ESCAPE: Equivariant Shape Completion via Anchor Point Encoding

Supplementary Material

A. Robustness Analysis

A.1. Error Bounds

This section provides detailed proofs for the theoretical guarantees mentioned in the main paper regarding our distance-based representation's error bounds compared to other equivariant features such as Vector Neurons.

In Vector Neurons, each layer performs transformation:

$$\mathbf{v}_{\text{out}} = \mathbf{R}(\mathbf{W}\mathbf{v}_{\text{in}}) \tag{1}$$

where $\mathbf{R}$ is a rotation matrix and $\mathbf{W}$ are learned weights. For an input perturbation $\boldsymbol{\epsilon}$, the error after L layers propagates as:

$$\|\mathbf{e}_L\| \leq \prod_{i=1}^{L} \|\mathbf{W}_i\| \cdot \|\mathbf{R}_i\| \cdot \|\boldsymbol{\epsilon}\| = \mathcal{O}(\alpha^L) \tag{2}$$

where $\alpha = \max_i \|\mathbf{W}_i\| \cdot \|\mathbf{R}_i\| > 1$ typically.

For our distance-based representation, given points P and anchor points A, a perturbation $\boldsymbol{\epsilon}$ affects distances as:

$$|d(p + \boldsymbol{\epsilon}, a) - d(p, a)| \leq \|\boldsymbol{\epsilon}\| \tag{3}$$

by the triangle inequality. This leads to our key result:

Theorem A.1 (Error Bounds) *For input perturbation $\boldsymbol{\epsilon}$, our distance matrix representation maintains constant error bounds:*

$$\|D(P + \boldsymbol{\epsilon}, A) - D(P, A)\|_\infty \leq \|\boldsymbol{\epsilon}\|_2 \tag{4}$$

independent of network depth.

This theoretical advantage is empirically validated in Section 4 of the main paper, where we demonstrate superior robustness on real-world datasets that are based on VectorNeurons.

A.2. Performance under Noise

To validate our theoretical error bounds empirically, we evaluate ESCAPE's robustness by adding Gaussian noise with varying standard deviations σ to input point clouds. As shown in Table 1, our method maintains stable performance under increasing noise levels (σ = 0.001 to 0.004), with average CD-L1 scores only degrading from 10.58 to 10.99.

Table 1. Results of ESCAPE under input noise.

Category	σ	0.0	0.001	0.002	0.004
Plane		**8.60**	8.66	8.83	8.71
Cabin		13.62	13.74	**13.45**	13.67
Car		**10.43**	10.46	10.47	10.76
Chair		10.71	10.72	**10.68**	11.08
Lamp		8.14	**8.11**	8.30	8.87
Sofa		**13.86**	14.00	14.05	14.73
Table		**9.23**	9.28	9.40	9.48
Boat		**10.00**	10.01	10.01	10.55
Avg		**10.58**	10.62	10.65	10.99

A.3. Different Level of Partiality

Similar to the noise experiment, here we exclude a random portion of the input points, determined by a removal ratio of p. The results of point removal are shown in Table 2. The experiment demonstrate that our method is robust to input perturbations, maintaining consistent performance across varying conditions. The robustness of ESCAPE is further illustrated in Figure 1. ESCAPE reliably handles input noise and maintains output quality, even when large portions of input points are removed.

Table 2. Results of removing portion of the input points.

Category	p	0.0	0.1	0.25	0.5
Plane		8.60	8.49	**8.32**	8.92
Cabin		13.62	12.69	**12.75**	13.00
Car		10.43	**10.39**	10.81	10.41
Chair		**10.71**	11.43	11.21	11.36
Lamp		**8.14**	8.72	8.70	8.84
Sofa		13.86	**13.74**	13.80	13.97
Table		**9.23**	9.96	10.06	10.65
Boat		10.00	9.96	**9.83**	10.02
Avg		**10.58**	10.68	10.69	10.90

A.4. Visualization of KITTI Dataset

To further demonstrate robustness on real-world data, we evaluate on the KITTI dataset using models pretrained on PCN car shapes. Unlike controlled environments, KITTI presents noisy, partial scans from actual LiDAR sensors with unknown object orientations. As shown in Figure 2, ESCAPE maintains consistent completion quality under various orientations, including single-axis rotations mimicking vehicle movement and arbitrary three-axis rotations. This validates our method's practical applicability in real-world

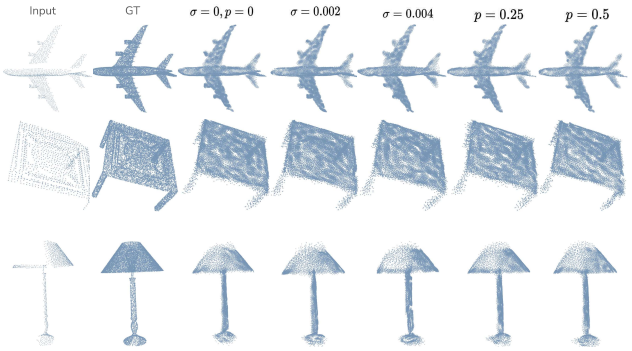

Figure 1. Quantitative results of additive noise and point removal experiments.

scenarios where clean, canonically-oriented inputs cannot be guaranteed.

