Supplementary

1. More Analysis on Keypoint Controllability with Beta Distribution

In Beta distributions, we have α and β controlling the keypoint probability accumulation around 0 and 1, respectively. This controllability is illustrated in Figure 1. We see that our model automatically identify those semantic keypoints with certain distribution requirements. By introducing Beta distribution and GAN sparsity control, our model is able to detect corner points when budget is tight, and detect less salient points (e.g., edge points) when more budget is given. In contrast, previous methods on keypoint detection do not give much control on the number of keypoints.

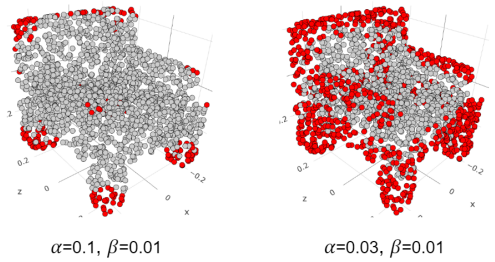

Figure 1. We can easily control the number of keypoints with Beta distribution parameters. Keypoints are shown in red with $p > 0.5$.

2. Precise Control of Number of Keypoints

Our model outputs a keypoint distribution, which lies in $[0, 1]$. In order for a precise control of specific number of keypoints, we could use iterative Non-Maximum-Suppression (NMS) with radius r. Specifically, suppose we want exactly K keypoints, at each iteration, we pick the point with largest Φ and then invalidate all geodesic neighbors within radius r, this iteration is repeated until we have K points.

3. Results on SMPL Models with NMS Applied

We evaluate all the methods on SMPL models by applying NMS with fixed number of 10, 20 and 40 keypoints. Qualitative comparison is given in Figure 2 and quantitative results are listed in Table 1.

4. Results on More ShapeNet Categories

We also evaluate a universal model that is trained on a large collection of ShapeNet models, including 11 categories. Quantitative results are given in Figure 3. Qualitative results are illustrated in Figure 4.

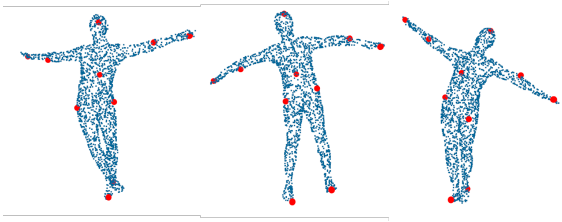

Figure 2. Results on SMPL models. NMS is applied to ensure 10 keypoints.

	IoU (%) ↑			Consis. ($\times 10^{-2}$) ↓		
	10	20	40	10	20	40
USIP	2.0	2.4	2.1	13.9	12.5	12.7
D3Feat	1.6	1.8	1.9	15.8	14.1	11.6
HARRIS-3D	0.4	0.5	0.9	21.5	15.1	10.4
ISS	1.0	0.9	1.2	20.8	14.8	10.3
SIFT-3D	0.1	0.4	1.0	21.9	14.9	10.4
Ours	**7.2**	**7.5**	**9.2**	**9.2**	**7.9**	**5.9**

Table 1. IoU (%) and Consistency Loss ($\times 10^{-2}$) results for SMPL dataset, given the budget of 10, 20, 40 keypoints.

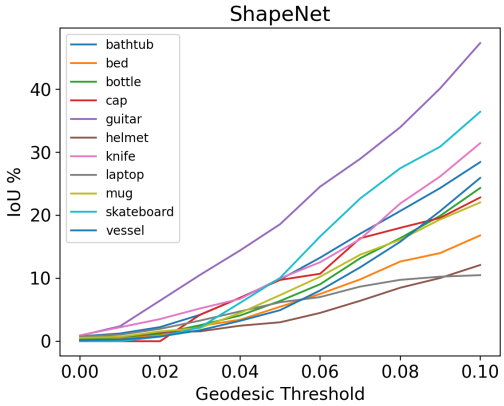

Figure 3. Quantitative results with a universal model on ShapeNet models. NMS with radius 0.1 is applied.

