Noise2Score3D: Tweedie's Approach for Unsupervised Point Cloud Denoising

Supplementary Material

A. Denoising and Estimating unknown noise parameters with Total Variation for Point cloud

In image analysis, Total Variation (TV) is primarily used as a quality metric to evaluate the denoising result by calculating the remaining high-frequency content [43]. A lower TV value indicates a smoother output with fewer irregularities, whereas a higher TV value suggests the presence of residual noise or structural inconsistencies. In TV regularization, the optimization process seeks to minimize the total variation of an image, reducing sharp pixel intensity changes that correspond to noise, while preserving significant transitions corresponding to image edges. This has inspired us to propose similar metrics for point clouds.

A.1. Definition of Total Variation for Point Cloud

In this section, we extend the idea of minimizing gradient magnitudes to point cloud data, where the goal is to reduce noise by minimizing the geometric differences between points and their neighbors. In particular, Total Variation for Point Cloud (TV_{PC}) is defined as:

$$TV_{PC} = \sum_{i=1}^{N} \sum_{j \in \text{neighbors}(i)} w_{i,j} \cdot \sqrt{\|\mathbf{p}_i - \mathbf{p}_j\|^2 + \epsilon^2} \quad (8)$$

where N is the total number of points, $w_{i,j}$ represents the weight between point $\mathbf{p}_i$ and its neighbor $\mathbf{p}_j$, ϵ is a small positive constant. $\sqrt{\|\mathbf{p}\|^2 + \epsilon^2}$ in Eq. (8) is a smooth approximation of the L_1 norm, which is introduced in [29] to handle outliers and maintain robustness. Here, ϵ helps to smooth the variations and prevent over-penalization of boundary outliers, ensuring numerical stability. For simplicity we set $w_{i,j}$ to a constant in our method. Another choice for $w_{i,j}$ is

$$w_{i,j} = \exp\left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\sigma^2}\right), \quad (9)$$

with σ controlling the scale of the Gaussian kernel [10] .

This is an analogy to the difference of pixel intensity in TV definition for images. Also we note that TV_{PC} is similar to the Graph Laplacian Regularizer [56] but with smoothed L_1 loss.

A.2. Validation of TV_{PC} in estimating the noise level

Fig. 4 shows the TV_{PC}, CD and $P2M$ values with different noise parameter σ for the inference result on PU-Net dataset with σ=0.03. We can observe that TV_{PC} follows a similar trend as CD and $P2M$ and reaches a minimal around the true σ value. Note that the clean point clouds are not needed for calculation of TV_{PC}. This validates the use of TV_{PC} in estimating unknown noise parameters for real world datasets.

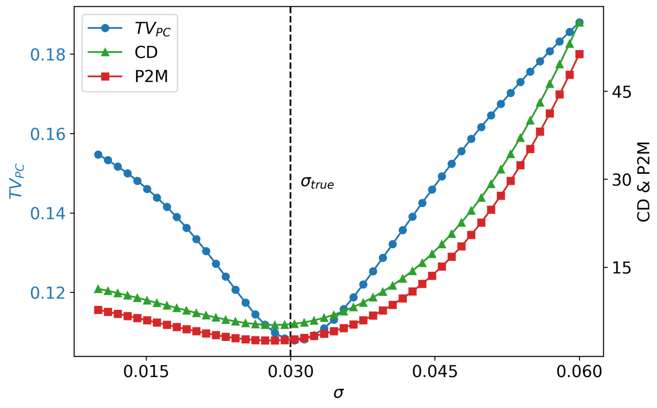

Figure 4. Change of average TV_{PC}, CD and $P2M$ values with noise parameter σ on the PU-Net dataset[55] .

B. Performance Comparison and Ablation Study of Point Cloud Backbones

Tab. 8 evaluates the effectiveness of different point cloud backbones,the analysis reveals that the KPConv backbone does not inherently improve the performance of DMR or Score-U, suggesting that the superior results of our method are not solely attributable to the KPConv network architecture but rather to our novel Bayesian approach. Additionally, we show the result using PointNet++ backbone, which underperforms in sparse point cloud regions and under higher noise conditions compared to KPConv, reinforcing the appropriateness of our chosen backbone for the proposed framework.