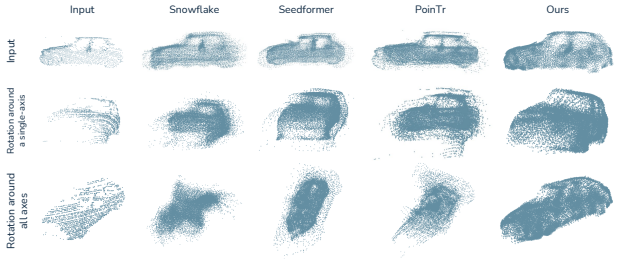

Figure 2. Qualitative comparison of models fine-tuned on PCN dataset cars category and tested on KITTI dataset. The first row contains the original input. The second row contains a single-axis rotation of the input mimicking the movement of a car. The final row contains the partial input rotated in all three axes.

B. Anchor Point Analysis

B.1. Deterministic FPS

In our method, we modified farthest point selection algorithm to make it deterministic and obtain exactly same results for the same input independent of rotation. To remove the randomness of FPS, we changed the selection of the first point by finding the point most distant from the center point calculated as $C = \frac{1}{|P|} \sum_{p_i \in P} p_i$. Selection of all other points executed normally. With this modification, our method's performance stays constant even under input rotation.

B.2. Comparison with Learned Keypoints

ESCAPE's modular design supports integration with any set of keypoints generated by various algorithms. For comparison, we selected two self-supervised keypoint detection methods: SNAKE [7] and SkeletonMerger [3] and benchmarked them against our curvature-based keypoint selection approach. We trained class-specific models to generate keypoints using SkeletonMerger and ensured the number of keypoints matched those used in ESCAPE. Similarly, we utilized its pre-trained weights for SNAKE and applied Farthest Point Sampling (FPS) to select a subset of predicted keypoints, aligning their count with ESCAPE's. The results are given in Table 3. The results demonstrated that our anchor points outperformed learning-based counterparts, achieving superior performance on rotated and canonical input datasets.

Table 3. Results on PCN dataset with different set of keypoints. We use CD-L1($\times 1000$) as an evaluation metric and report the results for normal and rotated inputs. Best results for both input type is written with bold letters.

Category	Keypoints	Skeleton [3]	SNAKE [7]	Anchors
Airplane	canonical	**6.73**	10.47	8.6
	rotated	19.17	10.66	**8.6**
Cabin	canonical	28.14	13.86	**13.62**
	rotated	30.23	13.64	**13.62**
Car	canonical	**9.91**	10.35	10.43
	rotated	20.89	10.49	10.43
Chair	canonical	**9.98**	10.57	10.71
	rotated	19.64	**10.80**	10.71
Lamp	canonical	**8.02**	8.67	8.14
	rotated	11.51	9.08	**8.14**
Sofa	canonical	**12.74**	14.65	13.86
	rotated	26.15	14.79	**13.86**
Table	canonical	**8.19**	9.54	9.23
	rotated	21.24	9.38	**9.23**
Boat	canonical	**9.59**	10.97	10.00
	rotated	17.66	11.25	**10.00**
Avg	canonical	11.67	11.14	**10.58**
	rotated	20.82	11.27	**10.58**

B.3. Effect of Number of Anchor Points

We analyze how the number of anchor points affects reconstruction quality during optimization. In this experiment, we compare using different subsets (k = 3, 4, and 6) of anchor points against our full set of 8 points, while maintaining the predicted distances for each subset. The results in Table 4 demonstrate a clear trend: increasing the number of anchor points improves reconstruction quality. With fewer anchor points, the optimization becomes more sensitive to noise and prediction errors, as there are fewer geometric constraints. However, using too many anchor points increases model complexity and computational overhead. Through extensive experiments, we found that k = 8 provides the optimal balance between reconstruction accuracy and computational efficiency.

C. Extended Comparisons

C.1. Performance on Canonical Inputs

While our method prioritizes rotation equivariance through invariant features instead of point coordinates, this design choice incurs a small performance trade-off on canonical

Table 4. Results of having different number of anchor points in the optimization.