5. Extension on Unsupervised 2D Binary Image Analysis

Our method is not only restricted to 3D shape analysis, but 2D binary images. In this experiment, we evaluate our method on MNIST, by viewing each pixel as a 2D point. The encoder is replaced with 2D convolution without local reference frame. Digits from all classes are trained jointly.

The results on unsupervised keypoint detection is shown in Figure 6. It can be shown that our algorithm captures important keypoint skeletons in MNIST digits, and they are consistent within each class. In the meantime, unsupervised

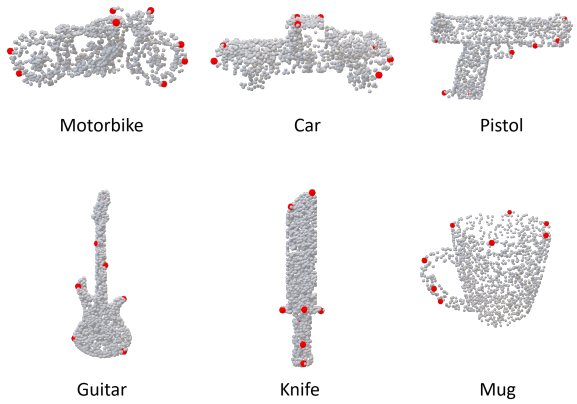

Figure 4. Qualitative keypoint detection results on more ShapeNet categories. NMS with radius 0.1 is applied.

dense embeddings are also predicted for each pixel. Quite interestingly, the generated embeddings (Figure 7) are consistent within each class, without acquiring any class label at training time.

6. More Visualizations on Detected Keypoints under Arbitrary Rotations

We plot more visualization results in Figure 8, where each model is rotated four times. Keypoints are filtered by $p > 0.5$ with no NMS is applied. We see that UKPGAN does a pretty good job in maintaining rotation repeatability.

7. More Qualitative Results on Real-World Keypoint Detection

Here, we plot more keypoint detection results on real-world scenarios, under both indoor (3DMatch) and outdoor (ETH) settings. Notice that our model is trained on synthetic models only, while it generalizes to real-world scenarios well. We can see that our method detect salient corner points on both indoor and outdoor datasets, thus boosting the performance of geometric registration.

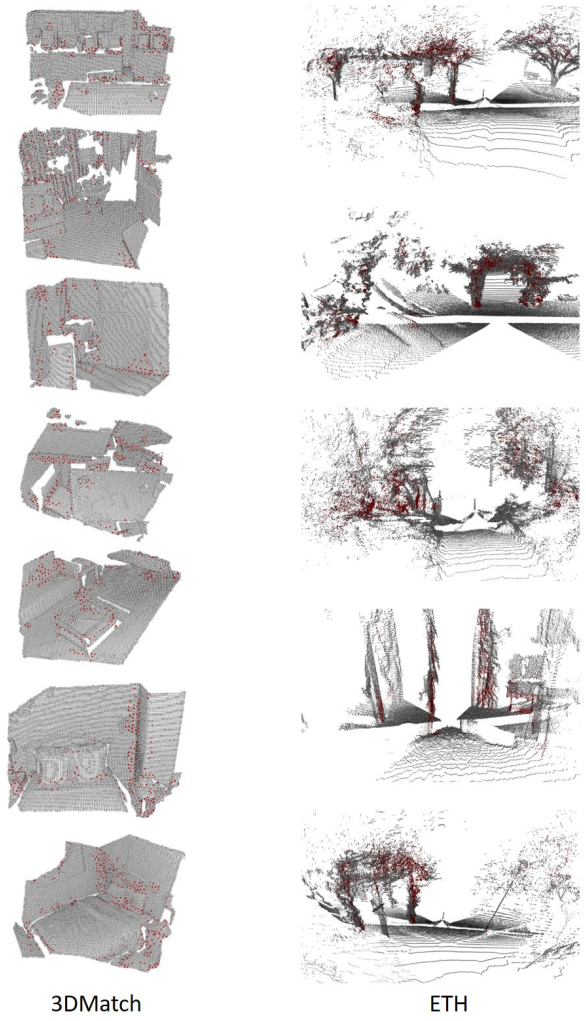

Figure 5. Keypoint detection results on real-world scenes. Left: indoor 3DMatch dataset. Right: outdoor ETH dataset.

