SNAKE: Shape-aware Neural 3D Keypoint Field: Appendix

A Network Architecture

Following [8], our implementation is a compilation of PointNet++ [9], 3D UNet [3], positional encoder and implicit surface occupancy decoder. The architecture of the implicit keypoint decoder is designed to be the same as the surface occupancy decoder. The dimensions of the feature embedding Z and Z' are both set to 32, *i.e.*, $C_1 = C_2 = 32$. And each point from a query set is also encoded into a 32-dimensional feature vector. More details can be found in the code we provide.

B Implementation Details

B.1 Training

SNAKE is implemented in PyTorch [7] using the Adam [4] optimizer with a mini-batch size of b on 4 NVIDIA A100 GPUs for el epochs. We use a learning rate of 10^{-4} for the first ef epochs, which is dropped ten times for the remainder. As discussed in Sec. 3.2 (repeatability loss), we perform random rigid transformation T on the input P to generate a second view input TP. Then, we use some data augmentation on TP to increase data diversity by downsampling with a random rate between 0 and 4, and Gaussian noise. Training hyper-parameters on four datasets are provided in Table 1.

In our formulation, occupied points are those on the input surface, and the others are considered all unoccupied, including the points inside the surface. Therefore, we can only use input point clouds to learn the surface occupancy model. Specifically, we randomly sample the positives from the input point cloud. The negatives are randomly sampled in the unit 3D space. Although some of the negatives are indeed on the surface of the object, their number is so limited compared to the whole query sets that they do not affect the training.

Table 1: **Training and testing hyper-parameters.** Sem.=Semantic consistency evaluation, Rep.=Repeatability evaluation, Reg.=Registration evaluation, KeypN.=KeypointNet [12], ModelN.=ModelNet40 [10].

Setting	Training Set	Test Set	N	$H/W/D$	$H_l/W_l/D_l$	U	n	b	ef/el	thr_o	thr_s	λ	J
Sem.	KeypN.	KeypN.	2048	64/64/64	8/8/8	8	500	16	40/60	0.5	0.7	10^{-3}	10
	SMPL [6]	SMPL	2048	64/64/64	8/8/8	8	500	16	20/30	0.5	0.7	10^{-3}	10
Rep.	ModelN.	ModelN.	5000	64/64/64	8/8/8	6	500	16	40/60	0.5	0.7	10^{-3}	10
	3DMatch [13]	Redwood [2]	10000	100/100/100	10/10/10	8	150	6	15/20	0.5	0.7	10^{-3}	10
Reg.	KeypN.	3DMatch	2048	64/64/64	6/6/6	12	500	16	40/60	0.5	0.4	10^{-3}	10

B.2 Testing

For the SMPL dataset, the correspondence between the paired point clouds can be generated by SMPL vertex index. Since the keypoint SNAKE generates may not be in the input point cloud (we enforce the keypoint scatter on the underlying surface of the input), we take the point closest to the generated keypoint in the input as the final keypoint. We use the same strategy on the 3DMatch dataset when performing geometric registration because D3feat [1] predicts descriptors for each point in the input. The testing hyper-parameters are shown in Table 1.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

C Results

C.1 Additional comparison with UKPGAN on keypoint repeatability

Due to the absence of pretrained model on the ModelNet40 and 3DMatch dataset, we do not report the keypoint repeatability of UKPGAN [11] on the main paper. We have tried to train UKPGAN (official implementation) on the ModelNet40 and 3DMatch datasets from scratch but observed divergence under default hyper-parameters. The training always reports NaN losses in early epochs. This instability also implies limitations in implementing the idea of joint reconstruction and keypoint detection with GAN-based methods. As such, we provide a new experiment to compare their repeatability on the KeypointNet dataset, on which the UKPGAN provided a pre-trained model.

Tabke 1 and Table 2 in the main paper show that SNAKE achieves significant gains over UKPGAN in most cases. Interestingly, when the inputs are disturbed, the performance of UKPGAN increases rather than decreases. Via visualizing the results in Fig. 1, we find that when the input point clouds are disturbed, the keypoints predicted by UKPGAN are clustered in a small area, which improves the repeatability of keypoints but fails to cover the input uniformly. This illustrates that the GAN-based method adopted by UKPGAN to control the keypoint sparsity is not robust to input point cloud disturbance. The keypoints of ours still remain meaningful under the drastic changes of inputs.

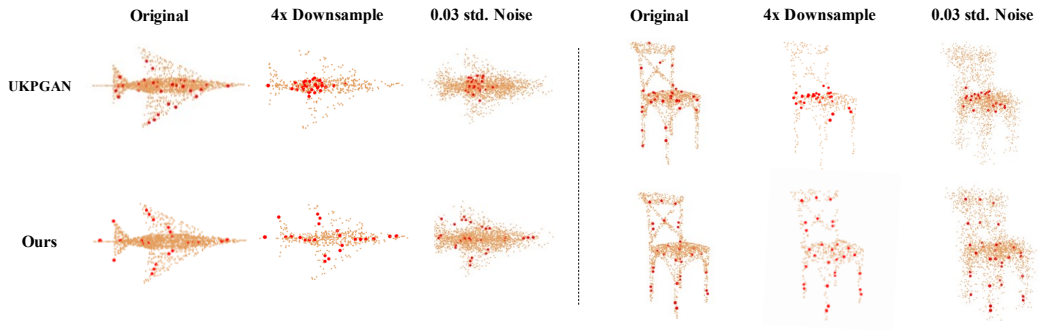

Figure 1: Keypoints of the KeypointNet data under some input disturbances.

