Unsupervised Learning of 3D Semantic Keypoints with Mutual Reconstruction

Haocheng Yuan[1], Chen Zhao[2], Shichao Fan[1], Jiaxi Jiang[1] and Jiaqi Yang[1]⋆

[1] Northwestern Polytechnical University, China
{hcyuan, fsc_smile, jshmjjx}@mail.nwpu.edu.cn,
jqyang@nwpu.edu.cn,
[2] CVLab EPFL, Switzerland
chen.zhao@epfl.ch

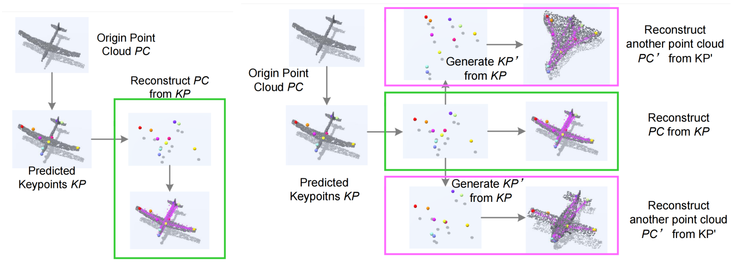
Abstract. Semantic 3D keypoints are category-level semantic consistent points on 3D objects. Detecting 3D semantic keypoints is a foundation for a number of 3D vision tasks but remains challenging, due to the ambiguity of semantic information, especially when the objects are represented by unordered 3D point clouds. Existing unsupervised methods tend to generate category-level keypoints in implicit manners, making it difficult to extract high-level information, such as semantic labels and topology. From a novel mutual reconstruction perspective, we present an unsupervised method to generate consistent semantic keypoints from point clouds explicitly. To achieve this, the proposed model predicts keypoints that not only reconstruct the object itself but also reconstruct other instances in the same category. To the best of our knowledge, the proposed method is the first to mine 3D semantic consistent keypoints from a mutual reconstruction view. Experiments under various evaluation metrics as well as comparisons with the state-of-the-arts demonstrate the efficacy of our new solution to mining semantic consistent keypoints with mutual reconstruction. Our code and pre-trained models are available at https://github.com/YYYYYHC/Learning-Semantic-Keypoints-with-Mutual-Reconstruction.git.

Keywords: Keypoint detection, 3D point cloud, unsupervised learning, reconstruction

1 Introduction

3D semantic keypoints generally refer to representative points on 3D objects, which possess category-level semantic consistency through categories. Detecting 3D semantic keypoints has a broad application scenarios, such as 3D registration [26], 3D reconstruction [15], shape abstraction [25] and deformation [10]. However, this task is quite challenging because of unknown shape variation among different instances in a category, unordered point cloud representations, and limited data annotations.

⋆ Corresponding author


Origin Point Cloud *PC*
Reconstruct *PC* from *KP*
Predicted Keypoints *KP*
Origin Point Cloud *PC*
Generate *KP'* from *KP*
Reconstruct another point cloud *PC'* from KP'
Reconstruct *PC* from *KP*
Predicted Keypoitns *KP*
Generate *KP'* from *KP*
Reconstruct another point cloud *PC'* from KP'

(a) Previous Methods (b) Our Method

Fig. 1: **Comparison of our method and previous methods**. Previous methods focus on self-reconstruction, which may fail to mine category-level semantic consistency information. We address this issue with mutual reconstruction (the keypoints of an object also reconstruct other objects in the same category).

From the technical view, 3D keypoint detection can be divided into geometry-only [21,36] and learning-based [10,12,18,9,5]. For geometry-only ones, they generally leverage shape attributes such as normals to detect distinctive and repeatable points, however, they generally fail to mine semantic information. Learning-based methods can learn semantics from massive training data and can be further classified into supervised and unsupervised. As illustrated in previous works [10,18], supervised methods may suffer from limited human annotated data [32], which greatly limits their applicability. Unsupervised learning of 3D semantic keypoints [18,12,10], however, is particularly challenging due to the ambiguity of semantic information when labels are not given. A few trails have been made toward this line, and we divide these unsupervised methods into two classes by examining if the method employs category-level information explicitly or implicitly. 1) Implicit methods focus on self-related tasks of a single object, such as self-reconstruction [18,5], where keypoints of each object are optimized to reconstruct the original object; category-level information is ensured in an indirect way, as all objects in a specific category are fed into the model during the training process. 2) There are only a few explicit methods [10,9], which consider category information directly. The networks are usually driven by losses of specific tasks involving more than one object from a category. Both explicit and implicit methods have made great success in terms of geometric consistency and robustness, but still fail to ensure semantic consistency. For the implicit methods [18,5], this is caused by a lack of semantic information, as they only consider a single object in a whole category, e.g., reconstructing the object itself based on its own keypoints. As for explicit methods [10,9], although category-level information are taken into consideration explicitly, they still tend to pursue consistency and fail to mine the hidden semantic information within keypoints.