Table 8. Results on the PU-Net 10k dataset (results of 50k dataset are not included due to space limitation). Results with * are taken from Tab. 2. in the main text for comparison purpose.

Noise level		1%		2%		3%	
Dataset	Model	CD	P2M	CD	P2M	CD	P2M
PU-Net	DMR-U$_{kpconv}$	33.89	28.16	25.53	20.68	20.69	15.99
	*DMR-U	5.313	2.522	6.455	3.317	8.134	4.647
	Score-U$_{kpconv}$	3.494	<u>1.077</u>	4.928	1.909	7.250	3.596
	*Score-U	<u>3.107</u>	**0.888**	4.675	<u>1.829</u>	7.225	3.762
	Ours$_{PointNet++}$	3.689	1.287	7.938	4.370	13.382	9.028
	***Ours**	**2.848**	1.106	**4.190**	**1.818**	**5.583**	**2.947**

C. Additional visualization results

Figs. 5 to 13 provide additional visualization results with varying noise levels on ModelNet-40 and PU-Net. Points with smaller error are colored more blue, and otherwise colored yellow.

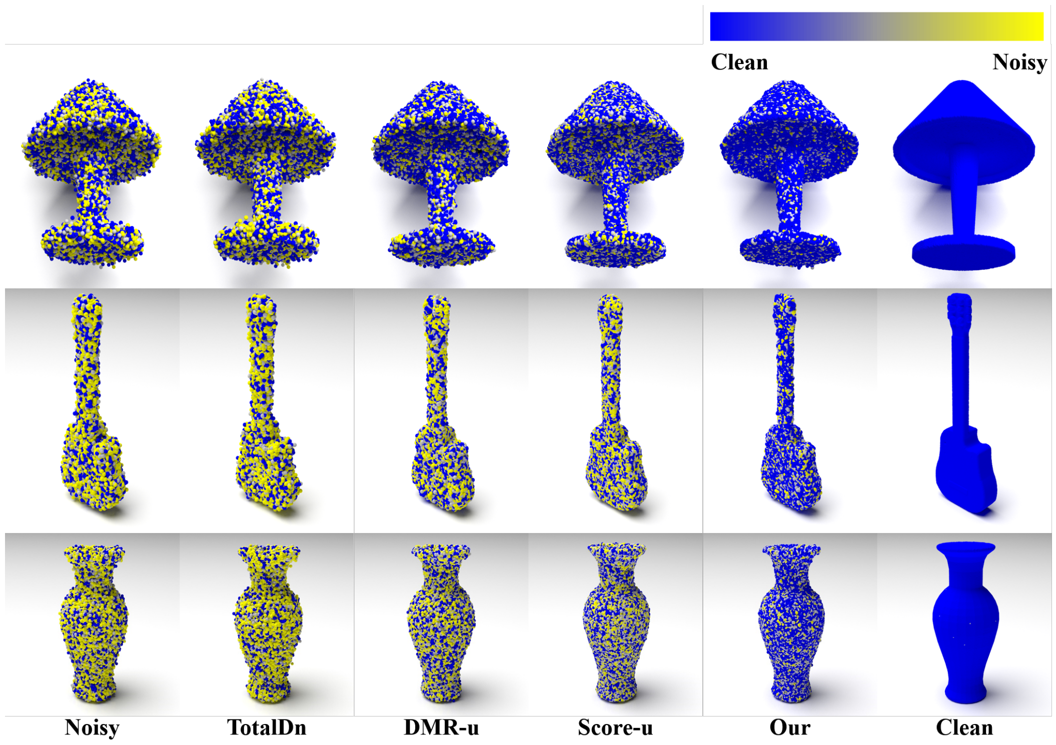

Figure 5. Additional visualization results of different algorithms on ModelNet-40 dataset with Gaussian noise. The noise level is set to 1%.

