ℓ_1-Avg($\times 10^3$)) result is 8.85, but the F-score@1% is a secondary result of 0.689. When using the manifold hypothesis and $r_2/r_1 = 3$, the best F-score@1% re-

sult is 0.701, but theCD-ℓ_1-Avg($\times 10^3$)) result is significantly lower than the former. Considering all factors, the hypersphere hypothesis,$r_2/r_1 = 2$ is chosen as the final model setting.

4. REFERENCES

[1] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun, "Grnet: Gridding residual network for dense point cloud completion," in *Computer Vision – ECCV 2020,Lecture Notes in Computer Science*, Jan 2020, p. 365–381.

[2] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari, "Softpoolnet: Shape descriptor for point cloud completion and classification," in *Computer Vision – ECCV 2020,Lecture Notes in Computer Science*, Jan 2020, p. 70–85.

[3] Liang Pan, Xinyi Chen, Zhongang Cai, and .etc, "Variational relational point completion network for robust 3d classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 11340–11351, 2023.

[4] Zhaoyang Lyu, Zhifeng Kong, Xudong Xu, Liang Pan, and Dahua Lin, "A conditional point diffusion-refinement paradigm for 3d point cloud completion," 2022.

[5] Haoqiang Fan, Hao Su, and Leonidas Guibas, "A point set generation network for 3d object reconstruction from a single image," in *2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[6] Tong Wu, Liang Pan, Junzhe Zhang, Tai Wang, Ziwei Liu, and Dahua Lin, "Density-aware chamfer distance as a comprehensive metric for point cloud completion," 2021.

[7] Ruihui Li, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng, "Pu-gan: A point cloud upsampling adversarial network," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 7202–7211.

[8] Xiaogang Wang, Marcelo H. Ang, and Gim Hee Lee, "Cascaded refinement network for point cloud completion," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 787–796.

[9] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le, "Pf-net: Point fractal network for 3d point cloud completion," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," 2023.

[11] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han, "Snowflake point deconvolution for point cloud completion and generation with skip-transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 6320–6338, Apr. 2023.

[12] Xumin Yu, Yongming Rao, Ziyi Wang, Jiwen Lu, and Jie Zhou, "Adapointr: Diverse point cloud completion with adaptive geometry-aware transformers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 12, pp. 14114–14130, 2023.

[13] Haoran Zhou, Yun Cao, Wenqing Chu, Junwei Zhu, Tong Lu, Ying Tai, and Chengjie Wang, "Seedformer: Patch seeds based point cloud completion with upsample transformer," in *Computer Vision – ECCV 2022*, Cham, 2022, pp. 416–432, Springer Nature Switzerland.

[14] Yijun Long, Zhaoyu Chen, Hong Lu, and Wenqiang Zhang, "Gsformer: Geometric-spatial transformer on point cloud completion," in *2023 IEEE International Conference on Multimedia and Expo (ICME)*, 2023, pp. 1175–1180.

[15] Shanshan Li, Pan Gao, Xiaoyang Tan, and Mingqiang Wei, "Proxyformer: Proxy alignment assisted point cloud completion with missing part sensitive transformer," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 9466–9475.

[16] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou, "Pointr: Diverse point cloud completion with geometry-aware transformers," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 12478–12487.

[17] Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu, "Variational relational point completion network," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 8524–8533.

[18] Junshu Tang, Zhijun Gong, Ran Yi, Yuan Xie, and Lizhuang Ma, "Lake-net: Topology-aware point cloud completion by localizing aligned keypoints," 2022.

[19] Binfu Ge, Shengyi Chen, Weibing He, Xiaoyong Qiang, Jingmei Li, Geer Teng, and Fang Huang, "Tree completion net: A novel vegetation point clouds completion model based on deep learning," *Remote Sensing*, vol. 16, no. 20, 2024.

[20] Xinhua Cheng, Nan Zhang, Jiwen Yu, Yinhuai Wang, Ge Li, and Jian Zhang, "Null-space diffusion sampling for zero-shot point cloud completion," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, Aug 2023, p. 618–626.

[21] Junxian Chen, Ying Liu, Yiqi Liang, Dandan Long, Xiaolin He, and Ruihui Li, "Sd-net: Spatially-disentangled point cloud completion network," *Proceedings of the 31st ACM International Conference on Multimedia*, 2023.

[22] Hang Xu, Chen Long, Wenxiao Zhang, Yuan Liu, Zhen Cao, Zhen Dong, and Bisheng Yang, "Explicitly guided information interaction network for cross-modal point cloud completion," 2024.

[23] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[24] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2014, vol. 27.

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020, vol. 33.

[26] Yaolei Qi, Yuting He, Xiaoming Qi, Yuan Zhang, and Guanyu Yang, "Dynamic snake convolution based on topological geometric constraints for tubular structure segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 6070–6079.

[27] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, USA, 2019, KDD '19, p. 2828–2837, Association for Computing Machinery.

[28] Guo Pu, Yifang Men, Yiming Mao, Yuning Jiang, Wei-Ying Ma, and Zhouhui Lian, "Controllable image synthesis with attribute-decomposed gan," in *Pattern Analysis and Machine Intelligence (TPAMI), 2022 IEEE Transactions on*, 2022.

[29] Qiyu Chen, Huiyuan Luo, Chengkan Lv, and Zhengtao Zhang, "A unified anomaly synthesis strategy with gradient ascent for industrial anomaly detection and localization," in *European Conference on Computer Vision*. Springer, 2024, pp. 37–54.

[30] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu, "Pct: Point cloud transformer," *Computational Visual Media*, vol. 7, no. 2, pp. 187–199, Apr. 2021.

[31] Andreas Geiger, Philip Lenz, and Raquel Urtasun, "KITTI vision benchmark suite," 2012.

[32] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert, "Pcn: Point completion network," in *2018 International Conference on 3D Vision (3DV)*, Sep 2018.

[33] Lyne P. Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese, "Topnet: Structural point cloud decoder," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.