To this end, from a novel mutual reconstruction perspective, we propose an unsupervised method to learn category-level semantic keypoints from point clouds. *We believe that semantic consistent keypoints of an object should be able to reconstruct itself as well as other objects from the same category.* The motivation behind is to fully leverage category-level semantic information and ensure the consistency based on an explicit manner. Compared with deformation tasks [10] based on cage deformation methods, shape reconstruction from keypoints have been well investigated [18,5] and is more straightforward and simpler. In particular, only reconstruction task is involved in our model. The overall technique pipeline of our method is as follows. First, given two point clouds of the same category, keypoints are extracted by an encoder; second, the source keypoint set is reshaped according to the offset of input point clouds; then, source and reshaped keypoint sets are used as the guidance for self-reconstruction and mutual reconstruction with a decoder [18]; finally, both self-reconstruction and mutual reconstruction losses are considered to train the network. Experimental results on KeypointNet [32] and ShapeNet Part [3] have shown that the proposed model outperforms the state-of-the-arts on human annotation datasets. It can be also generalized to real-world scanned data [6] without human annotations.

Overall, our method has two key contributions:

- To the best of our knowledge, we are the first to mine semantic consistency with mutual reconstruction, which is a simple yet effective way to detect consistent 3D semantic keypoints.
- We propose a network to ensure keypoints performing both self reconstruction and mutual reconstruction. It achieves the overall best performance under several evaluation metrics on KeypointNet [32] and ShapeNet Part [3] datasets.

2 Related Work

This section first gives a review on unsupervised semantic keypoints and geometric keypoint detection. Supervised methods are not included, since the task of 3D keypoint detection is seldomly accomplished in a supervised way due to the lack of sufficient labelled datasets. Then, a recap on deep learning on point clouds is given.

2.1 Unsupervised Semantic Keypoint Detection

We divide current methods into two classes according to if the category-level information is leveraged implicitly or explicitly.

Implicit methods. Implicit methods employ self-related metrics to measure the quality of keypoints. A typical implicit method is skeleton merger [18], whose key idea is to reconstruct skeleton-liked objects based on its keypoints through an encoder-decoder architecture. Another implicit way [5] utilizes a convex combination of local points to generate local semantic keypoints, which are then measured by how close they are to the origin point cloud. Unsupervised stable interest point (USIP) [12] predicts keypoints with a Siamese architecture, and the two inputs are two partial views from a 3D object. Implicit methods can achieve good spatial consistency and are relatively light-weight. However, they generally fail to mine semantic consistency information.

Explicit methods. Explicit methods cope with category-level information directly. Keypoint deformer [10] employs a Siamese architecture for shape deformation to detect shape control keypoints; the difference between two input shapes is analysed by comparing their keypoint sets. The cage [31] method is crucial to keypoint deformer [10]; to deform a point cloud, cage [31] takes the origin point cloud, shape control points on point cloud, and target cage as input, the output of cage consists of a deformed cage and a deformed point cloud under the constraint of cage. Another explicit method [9] learns both category-specific shape basis and instance-specific parameters such as rotations and coefficients during training; however, the method requires a symmetric hypothesis. Different from the two previous works, our method evaluates keypoints from the self and mutual reconstruction quality by estimated keypoints and do not require additional hypotheses on inputs.

2.2 Geometric Keypoint Detection

Besides semantic keypoints, detection of geometric keypoints has been well investigated in previous works [21,36]. Different from semantic keypoints that focus on category-level semantic consistency, geometric keypoints are defined to be repeatable and distinctive keypoints on 3D surfaces. In a survey on geometric keypoints, Tombari et al. [23] divided 3D geometric detectors into two categories, i.e., fixed-scale and adaptive-scale. Fixed-scale detectors, such as LSP [4], ISS [36], KPQ [14] and HKS [22], find distinctive keypoints at a specific scale with a non-maxima suppression (NMS) procedure, which is measured by saliency. Differently, adaptive-scale detectors such as LBSS [24] and MeshDoG [35] first build a scale-space defined on the surface, and then pick up distinctive keypoints with an NMS of the saliency at the characteristic scale of each point. Geometric keypoints focus on repeatable and distinctive keypoints rather than semantically consistent keypoints.

2.3 Deep Learning on Point Clouds

Because our method relies on reconstruction, which is typically performed with an encoder-decoder network on point clouds. We will briefly discuss deep learning methods from the perspectives of encoder and decoder.

Encoder. A number of neural networks have been proposed, e.g., PointNet [16], PointNet++ [17], and PointConv [28], which directly consume 3D point clouds. PointNet [16] is a pioneering work, which extracts features from point clouds with point-wise MLPs and permutation-invariant functions. Based on PointNet,

PointNet++ [17] introduces a hierarchical structure to consider both local and global features; PointNet is applied after several sampling and grouping layers; PointNet++ is also employed as an encoder by several unsupervised 3D semantic keypoint detection methods [18,5]. More recent point cloud encoders include [20,28,27]. These encoders have achieved success in tasks like registration [26] and reconstruction [15]. Several keypoint detection methods[10,18,5] also employ PointNet++ [17] as the encoder.