C.2 Quantitative Results

The specific numerical results on semantic consistency and repeatability are summarized in Table 3- 9, which correspond to Figure 5 in the main paper. We present the mean and standard deviation of our results over 6 models trained under different random seeds.

C.3 Qualitative Visualization of Saliency Field and Keypoints

We show more qualitative results on keypoint semantic consistency between intra-class instances from rigid objects plane, guitar, motorcycle, and deformable human shapes in Figure 2- 5. Owing to entangling shape reconstruction and keypoint detection, SNAKE can extract aligned representation for intra-class instances. As shown in Figure 6- 11, we provide more visualizations of keypoints under some disturbances on object-level (ModelNet40) and scene-level (Redwood) datasets. It can be seen that SNAKE can generate more consistent keypoints than other methods under significant variations of inputs. We also show the detected keypoints of the same object/scene from different views to demonstrate the repeatability of keypoint in Figure 12- 14.

C.4 Qualitative Visualization of Surface Occupancy Field and Shape Reconstruction

As shown in Figure 15, we show visualizations of the occupancy field and shape reconstruction on the ModelNet40 dataset. These five samples are taken from the unseen test set. As shown by the second row, only points on the input surface have a high occupancy value, and the other points (inside or outside of the surface) have a near-zero occupancy value. Under our definition, two surfaces can be obtained through the marching cube, and we only show the outer surface.

D Computation Cost

As shown in Table 2, we report the time taken to generate keypoints of hand-crafted detector ISS, deep-learning (DL) based methods USIP [5], UKPGAN [11] and ours. ISS [14] is implemented by Open3d [15] and deployed on an AMD EPYC 7742 64-Core CPU. DL-based methods are deployed on an NVIDIA GeForce RTX 3090 GPU. USIP requires the lowest computational time to generate keypoints, while UKPGAN requires the highest cost since it takes much time to compute smoothed density values. The inference time of our model is comparable to ISS when we do not refine the keypoint by optimization (J=0), and the repeatability is still as high as around 81% when the input point number is 4096. The time increases with the increasing number of optimization iterations J. As discussed before, when J becomes larger (below 15), the performance of keypoint gets better. It suggests that there is a trade-off between keypoint performance and inference speed of our method. The GPU memory cost (MB) for USIP, UKPGAN, and SNAKE during a single batch inference is 3747, 10727, and 2785, which illustrates that SNAKE requires the lowest GPU memory cost to generate keypoints.

Table 2: Average time (s) taken to compute keypoints from input point clouds on ModelNet40 dataset. The hyper-parameters of ours can be found in the Table 1. Decimals in parentheses in italics are relative repeatability (%). Here, the experiment setting is the same as in Sec. 4.2.

Input Point #	ISS	USIP	UKPGAN	Ours J=0	Ours J=5	Ours J=10
2048	0.07 (*0.088*)	0.006 (*0.748*)	14.41	0.08 (*0.795*)	0.50 (*0.835*)	0.81 (*0.851*)
4096	0.11 (*0.096*)	0.007 (*0.799*)	36.80	0.09 (*0.811*)	0.50 (*0.850*)	0.83 (*0.864*)

E Illustrations on the Assets We Used and Released

The license of assets we used is as follows: (a) MIT License for KeypointNet dataset. (b) Software Copyright License for non-commercial scientific research purposes on SMPL-Model. (c) GPL-3.0 License for ModelNet40, 3DMatch, Redwood dataset, and USIP. (d) Microsoft research license for 3DMatch registration benchmark.

All existing datasets and codes we used in this paper are allowed for research and do not contain personally identifiable information or offensive content. Note that SMPL only has human shapes without the identity information of the person, such as the face or body texture. Our code is released under the MIT license.

Table 3: mIoU (%) with different geodesic distance thresholds on the KeypointNet dataset. This table corresponds to Figure 5-(a) in the main paper.

	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
Random	0.005	0.010	0.017	0.020	0.023	0.026	0.028	0.032	0.036	0.042	0.049
ISS	**0.008**	**0.012**	0.024	**0.040**	**0.060**	0.088	0.121	0.160	0.198	0.242	0.286
SIFT3D	0.005	0.010	0.015	0.022	0.043	0.065	0.089	0.120	0.160	0.189	0.221
Harris3D	0.005	0.010	0.014	0.023	0.040	0.060	0.084	0.110	0.150	0.180	0.216
USIP	0.003	0.006	0.013	0.024	0.045	0.078	0.116	0.160	0.212	0.264	0.314
UKPGAN	0.005	0.009	0.021	0.036	0.059	0.084	0.114	0.147	0.179	0.207	0.238
Ours	0.006±0.000	**0.012±0.000**	**0.025±0.001**	0.039±0.001	0.058±0.001	**0.091±0.002**	**0.144±0.005**	**0.214±0.005**	**0.291±0.005**	**0.361±0.002**	**0.412±0.002**

Table 4: mIoU (%) with different Euclidean distance thresholds on SMPL mesh. This table corresponds to Figure 5-(e) in the main paper.

	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
Random	0.008	0.011	0.015	0.021	0.038	0.056	0.075	0.103	0.136	0.161	0.195
ISS	**0.078**	**0.095**	**0.101**	0.113	0.129	0.148	0.174	0.206	0.231	0.258	0.293
SIFT3D	0.009	0.011	0.016	0.026	0.043	0.064	0.084	0.108	0.146	0.183	0.213
Harris3D	0.012	0.013	0.016	0.021	0.032	0.047	0.065	0.097	0.129	0.159	0.187
USIP	0.037	0.043	0.051	0.081	0.129	0.198	0.278	0.338	0.390	0.440	0.492
UKPGAN	0.036	0.041	0.059	0.085	0.126	0.171	0.235	0.302	0.369	0.424	0.476
Ours	0.063±0.018	0.079±0.019	0.094±0.023	**0.128±0.028**	**0.182±0.036**	**0.255±0.041**	**0.355±0.041**	**0.457±0.046**	**0.557±0.043**	**0.639±0.037**	**0.704±0.036**

Table 5: Relative repeatability (%) with different distance thresholds on the ModelNet40 dataset. This table corresponds to Figure 5-(b) in the main paper.