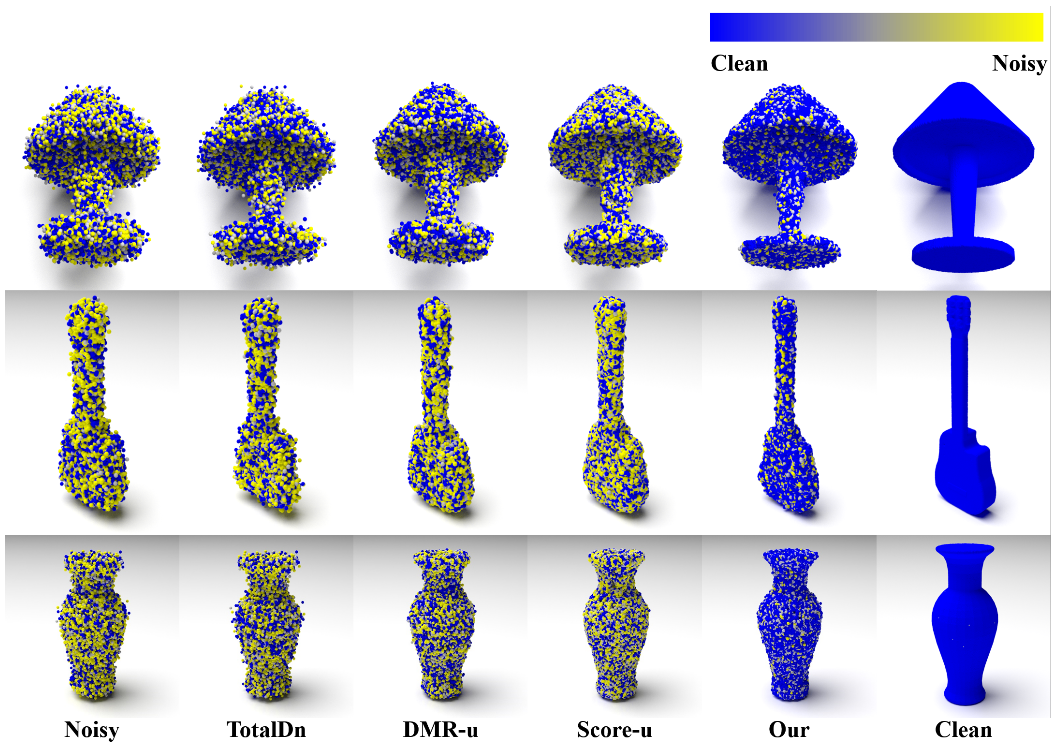

Figure 6. Additional visualization results of different algorithms on ModelNet-40 with Gaussian noise. The noise level is set to 2%.