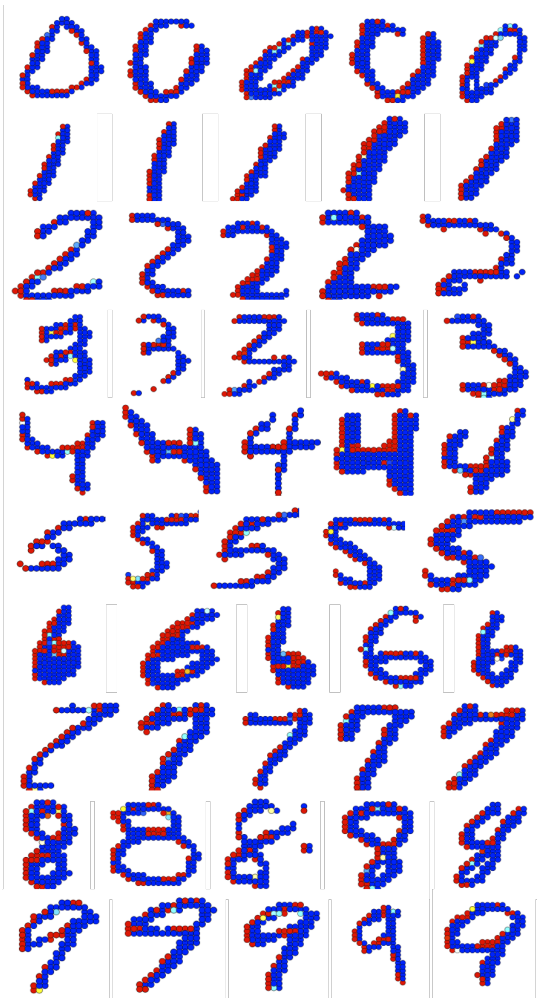

Figure 6. **Keypoint probability heat-map on MNIST.** Red indicates high probability.

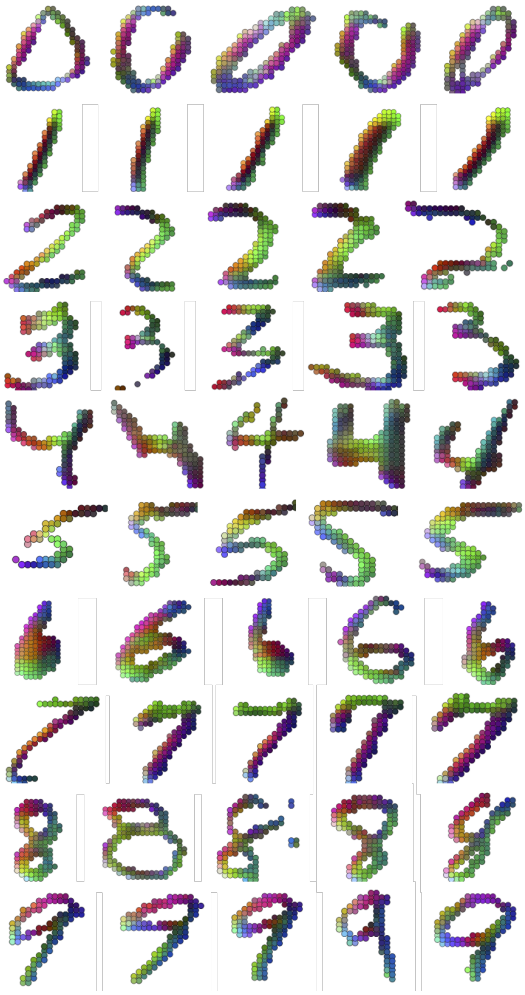

Figure 7. **Dense embeddings generated on MNIST, which are consistent across digits within the same class.** Best viewed in color.