	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
Random	0.056	0.094	0.14	0.191	0.249	0.308	0.368	0.429
ISS	0.058	0.096	0.14	0.192	0.247	0.306	0.367	0.427
SIFT3D	0.055	0.092	0.138	0.191	0.249	0.308	0.369	0.429
Harris3D	0.056	0.096	0.147	0.21	0.277	0.347	0.415	0.48
USIP	**0.771**	0.799	0.815	0.827	0.836	0.844	0.851	0.857
Ours	0.763±0.011	**0.864±0.009**	**0.897±0.007**	**0.910±0.005**	**0.917±0.005**	**0.923±0.005**	**0.927±0.005**	**0.930±0.005**

Table 6: Relative repeatability (%) when the input is randomly downsampled by some rates on the ModelNet40 dataset. This table corresponds to Figure 5-(c) in the main paper.

	1	2	4	8	16
Random	0.094	0.093	0.093	0.091	0.092
ISS	0.096	0.088	0.088	0.083	0.076
SIFT3D	0.092	0.089	0.087	0.082	0.075
Harris3D	0.096	0.093	0.093	0.093	0.092
USIP	0.799	0.748	0.685	0.554	0.321
Ours	**0.864±0.009**	**0.851±0.009**	**0.820±0.008**	**0.730±0.009**	**0.528±0.012**

Table 7: Relative repeatability (%) when the input is disturbed by Gaussian noise $N(0, \sigma)$ on the ModelNet40 dataset. This table corresponds to Figure 5-(d) in the main paper.

	0.00	0.02	0.04	0.06	0.08	0.10	0.12
Random	0.094	0.062	0.038	0.027	0.021	0.016	0.014
ISS	0.096	0.061	0.037	0.025	0.02	0.016	0.015
SIFT3D	0.092	0.06	0.036	0.025	0.019	0.016	0.014
Harris3D	0.096	0.063	0.038	0.029	0.02	0.015	0.015
USIP	0.799	**0.872**	**0.844**	0.746	0.558	0.341	0.192
Ours	**0.864±0.009**	0.869±0.008	0.841±0.015	**0.766±0.013**	**0.619±0.041**	**0.464±0.049**	**0.354±0.045**

Table 8: Relative repeatability (%) with the different distance thresholds (m) on the Redwood dataset. This table corresponds to Figure 5-(f) in the main paper.

	0.1	0.12	0.14	0.16	0.18	0.2	0.22	0.24
Random	0.09	0.126	0.163	0.204	0.246	0.287	0.326	0.362
ISS	0.087	0.119	0.156	0.191	0.228	0.264	0.301	0.336
SIFT3D	0.088	0.123	0.168	0.21	0.254	0.297	0.33	0.367
Harris3D	0.079	0.109	0.14	0.175	0.209	0.243	0.278	0.31
USIP	**0.255**	**0.285**	**0.314**	**0.342**	**0.368**	0.392	0.417	0.439
Ours	0.205±0.005	0.246±0.007	0.286±0.008	0.323±0.008	0.359±0.009	**0.393±0.010**	**0.425±0.010**	**0.454±0.009**

Table 9: Relative repeatability (%) when the input is randomly downsampled by some rates on the Redwood dataset. This table corresponds to Figure 5-(g) in the main paper.

	1	2	4	8	16
Random	0.287	0.289	0.291	0.292	0.287
ISS	0.264	0.277	0.158	0.067	0.021
SIFT3D	0.297	0.286	0.28	0.271	0.22
Harris3D	0.243	0.288	0.285	0.292	0.286
USIP	0.392	0.388	0.377	0.351	0.313
Ours	**0.393±0.010**	**0.394±0.008**	**0.391±0.009**	**0.381±0.008**	**0.362±0.007**

Table 10: Relative repeatability (%) when the input is disturbed by Gaussian noise $N(0, \sigma)$ on the Redwood dataset. This table corresponds to Figure 5-(h) in the main paper.

	0.00	0.02	0.04	0.06	0.08	0.10
Random	0.287	0.289	0.275	0.252	0.23	0.21
ISS	0.264	0.26	0.268	0.259	0.25	0.214
SIFT3D	0.297	0.289	0.27	0.261	0.241	0.217
Harris3D	0.243	0.239	0.225	0.206	0.193	0.178
USIP	0.392	0.386	0.375	0.341	0.317	**0.295**
Ours	**0.393±0.010**	**0.392±0.008**	**0.381±0.009**	**0.359±0.009**	**0.318±0.007**	0.256±0.013