Decoder. In previous point cloud learning works [1,33], MLP is frequently leveraged to generate point clouds from the encoded features. Specifically, FoldingNet [30] proposes a folding-based decoder to deform 2D grid onto 3D object surface of a point cloud. Many works [7,8,29] follow FoldingNet [30] and decode the features based on structure deformation. In [19], tree structure is used to decode structured point clouds. From the functional view, most decoders leveraged by 3D semantic keypoint detection methods [18,12,5,9] focus on reconstructing the original shape of the input. An exception is keypoint deformer [10], whose decoder tries to deform the source shape into the target shape through cage-based deformation.

3 The Proposed Method

The pipeline of our method is shown in Fig. 2. Self reconstruction and mutual reconstruction are performed simultaneously through encoder-decoder architectures.

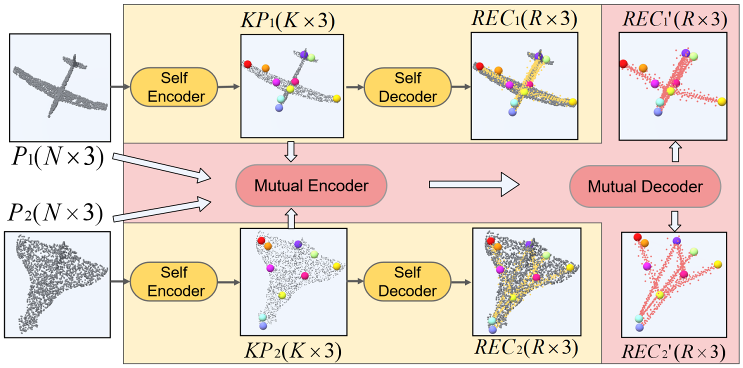


Fig. 2: **Pipeline of our method.** Two input point clouds (each with N points) P_1, P_2 are fed into self and mutual encoders, the outputs are two keypoint sets KP_1, KP_2 and mutual features. Self and mutual decoders then decode the source keypoint set KP_1, KP_2 into REC_1, REC_2 and REC_1', REC_2'. Reconstruction loss is calculated by Chamfer distance between P, REC (self reconstruction) and P, REC' (mutual reconstruction).

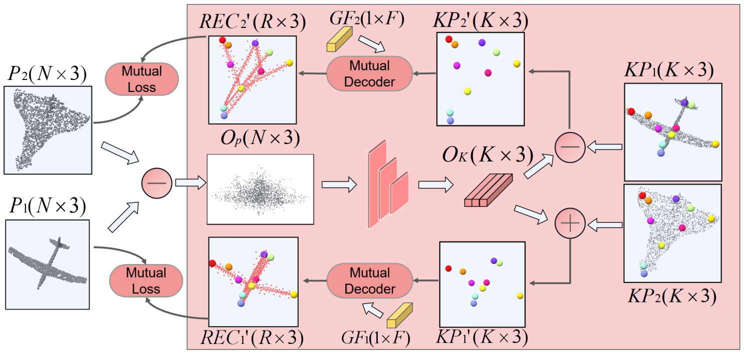


Fig. 3: **Mutual reconstruction process of our method.** Two predicted keypoint sets KP_1 and KP_2 are reshaped into KP_1' and KP_2' with offsets generated between input point clouds PC_1, PC_2. A **shared** decoder from Skeleton Merger [18], then decodes KP_1', KP_2' into REC_1', REC_2'. Mutual reconstruction loss is calculated by Chamfer distance between REC, REC'.

3.1 Self and Mutual Reconstructions

Self and mutual reconstructions are the key components of our method. For an input point cloud P_1, self-reconstruction is supposed to reconstruct the origin point cloud P_1 from its own keypoint set KP_1; mutual reconstruction is to reconstruct another point cloud P_2 with KP_1 and the offset between P_1, P_2.

Mutual reconstruction. Our mutual reconstruction module is depicted in Fig. 3. The mutual reconstruction process utilizes several outputs from self reconstruction, including keypoint sets KP_1, KP_2 and the global feature GF.

Mutual reconstruction is supposed to be able to extract category-level semantic consistent keypoints as illustrated in Fig. 4. The figure illustrates the semantic ambiguity of self-reconstruction, which can be resolved by the mutual reconstruction module. When the method with only self-related tasks (e.g., self reconstruction) predicts object-wise keypoints which are not semantically consistent, it may fail to notice the inconsistency as the topology information is inconsistent as well (we visualize the topology information as a sequence of connection, while some methods employ topology information implicitly); however, the mutual reconstruction model is sensitive when either the topology or semantic label prediction is not correct, as additional shapes are considered in mutual reconstruction and the constraint on keypoint consistency is much tighter.