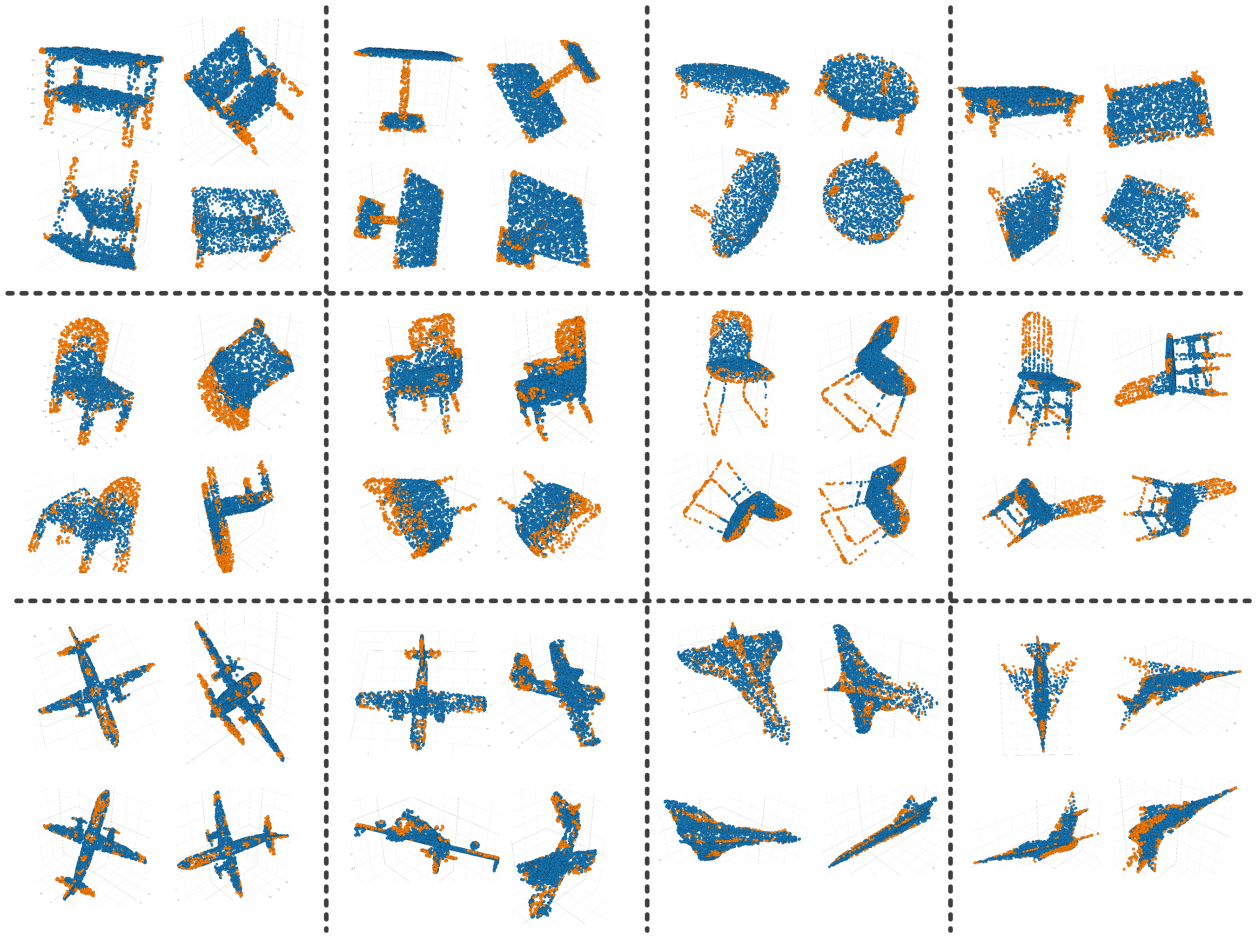

Figure 8. More visualization results. Each model is rotated four times. Point clouds are shown in blue while keypoints are shown in orange.