Category	K	3	4	6	8
Airplane		10.21	9.98	8.68	**8.6**
Cabin		14.87	14.77	13.54	**13.62**
Car		12.54	12.50	10.84	**10.43**
Chair		13.13	13.37	11.10	**10.71**
Lamp		9.70	9.79	8.43	**8.14**
Sofa		16.19	16.32	14.28	**13.86**
Table		12.78	12.65	9.51	**9.23**
Boat		11.18	11.13	9.96	**10.00**
Avg		12.58	12.57	10.79	**10.58**

inputs. As shown in Table 5, existing methods achieve slightly better CD-L1 scores on canonical inputs by exploiting aligned data, but their performance degrades significantly (up to 14x worse) under rotation. Notably, methods with better canonical performance show more severe degradation under rotation, suggesting dataset memorization rather than true geometric understanding. This observation is further validated by OmniObject results, where models with lower PCN scores demonstrated better generalization to real-world data. In contrast, ESCAPE maintains consistent performance (10.58) regardless of orientation, demonstrating the value of our rotation-invariant representation.

C.2. Baseline Comparison - PCA

To evaluate existing non-equivariant methods, we applied PCA [1] to align the rotated point clouds, thereby normalizing the inputs before completion. After predicting the complete geometry, we applied the inverse rotation to align the completed shapes with the ground truth. The results are given in Table6. Our method is the only model where prediction is unaffected by the input rotation and achieves rotation-equivariance.

C.3. Baseline Comparison - Augmentation

We have conducted additional experiments training the baselines with comprehensive augmentations, including random rotations ($\pm 180^\circ$) and translations (± 0.1) on all axes. Below are the results showing how augmentation affects the performance of baseline methods:

Table 7. Comparison of ESCAPE with the existing methods trained with augmentation

	Snowflake	PoinTr	AdaPoinTr	AnchorFormer	Ours
CD L1	43.60	47.0	31.25	47.27	10.58

Our method still maintains advantages in the CD L1 metric. These results strengthen our contribution by demonstrating benefits even compared to augmentation-trained baselines.

D. Computational efficiency

We measured the average inference time of ESCAPE and compared to against existing methods, as summarized in Table 8. The results show that our method ranks second fastest regarding the milliseconds required to predict a single sample.

Our optimization step currently takes ~1.72 seconds per shape, which is competitive with many post-processing steps in 3D reconstruction pipelines. Additionally, hardware acceleration can significantly reduce this, making it more feasible for real-world applications.

E. Analysis of Point Selection and Distribution

E.1. Visualization of point curvatures

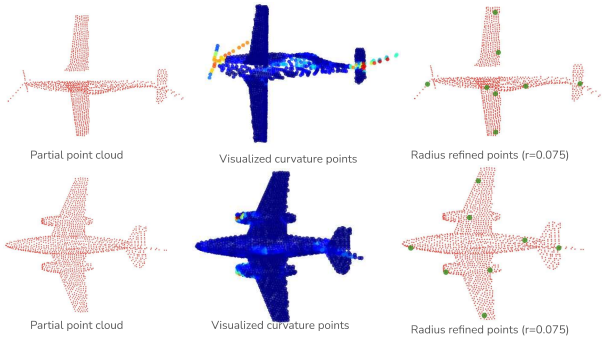

Figure 3. Heatmap of the point curvature values and derived anchor points.

Figure 3 illustrates the calculated curvature value for each point in the input and derived refined anchor points using the described algorithm in the main paper. The figure depicts that our algorithm can select points with high curvature and still cover all regions of the shape, hence good anchor points for point cloud completion.

E.2. Analysis of Anchor Point Selection Strategies

Figure 4 provides a comprehensive comparison of different anchor point selection strategies. Our analysis reveals that initializing with Farthest Point Sampling (FPS) provides effective coverage by selecting well-distributed points across the shape. While these points can be refined using curvature information (as shown in Figure 3), we find that the refinement process must be constrained. Without proper limits, the refinement can cause anchor points to cluster (visible

Table 5. Results on PCN dataset. We use CD-L1($\times$1000) as an evaluation metric and report the results for canonical inputs versus rotated frame. The best results for both input types are written with bold letters.

	Snowflake [4]	Seedformer [8]	PointTr [5]	AdaPoinTr [6]	AnchorFormer [2]	Ours
Canonical	7.21	6.74	8.38	**6.53**	6.59	10.58
Rotated	88.85	92.15	30.20	33.52	26.65	**10.58**

Table 6. Results on PCN dataset. We use CD-L1($\times$1000) as an evaluation metric and report the results for rotated inputs. The best results are written with bold letters.