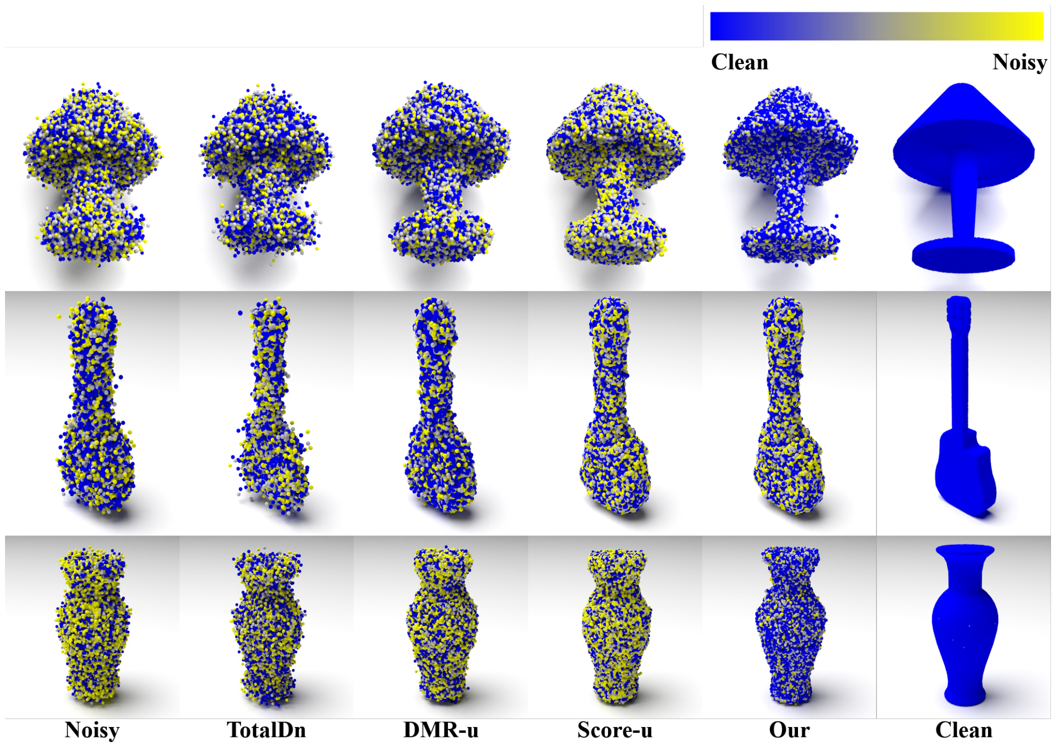

Figure 7. Additional visualization results of different algorithms on ModelNet-40 dataset with Gaussian noise. The noise level is set to 3%.

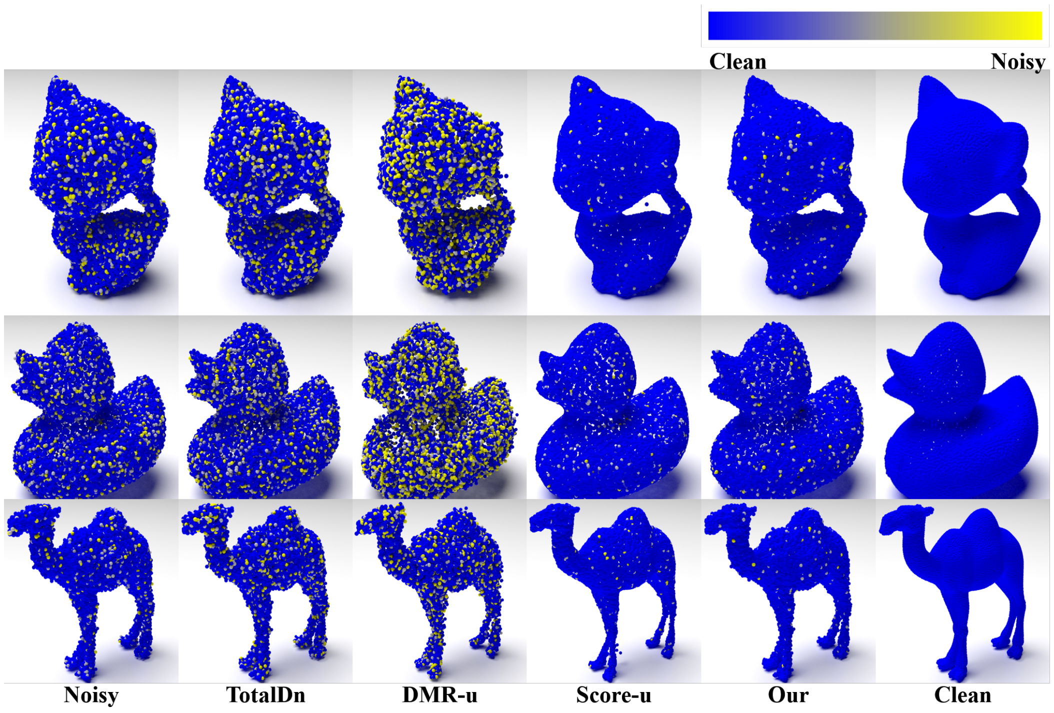

Figure 8. Additional visualization results from PU-Net dataset. The noise level is set to 1%.