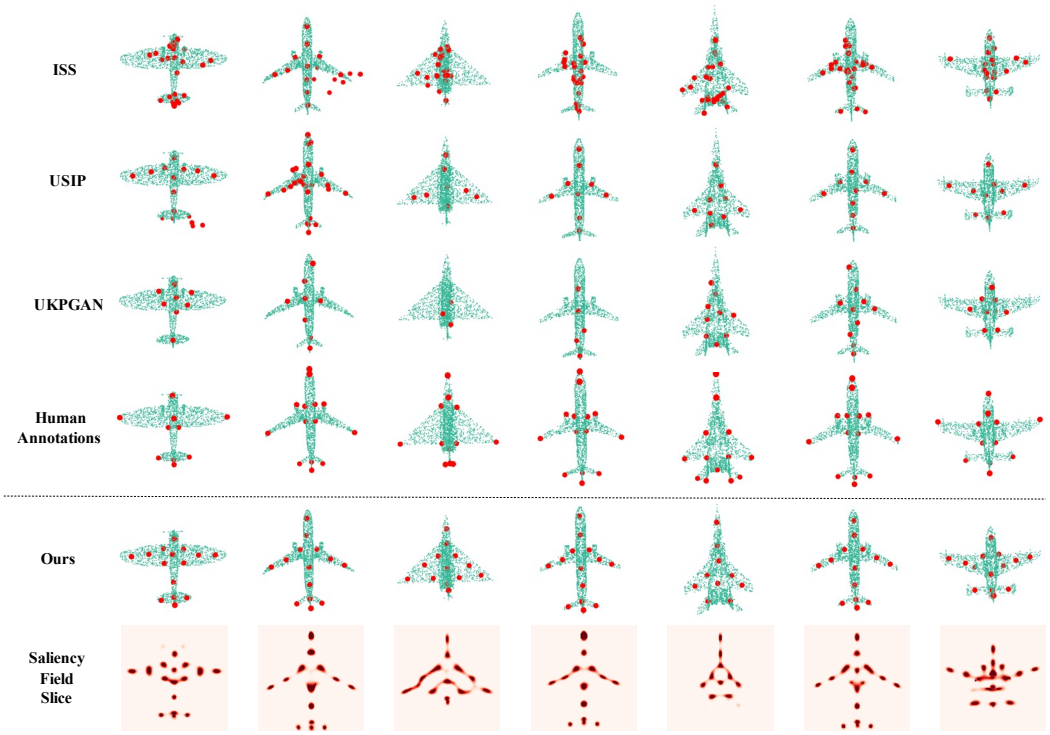

Figure 2: Keypoint semantic consistency comparison on the plane.

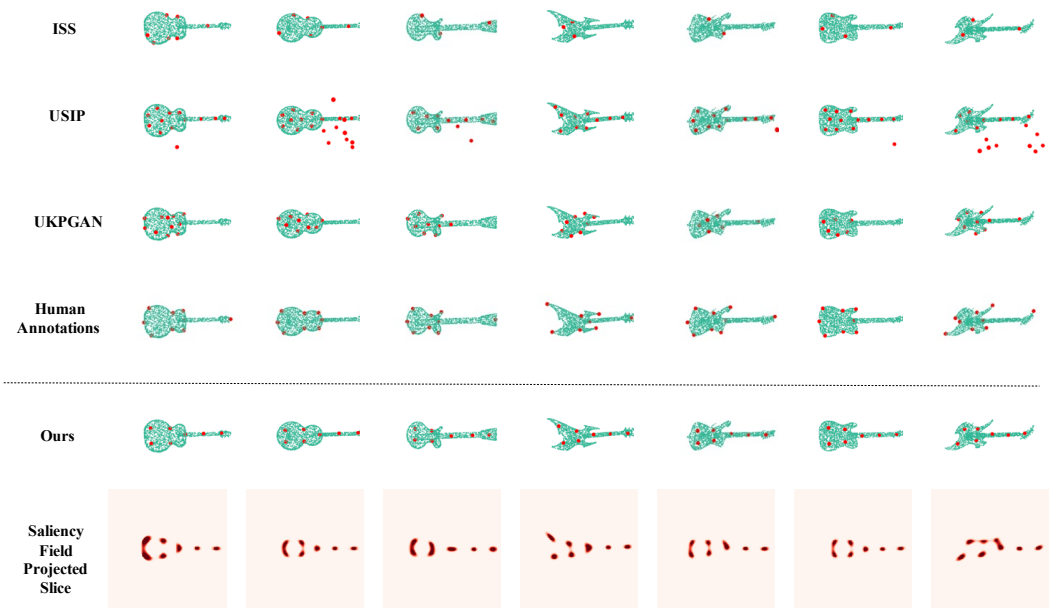

Figure 3: Keypoint semantic consistency comparison on the guitar.

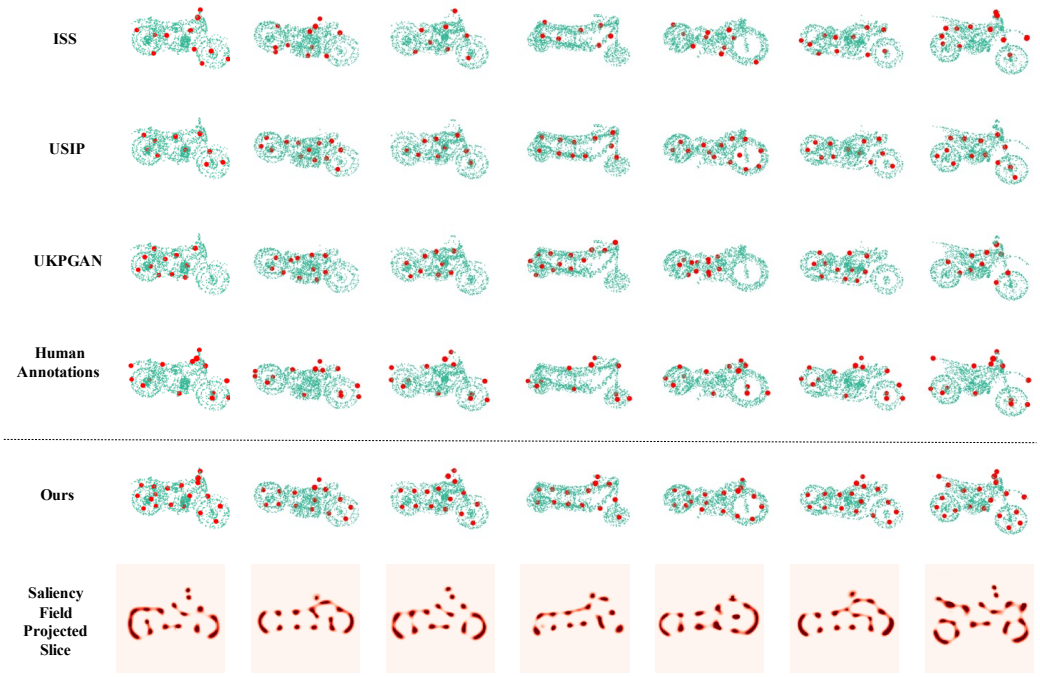

Figure 4: Keypoint semantic consistency comparison on the motorcycle.