Self reconstruction. The self reconstruction module is presented in Fig. 5. Specifically, the point-wise feature can also be considered as point-wise score, because the keypoints are actually generated by linear combination of origin points. In other words, for the point with a higher score (feature value), it con-

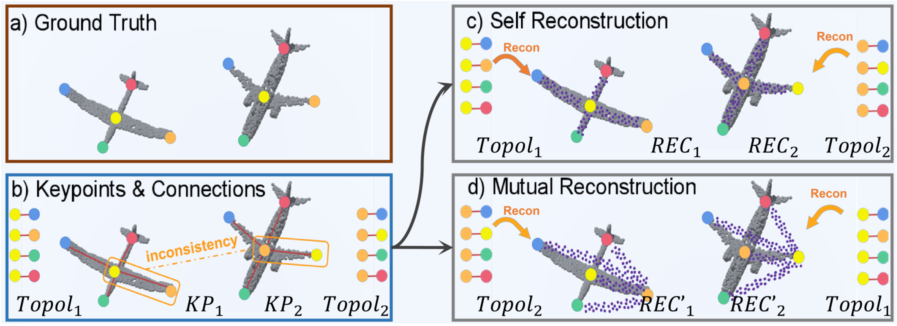


Fig. 4: **Illustration of the difference between self-related tasks and mutual reconstruction.** The purple points are reconstruction results REC, REC'. Loss is computed between the purple reconstruction points and the grey original input points. **a)** Consistent ground truth keypoints. **b)** Inconsistent prediction of the model. Both keypoints and the topology are inconsistent. **c)** For self-reconstruction, REC_1, REC_2 are reconstructed from KP_1, KP_2 separately with topology $Topol_1, Topol_2$. That may cause inconsistency problem as the encoder may learn to predict inconsistent $Topol$ for inconsistent KP. The Chamfer distance loss between a single REC and its original point cloud is low, **despite the inconsistency of predicted keypoints**. **d)** Mutual-reconstruction can alleviate this problem. The mutual reconstruction decoder first reshapes KP_1, KP_2 into KP'_2, KP'_1, which are predictions of KP_2, KP_1. (The visualized keypoints in mutual reconstruction are KP' instead of KP.) Then, the decoder reconstructs REC'_1, REC'_2 based on KP'_1, KP'_2 and $Topol_1, Topol_2$. The chamfer distance between reconstruction and original point cloud would be much greater due to the inconsistent topology.

tributes more to the keypoint prediction. We simply define the point-wise score to be the sum of the k-dim feature, as visualized in Fig. 3 (the airplane in red).

Self reconstruction is also a critical component for mining the semantic information from an instance [18,5]. To ensure category-level semantic consistent information, instance and cross-instance information should be mined, such that self reconstruction is utilized as complementary to mutual reconstruction.

3.2 Network Architecture

The whole pipeline of our method is illustrated in Fig. 2. All **decoders** in self and mutual reconstruction processes are shared, and the only difference between the self and mutual **encoder** is that the mutual one needs to **reshape keypoint set** after the same architecture as the self one. Thus, the core of our network architecture are encoder, reshaping keypoint set and decoder. The three technical modules are detailed in the following.

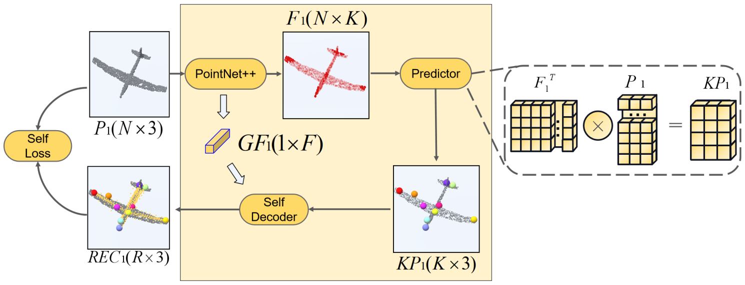


Fig. 5: **Self reconstruction process of our method.** The input point cloud (with N 3D points) P_1 is first fed into a shared PointNet++ encoder, whose output is a group of $K \times N$ point-wise feature F_1, where K indicates the number of keypoints. Keypoint sets KP_1 is calculated by the inner product of F_1^T and P. A shared decoder then reconstructs the source keypoint KP_1 sets into REC_1. Self reconstruction loss is calculated by Chamfer distance between P_1, REC_1. The GF_1 indicates the global feature, which consists of activation strengths and trainable offsets.