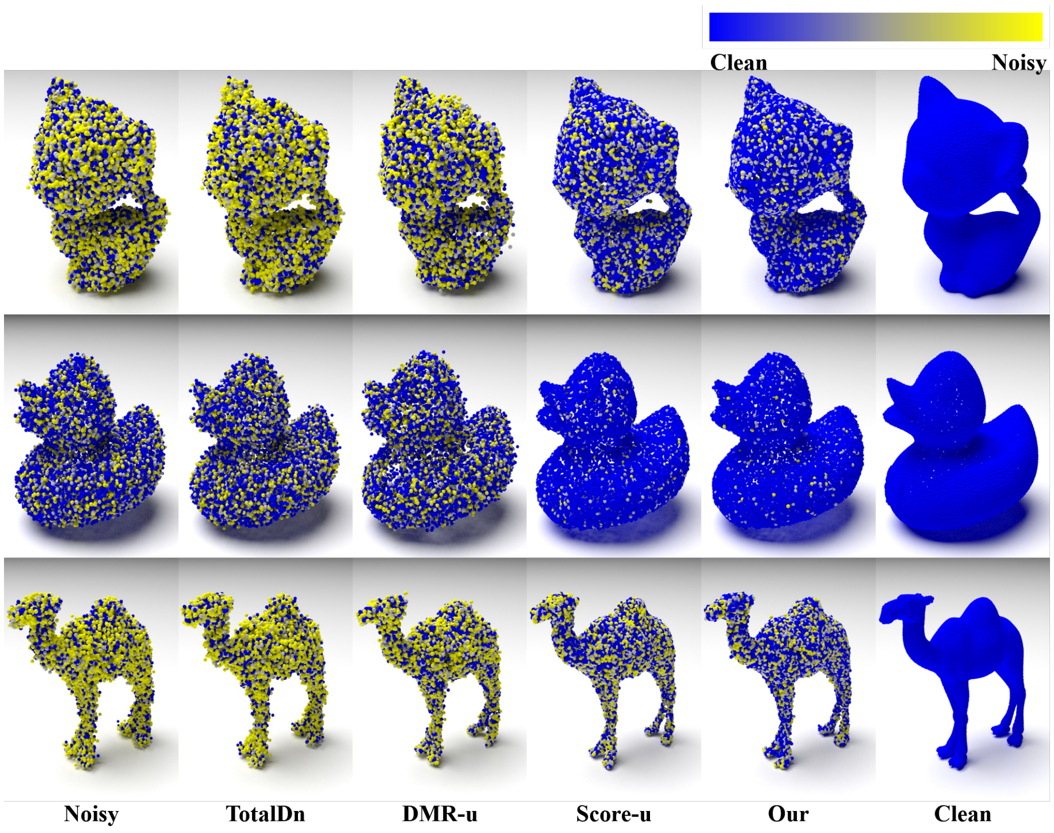

Figure 9. Additional visualization results from PU-Net dataset. The noise level is set to 2%.