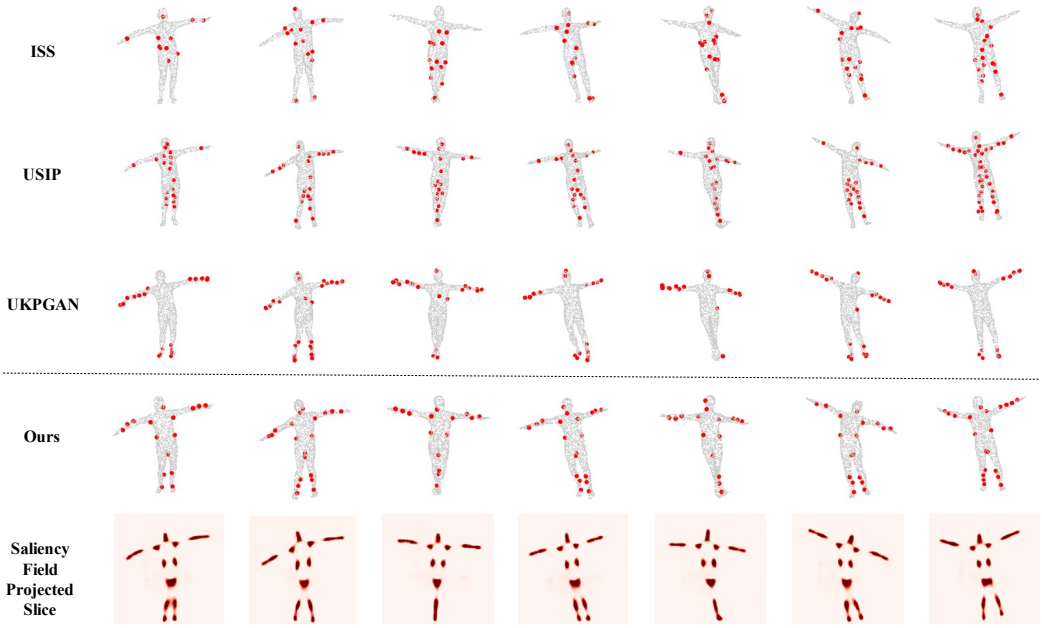

Figure 5: Keypoint semantic consistency comparison on the human shape.

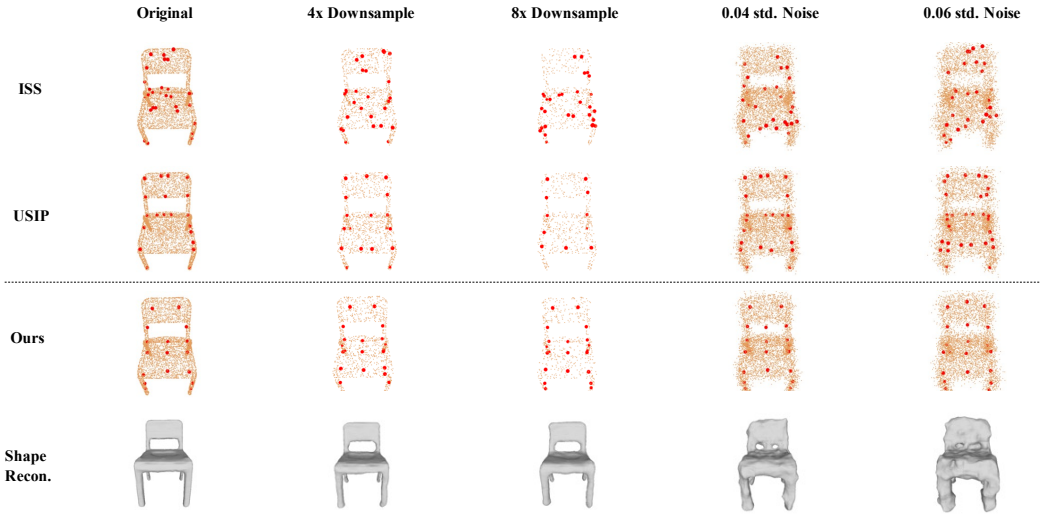

Figure 6: Keypoints of the chair under some input disturbances.