	Snowflake[4]	Seedformer[8]	PointTr[5]	AdaPoinTr[6]	AnchorFormer[2]	Ours
Plane	72.71	76.19	13.03	12.10	11.88	**8.6**
Cabin	85.81	85.99	47.97	50.35	32.93	**13.62**
Car	78.76	82.28	37.42	40.90	28.97	**10.43**
Chair	64.57	66.28	30.53	37.24	34.94	**10.71**
Lamp	141.04	148.67	19.01	19.77	17.73	**8.14**
Sofa	73.63	77.47	44.13	49.46	33.89	**13.86**
Table	87.31	89.58	29.03	37.74	36.57	**9.23**
Boat	106.97	110.72	20.51	20.60	16.26	**10.00**
Avg	88.85	92.15	30.20	33.52	26.65	**10.58**

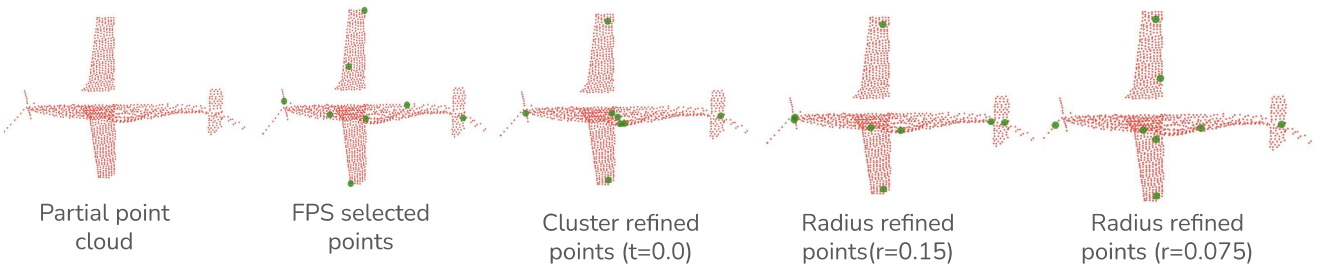

Figure 4. Comparison of different anchor point selection algorithms.

Table 8. Inference time of a single input with existing methods.

Models	Elapsed Time
Snowflake[4]	16.7 ms
Seedformer[8]	45.1 ms
PoinTr[5]	38.5 ms
AdaPoinTr[6]	35.6 ms
AnchorFormer[2]	21.8 ms
ESCAPE	19.3 ms

in top-right and bottom-left of Figure 4), reducing spatial coverage and potentially leading to fewer effective anchor points. Our proposed algorithm achieves an optimal balance between salient anchors (high curvature) and spatial distribution, as demonstrated by the improved reconstruction results in the bottom-right of Figure 4.

F. Limitations

While our approach demonstrates significant advantages in achieving rotation-equivariant shape completion, it has limitations. Firstly, although our method is less data-driven than techniques that learn rotation through augmentation, it still requires substantial training data to achieve high performance. This dependency on data can be a bottleneck, especially for applications where labeled data is scarce or expensive to obtain.

Another key limitation of our approach is the optimization procedure to find the coordinates of the complete shape. This procedure prevents the model from being rotation-invariant, which will limit its applicability to some real-world applications.

Furthermore, while distance-based encoding contributes to rotation invariance, it also introduces additional complexity to the learning process. This complexity can sometimes

result in lower performance on standard, non-rotated cases. In scenarios where the objects are consistently presented in a canonical alignment, methods that memorize these aligned shapes may outperform our approach. The trade-off between achieving rotation invariance and maintaining high performance on canonical shapes is an essential consideration for the practical deployment of our model.

References

[1] Abdi, H., Williams, L.J.: Principal component analysis. Wiley interdisciplinary reviews: computational statistics **2**(4), 433–459 (2010)

[2] Chen, Z., Long, F., Qiu, Z., Yao, T., Zhou, W., Luo, J., Mei, T.: Anchorformer: Point cloud completion from discriminative nodes. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13581–13590 (2023)

[3] Shi, R., Xue, Z., You, Y., Lu, C.: Skeleton merger: an unsupervised aligned keypoint detector. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 43–52 (2021)

[4] Xiang, P., Wen, X., Liu, Y.S., Cao, Y.P., Wan, P., Zheng, W., Han, Z.: Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5499–5509 (2021)

[5] Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., Zhou, J.: Pointr: Diverse point cloud completion with geometry-aware transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12498–12507 (2021)

[6] Yu, X., Rao, Y., Wang, Z., Lu, J., Zhou, J.: Adapointr: Diverse point cloud completion with adaptive geometry-aware transformers. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)

[7] Zhong, C., You, P., Chen, X., Zhao, H., Sun, F., Zhou, G., Mu, X., Gan, C., Huang, W.: Snake: Shape-aware neural 3d keypoint field. Advances in Neural Information Processing Systems **35**, 7052–7064 (2022)

[8] Zhou, H., Cao, Y., Chu, W., Zhu, J., Lu, T., Tai, Y., Wang, C.: Seedformer: Patch seeds based point cloud completion with upsample transformer. In: European conference on computer vision. pp. 416–432. Springer (2022)