Encoder. The designed encoder is supposed to generate keypoints proposals K_1, K_2 from input point clouds P_1, P_2. First, we employ the PointNet++ [17] encoder and it offers a K-dimension point-wise feature for every point in the origin point cloud, thus the shape of point feature matrix F is $K \times N$. Keypoints are calculated by:

$$KP = F \cdot P. \tag{1}$$

Reshape keypoint set. After keypoints proposals KP are generated by the encoder, they are reshaped into new keypoint sets KP', which are utilized by the decoder for mutual reconstruction. We reshape source keypoint set KP with a point-wise offsets O_{kp} as:

$$KP_1' = KP_2 + O_K, \tag{2}$$

and

$$KP_2' = KP_1 - O_K, \tag{3}$$

where O_{kp} is calculated by feeding offsets of origin point clouds O_p into a 3-layers MLP, as in the following:

$$O_K = MLP(O_P), \tag{4}$$

and

$$O_P = P_1 - P_2. \tag{5}$$

The reshaped source keypoints are fed to the decoder for reconstruction.

Decoder. We build our decoder following skeleton merger [18]. The decoder takes keypoint sets KP, KP' and global feature (activation strengths and trainable offsets) as input. It first generates $n(n-1)/2$ line-like skeletons, each of them is composed of a series of points with fixed intervals. Second, trainable offsets are added to every point on the skeleton-like point cloud. Finally, $n(n-1)/2$ activation strengths are applied to the $n(n-1)/2$ skeletons for reconstruction; only skeletons with high activation strengths contribute to the reconstruction process. As such, shapes are reconstructed by decoders.

3.3 Losses and Regularizers

Both self and mutual reconstruction losses are employed to train our model in an unsupervised way.

Reconstruction losses. We calculate reconstruction loss with Composite Chamfer Distance (CCD) [18]. CCD is a modified Chamfer Distance which takes the activation strengths into consideration. For fidelity loss, the CCD between $\hat{X}$ and X is given as:

$$L_f = \sum_i a_i \sum_{\hat{p} \in \hat{X}_i} \min_{p_0 \in X} \|\hat{p} - p_0\|_2, \tag{6}$$

where $\hat{X}_i$ is the i-th skeleton of point cloud $\hat{X}$, and a_i is the activation strength of $\hat{X}_i$. For the coverage loss, there is a change from the fidelity loss that more than one skeleton are considered in the order of how close they are to the given point, until the sum of their activation strengths exceeds 1 [18].

We apply the CCD loss in both self and mutual reconstruction tasks. For self reconstruction, we calculate the CCD between the input target shape P_t and output target shape P'_t:

$$L_{rec_s} = CCD(P_1, REC_1) + CCD(P_2, REC_2). \tag{7}$$

For mutual reconstruction, we calculate the CCD between the input target shape P_t and output source shape P'_s:

$$L_{rec_m} = CCD(P_1, REC'_1) + CCD(P_2, REC'_2). \tag{8}$$

The eventual reconstruction loss is a combination of the two losses:

$$L_{rec} = \lambda_s L_{rec_s} + \lambda_m L_{rec_m}, \tag{9}$$

where λ_s and λ_m are weights to control the contributions of self and mutual reconstructions.

Regularizers. The trainable offsets in our decoder are calculated by multiple MLPs. To keep the locality of every points on the skeleton, we apply an L_2 regularization on them. L_2 regularization is also imposed on the keypoint offset O_K, in order to reduce the geometric changes of keypoints when reconstructing the other shape.

4 Experiments

Experimental setup. In our experiments, we follow [18] and report the dual alignment score (DAS), mean intersection over union (mIoU), part correspondence ratio, and robustness scores of tested methods. We choose learning-based methods including skeleton merger [18], Fernandez et al. [9], USIP [12], and D3Feat [2]; and geometric methods including ISS [36], Harris3D [21], and SIFT3D [13], for a thorough comparison. Note that there is a lack of supervised methods [34] and valid annotated datasets [32] in this field. For this reason, only several unsupervised ones are chosen. We also perform an ablation study, in which we analyze the effectiveness of our mutual-reconstruction module. For training, We employ ShapeNet [3] with the standard split of training and testing data, in which all shapes are normalized into a unit box. For evaluation, we utilize the following datasets, i.e., the human-annotated keypoint dataset KeypointNet [32], a part segmentation dataset named ShapeNet Part [3], and a real-world scanned dataset ScanNet [6].

Implementation details. We randomly split the training dataset into two groups. Mutual reconstruction is performed by respectively taking two shapes from two different groups per time. Point clouds are down-sampled to 2048 points with farthest-point-sampling[17]. The number of keypoints for all categories is restricted to 10. The model is trained on a single NVIDIA GTX 2080Ti GPU, and the Adam [11] is used as the optimizer. We train the KeypointNet[32] for 80 epochs in 8 hours. By default, the weights (λ_s and λ_m) of self reconstruction and mutual reconstruction losses are set to 0.5.

4.1 Semantic Consistency

Semantic consistency means a method can predict keypoints that are of the same semantic information. There are several popular metrics to evaluate semantic consistency, all of which are considered in the experiments for a comprehensive evaluation.

Dual alignment score. We first evaluate the semantic consistency on KeypointNet[32] with DAS, which is introduced by [18]. Given the estimated keypoints on a source point cloud, DAS employs a reference point cloud for keypoint quality evaluation. We predict keypoints with our model on both source and reference point clouds, and use the human annotation on the reference point cloud to align our keypoints with annotated keypoints. The closet keypoint to a human annotation point is considered to be aligned with the annotation. DAS then calculates the ratio of of aligned keypoints between the source and reference point clouds.