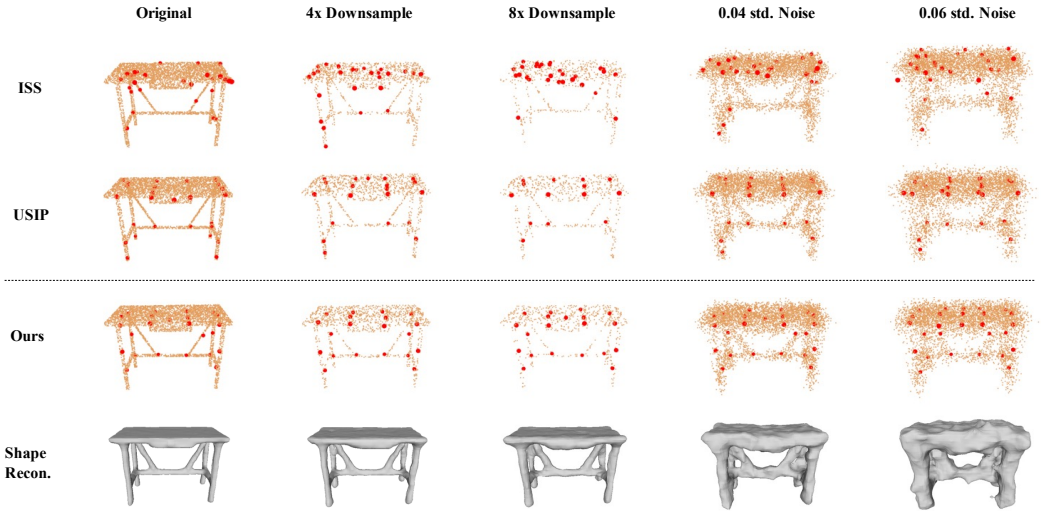

Figure 7: Keypoints of the desk under some input disturbances.

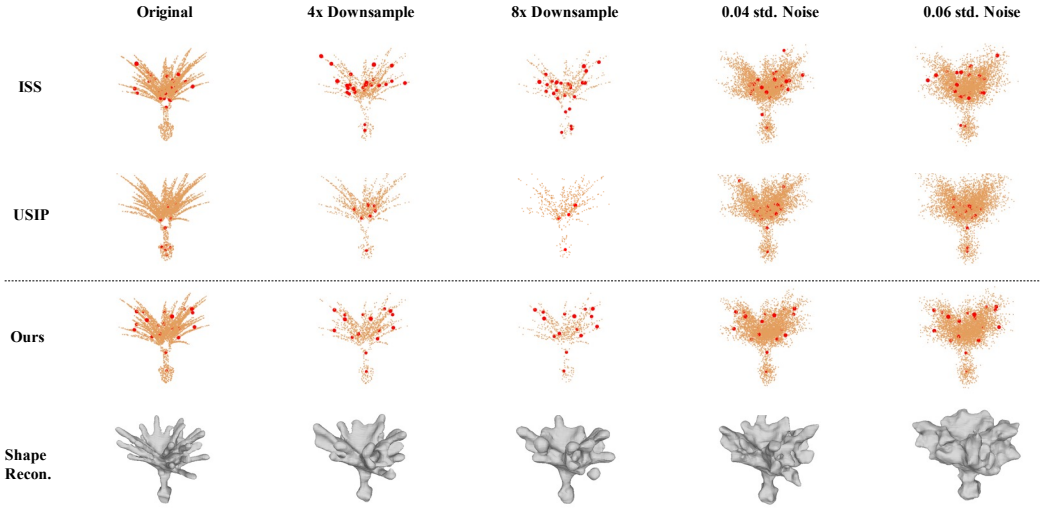

Figure 8: Keypoints of the flower under some input disturbances.

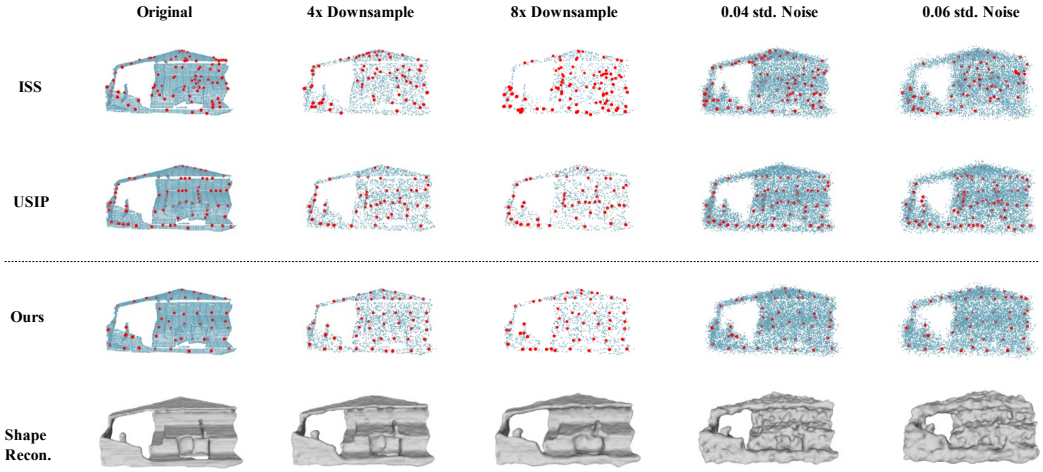

Figure 9: Keypoints of the indoor scene (1) under some input disturbances.