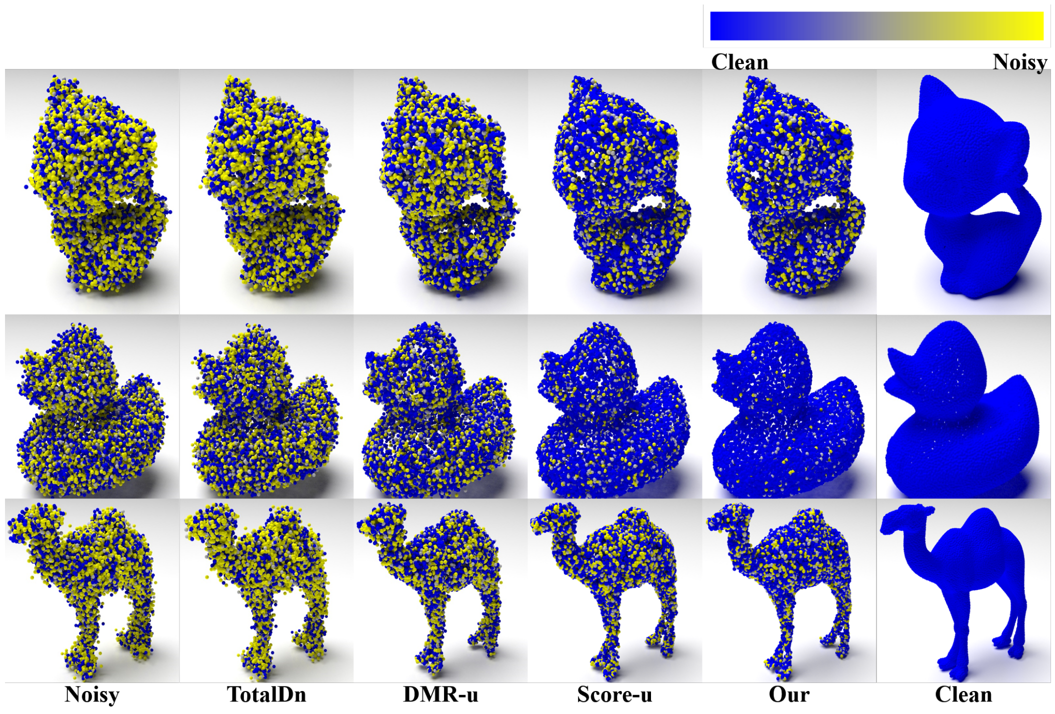

Figure 10. Additional visualization results from PU-Net dataset. The noise level is set to 3%.

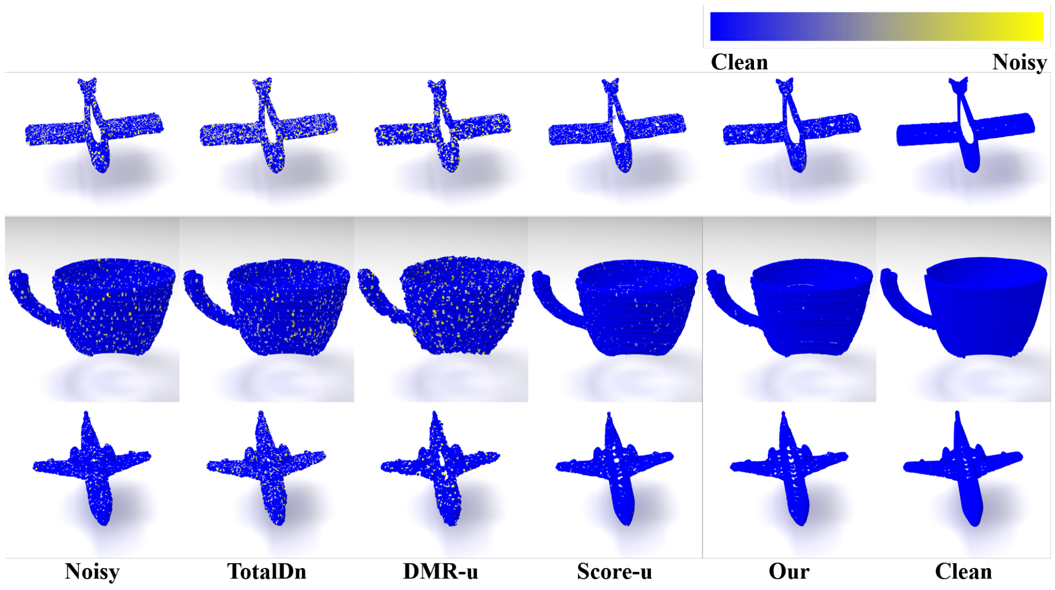

Figure 11. Visual comparison of additional denoising results from ModelNet-40 dataset with simulated LiDAR noise. The noise level is set to 0.5%.