The results are shown in Table 1. It can be found that ISS is significantly inferior to others, because it tries to find distinctive and repeatable points rather than points with semantic information. Compared with two recent unsupervised learning methods, our method also surpasses them in most categories.

Table 1: Comparative DAS performance on KeypointNet.

	Airplane	Chair	Car	Table	Bathtub	Guitar	Mug	Cap	Mean
Fernandez et al. [9]	61.4	64.3	–	–	–	–	–	–	62.85
Skeleton Merger [18]	77.7	76.8	**79.4**	70.0	69.2	**63.1**	67.2	53.0	69.55
ISS [36]	13.1	10.7	8.0	16.2	9.2	8.7	11.2	13.1	11.28
Ours	**81.0**	**83.1**	74.0	**78.5**	**71.2**	61.3	**68.2**	**57.1**	**71.8**

Table 2: Comparative mIoU performance on KeypointNet.

	Airplane	Chair	Car	Table	Bed	Skateboard	Mean
Fernandez et al. [9]	69.7	51.2	–	–	–	–	–
Skeleton Merger [18]	**79.4**	68.4	47.8	50.0	**47.2**	40.1	55.48
ISS [36]	36.3	11.6	20.3	24.1	33.7	31.0	26.16
Ours	79.1	**68.9**	**51.7**	**54.1**	45.4	**43.3**	**57.08**

Mean intersection over union. We report the mIoU of predicted keypoints and ground truth ones. The results are shown in Table 2.

As witnessed by the table, our method achieves the best performance on four categories. Note that Fernandez et al. [9] only reported results on the 'Airplane' and 'Chair' categories, while our method still outperforms it on these two categories.

Part correspondence ratio. We also test the mean part correspondence ratio on the ShapeNet Part dataset. This metric is not as strict as DAS and mIoU, because it defines two semantic keypoints as corresponding if they are in the same semantic part of objects in a category. The comparative results are shown in Table 3.

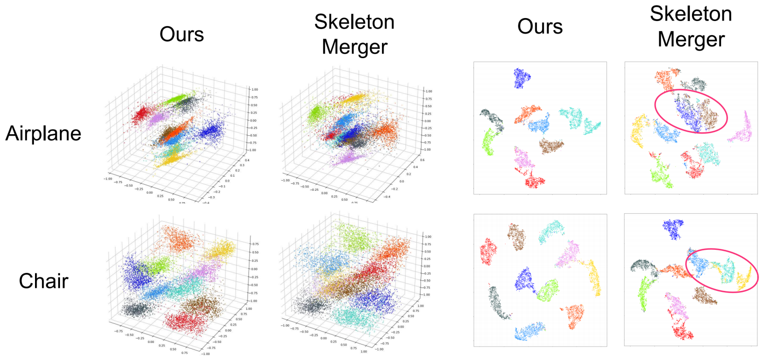
Due the that the part correspondence ratio is a loose metric, the gaps among tested methods are not as dramatic as those in Tables 1 and 2. Remarkably, our method also achieves the best performance under this metric.

Visualization. We first visualize the 3D keypoint distribution and the keypoint features based on t-SNE in Fig. 6, where different colors indicate different semantic labels. Here, we take the skeleton merger method as a comparison.

It can bee seen that our method ensures more consistent alignment of semantic keypoints, as keypoints of the same semantic label tend to be close to each other in the 3D space. Besides, the t-SNE results suggest that our encoder learns more distinctive category-level information from point clouds. Finally, we give a comparative semantic keypoint detetcion results in Fig. 7.

Table 3: Mean correspondence ratio results on ShapeNet part dataset.

	Airplane	Chair	Table	Mean
USIP [12]	77.0	70.2	81.5	76.23
D3Feat [2]	79.9	84.0	79.1	81.00
Harris3D [21]	76.9	70.3	84.2	77.13
ISS [36]	72.2	68.1	83.3	74.53
SIFT3D [13]	73.5	70.9	84.1	76.17
Ours	**81.5**	**85.2**	**85.7**	**84.13**



(a) 3D coordinates distribution

(b) 2D t-SNE feature distribution

Fig. 6: **Distribution of semantic keypoints in the 3D space and keypoint features in the 2D space with t-SNE.** Points with the same semantic label are rendered with the same color.

4.2 Robustness

We test the repeatability of predicted keypoints under Gaussian noise to show the robustness of our method. Specifically, Gaussian noise with different scales are injected to the point cloud, and if the keypoint localization error on noisy point clouds are greater than a distance threshold (0.1 in this experiment), we treat the detected keypoint is not repeatable. The results in shown in Fig. 8.

It suggests that our method holds good robustness to noise, which can be more clearly reflected by the right visualization results in Fig. 8.