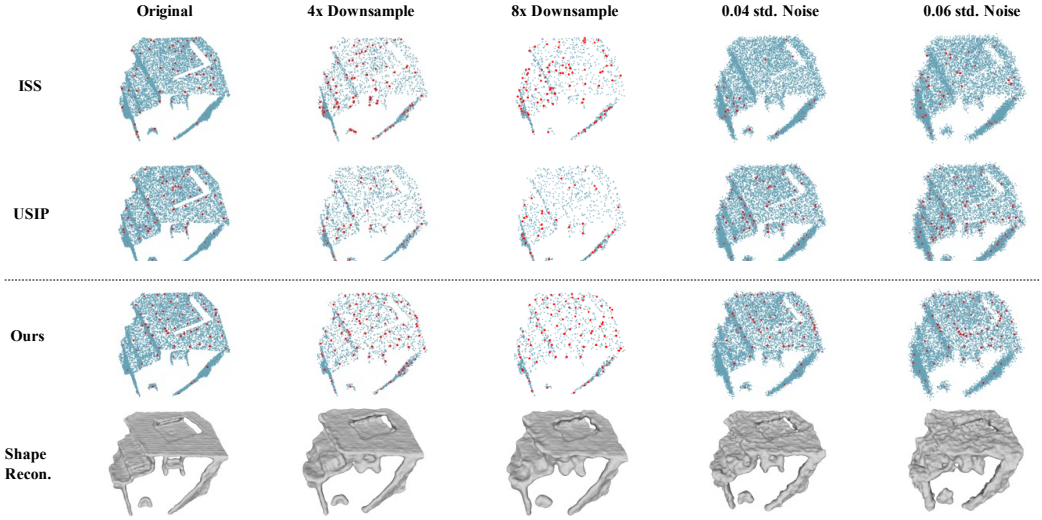

Figure 10: Keypoints of the indoor scene (2) under some input disturbances.

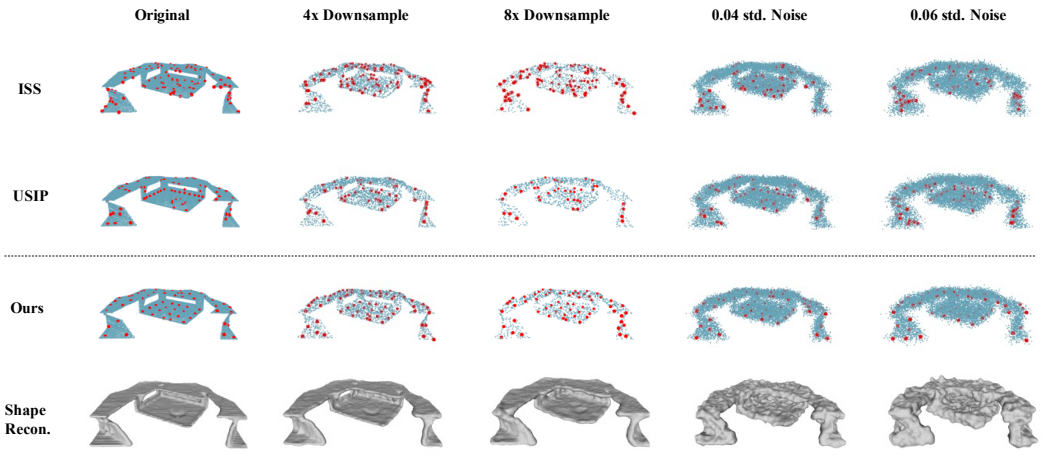

Figure 11: Keypoints of the indoor scene (3) under some input disturbances.

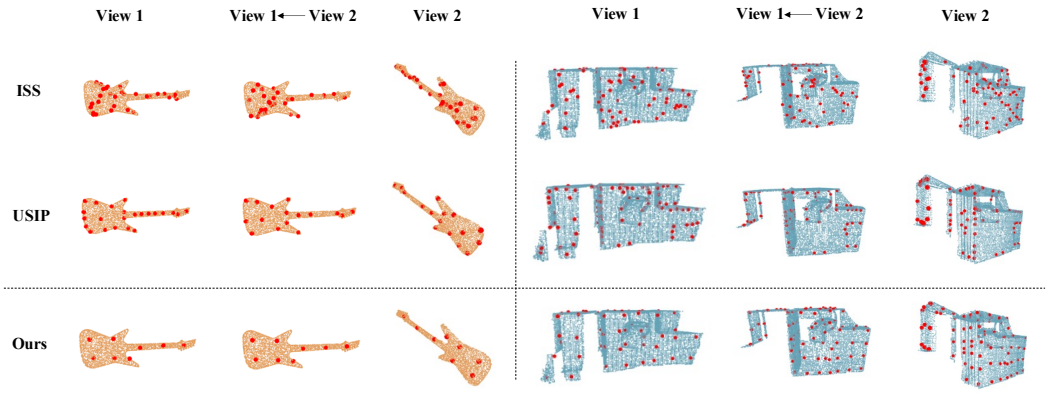

Figure 12: Keypoints repeatability comparison when the input is not corrupted. Note that in the Redwood dataset (right panel), two-view point clouds are partially overlapped.

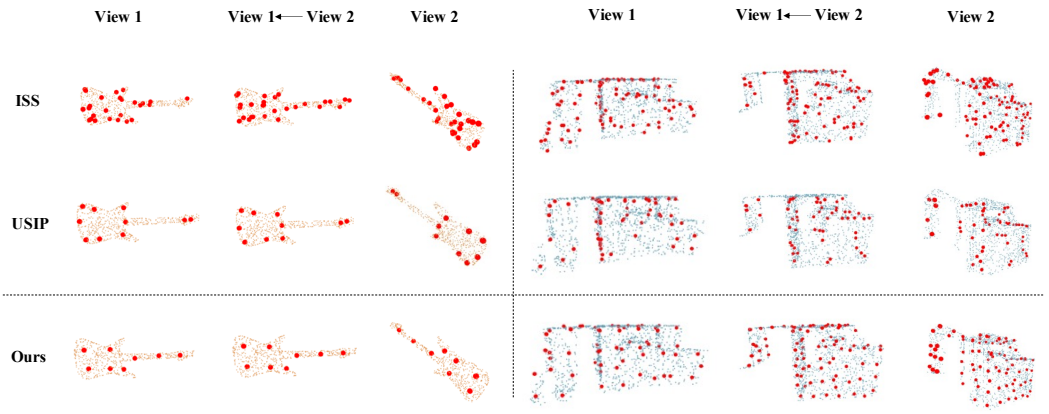

Figure 13: Keypoints repeatability comparison when the input is 8x down sampled. Note that in the Redwood dataset (right panel), two-view point clouds are partially overlapped.