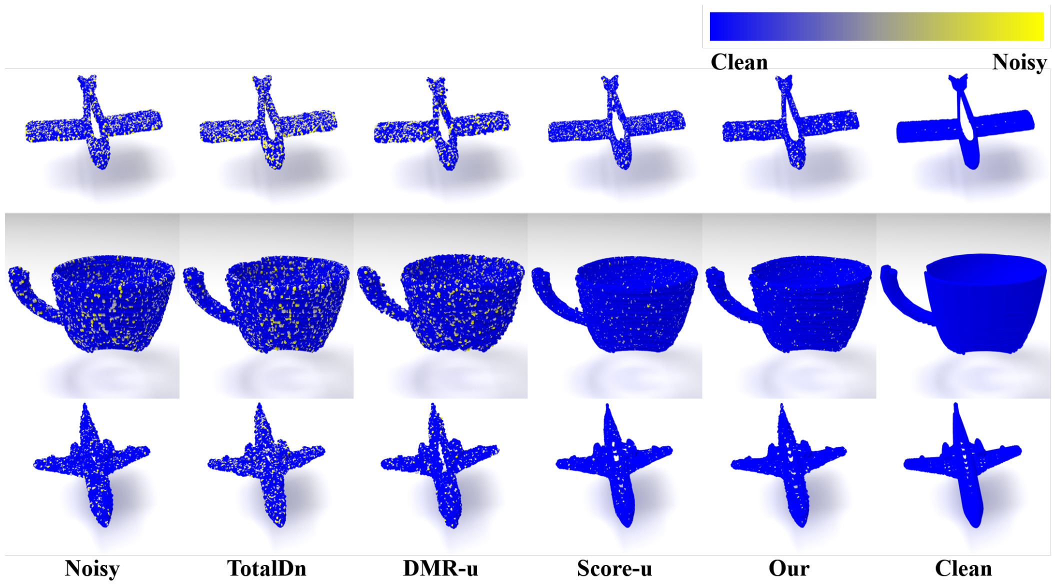

Figure 12. Visual comparison of additional denoising results from ModelNet-40 dataset with simulated LiDAR noise. The noise level is set to 1%.

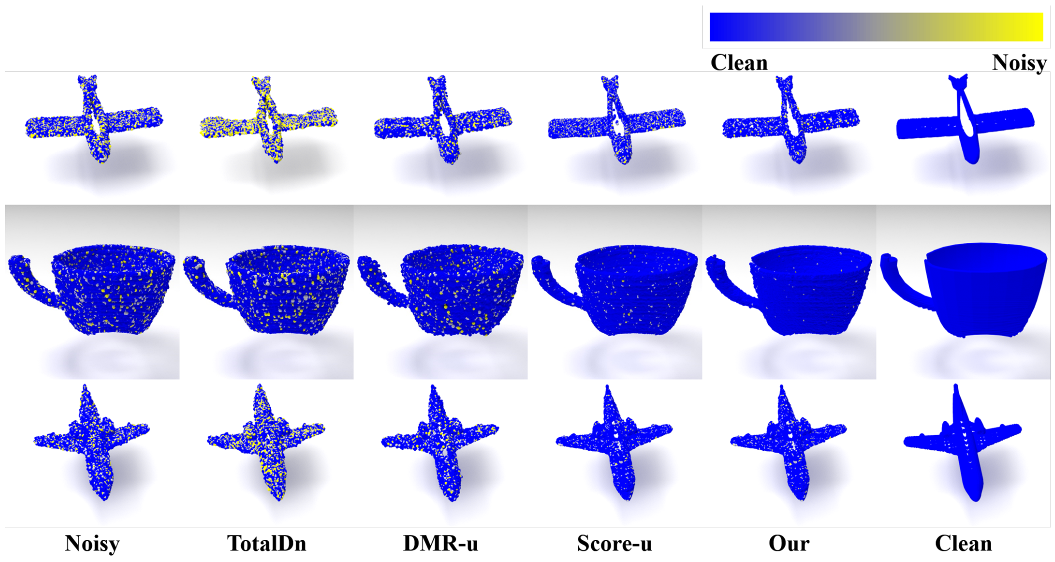

Figure 13. Visual comparison of additional denoising results from ModelNet-40 dataset with simulated LiDAR noise. The noise level is set to 1.5%.