We also test the generalization ability on a real-world scanned dataset [6]. We split chairs from the large scene in ScanNet[6] according to the semantic label, and perform random sampling to opt 2048 points from the raw data. Our model is trained on normalized ShapeNet, and tested on the real-world scanned chairs. The result is shown in Fig. 9. One can see that on real-world data, the model can still predict semantic consistent points without re-training.

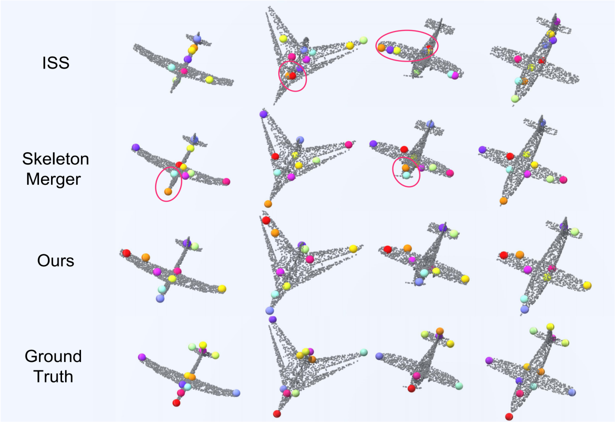
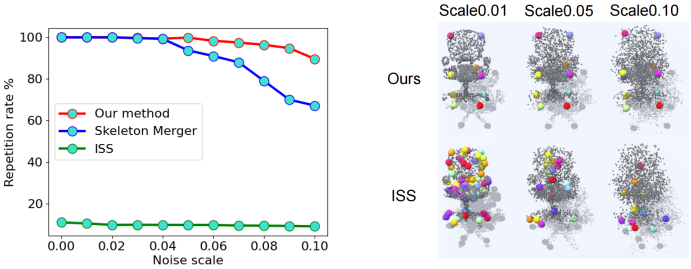


Fig. 7: **Keypoints predicted by different methods.** Keypoints are rendered with different colors to show semantic consistency.



(a) Repeatability curve.

(b) Keypoint visualization under noise.

Fig. 8: **Robustness test.** This experiment is tested on the airplane (Fig. a) and chair (Fig. b) categories from ShapeNet[3].

4.3 Ablation Study

To verify the effectiveness of mutual reconstruction, we compare a variation of our method without mutual reconstruction ('w/o m-rec') with the original method. The results are shown in Table 4.

It can be found that mutual reconstruction can significantly improve the performance as verified by both DAS and mIoU metrics. This clearly verifies the effectiveness of mutual reconstruction for unsupervised 3D semantic keypoint detection.

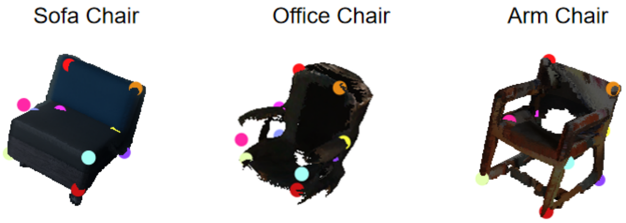


Fig. 9: **Test on a real-world scanned dataset.** Real-world scanned "Chairs" are taken from the ScanNet [6] dataset.

Table 4: Comparison of the full method and the one without mutual reconstruction.

	Airplane	Chair	Car	Table	Mean
Full method (DAS)	**81.0**	**83.1**	**74.0**	**78.5**	**79.15**
w/o m-rec (DAS)	67.2	61.3	60.3	71.2	65.0
Full method (mIoU)	**79.1**	**68.8**	**51.7**	54.1	**62.85**
w/o m-rec (mIoU)	77.2	52.1	48.2	**56.1**	58.4

5 Conclusions

In this paper, we proposed mutual reconstruction for 3D semantic keypoint detection. Compared with previous works, we mine *category-level semantic information* from 3D point clouds from a novel mutual reconstruction view. In particular, we proposed an unsupervised Siamese network, which first encodes input point clouds into keypoint sets, and then decoding the keypoint features to achieve both self and mutual reconstructions. In the experiments, our method delivers outstanding semantic consistency and robustness performance. Ablation study also validates the effectiveness of mutual reconstruction for unsupervised 3D semantic keypoint detection.

Though preserving global information (e.g., topology) well, the designed decoder tends to reconstruct point clouds in a skeleton-like manner, which consists limited local information. In our future work, we expect the mutual reconstruction model to be capable of detecting keypoints capturing both local and global structures.

6 Acknowledgment

This work is supported in part by the National Natural Science Foundation of China (NFSC) under Grant 62002295, Natural Science Basic Research Program of Shaanxi (No. 2021JCW-03).