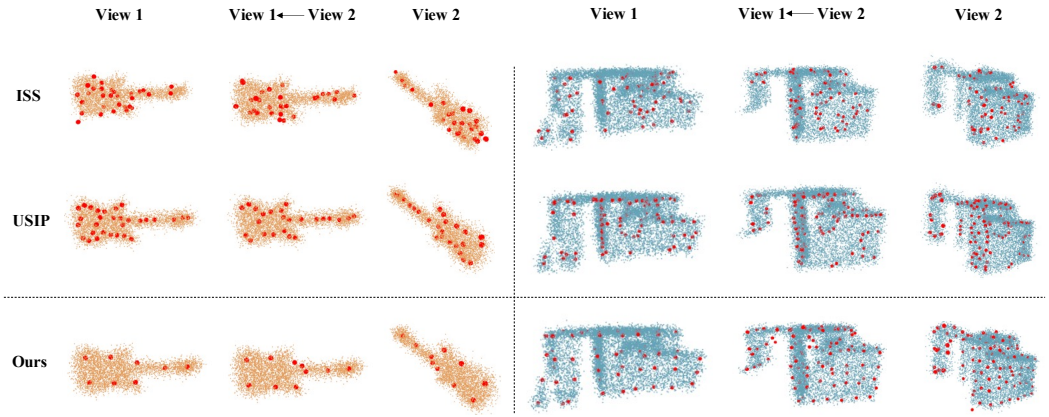

Figure 14: Keypoints repeatability comparison when the input is added Gaussion noises (std=0.06). Note that in the Redwood dataset (right panel), two-view point clouds are partially overlapped.

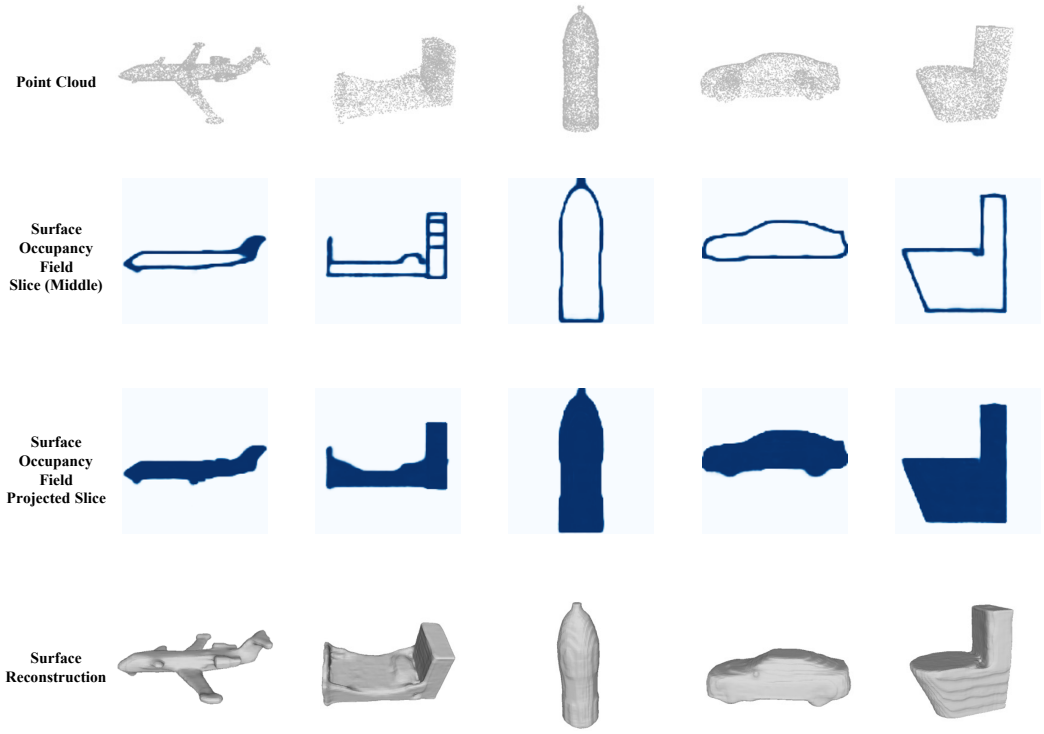

Figure 15: Visualization for surface occupancy field and surface reconstruction of test instances (unseen) from ModelNet40 dataset. The second row shows the middle slice of the surface occupancy field of these objects. The third row shows the projected surface occupancy field on the same slice by taking the maximum value. The fourth row shows the outer surface reconstructed by applying marching cubes on the surface occupancy field, using a threshold of 0.4.

References

[1] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6359–6367, 2020.

[2] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5556–5565, 2015.

[3] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[5] Jiaxin Li and Gim Hee Lee. Usip: Unsupervised stable interest point detection from 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 361–370, 2019.

[6] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-person linear model. *ACM Trans. Graph.*, 34(6), oct 2015.

[7] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[8] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision*, pages 523–540. Springer, 2020.

[9] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[10] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[11] Yang You, Wenhai Liu, Yanjie Ze, Yong-Lu Li, Weiming Wang, and Cewu Lu. Ukpgan: A general self-supervised keypoint detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17042–17051, June 2022.

[12] Yang You, Yujing Lou, Chengkun Li, Zhoujun Cheng, Liangwei Li, Lizhuang Ma, Cewu Lu, and Weiming Wang. Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13647–13656, 2020.

[13] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas A. Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 199–208, 2017.

[14] Yu Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 689–696. IEEE, 2009.

[15] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.