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: International Conference on Machine Learning. pp. 40–49. PMLR (2018)
2. Bai, X., Luo, Z., Zhou, L., Fu, H., Quan, L., Tai, C.L.: D3feat: Joint learning of dense detection and description of 3d local features. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 6359–6367 (2020)
3. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
4. Chen, H., Bhanu, B.: 3d free-form object recognition in range images using local surface patches. Pattern Recognition Letters **28**(10), 1252–1262 (2007)
5. Chen, N., Liu, L., Cui, Z., Chen, R., Ceylan, D., Tu, C., Wang, W.: Unsupervised learning of intrinsic structural representation points. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 9121–9130 (2020)
6. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 5828–5839 (2017)
7. Deng, H., Birdal, T., Ilic, S.: Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In: European Conference on Computer Vision. pp. 602–618 (2018)
8. Deprelle, T., Groueix, T., Fisher, M., Kim, V., Russell, B., Aubry, M.: Learning elementary structures for 3d shape generation and matching. Advances in Neural Information Processing Systems **32** (2019)
9. Fernandez-Labrador, C., Chhatkuli, A., Paudel, D.P., Guerrero, J.J., Demonceaux, C., Gool, L.V.: Unsupervised learning of category-specific symmetric 3d keypoints from point sets. In: European Conference on Computer Vision. pp. 546–563 (2020)
10. Jakab, T., Tucker, R., Makadia, A., Wu, J., Snavely, N., Kanazawa, A.: Keypointdeformer: Unsupervised 3d keypoint discovery for shape control. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 12783–12792 (2021)
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
12. Li, J., Lee, G.H.: Usip: Unsupervised stable interest point detection from 3d point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 361–370 (2019)
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vsion **60**(2), 91–110 (2004)
14. Mian, A., Bennamoun, M., Owens, R.: On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. International Journal of Computer Vision **89**(2), 348–361 (2010)
15. Novotny, D., Ravi, N., Graham, B., Neverova, N., Vedaldi, A.: C3dpo: Canonical 3d pose networks for non-rigid structure from motion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7688–7697 (2019)
16. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660 (2017)
17. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413 (2017)

18. Shi, R., Xue, Z., You, Y., Lu, C.: Skeleton merger: an unsupervised aligned keypoint detector. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 43–52 (2021)
19. Shu, D.W., Park, S.W., Kwon, J.: 3d point cloud generative adversarial network based on tree structured graph convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3859–3868 (2019)
20. Simonovsky, M., Komodakis, N.: Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3693–3702 (2017)
21. Sipiran, I., Bustos, B.: Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. The Visual Computer **27**(11), 963–976 (2011)
22. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multiscale signature based on heat diffusion. In: Computer Graphics Forum. vol. 28, pp. 1383–1392. Wiley Online Library (2009)
23. Tombari, F., Salti, S., Di Stefano, L.: Performance evaluation of 3d keypoint detectors. International Journal of Computer Vision **102**(1), 198–220 (2013)
24. Tombari, F., Salti, S., Stefano, L.D.: Unique signatures of histograms for local surface description. In: European Conference on Computer Vision. pp. 356–369. Springer (2010)
25. Tulsiani, S., Su, H., Guibas, L.J., Efros, A.A., Malik, J.: Learning shape abstractions by assembling volumetric primitives. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2635–2643 (2017)
26. Wang, H., Guo, J., Yan, D.M., Quan, W., Zhang, X.: Learning 3d keypoint descriptors for non-rigid shape matching. In: European Conference on Computer Vision. pp. 3–19 (2018)
27. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics **38**(5), 1–12 (2019)
28. Wu, W., Qi, Z., Fuxin, L.: Pointconv: Deep convolutional networks on 3d point clouds. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 9621–9630 (2019)
29. Yang, G., Huang, X., Hao, Z., Liu, M.Y., Belongie, S., Hariharan, B.: Pointflow: 3d point cloud generation with continuous normalizing flows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4541–4550 (2019)
30. Yang, Y., Feng, C., Shen, Y., Tian, D.: Foldingnet: Point cloud auto-encoder via deep grid deformation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 206–215 (2018)
31. Yifan, W., Aigerman, N., Kim, V.G., Chaudhuri, S., Sorkine-Hornung, O.: Neural cages for detail-preserving 3d deformations. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 75–83 (2020)
32. You, Y., Lou, Y., Li, C., Cheng, Z., Li, L., Ma, L., Lu, C., Wang, W.: Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 13647–13656 (2020)
33. Yu, L., Li, X., Fu, C.W., Cohen-Or, D., Heng, P.A.: Pu-net: Point cloud upsampling network. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2790–2799 (2018)
34. Yumer, M.E., Chaudhuri, S., Hodgins, J.K., Kara, L.B.: Semantic shape editing using deformation handles. ACM Transactions on Graphics **34**(4), 1–12 (2015)

35. Zaharescu, A., Boyer, E., Varanasi, K., Horaud, R.: Surface feature detection and description with applications to mesh matching. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 373–380. IEEE (2009)
36. Zhong, Y.: Intrinsic shape signatures: A shape descriptor for 3d object recognition. In: International Conference on Computer Vision Workshops. pp. 689–696. IEEE (2009)