Supplementary Materials

This supplementary includes an introduction to the background of 2D-3D projection (A), implementation details of both 3D-TSDFNet and 3D-SSCNet (B), ablation results of CleanerS with different 2DNets (C) and with different resolutions (D), comparison results of using different methods of mitigating the depth noise (E), the correlation between the noise accuracy degrade (F), and more visualization results (G).

A. 2D-3D Projection

This supplementary is for the background introduction of the main paper. 2D-3D projection layer is used to recover the visible surface in a 3D scene such as to map every pixel in a 2D image to its corresponding 3D spatial position. Given the depth image $\mathbf{I}_d$, each pixel at 2D position $[u, v]$, with a depth value $d = \mathbf{I}_d(u, v)$, is projected to the 3D position $[x, y, z]$. This mapping $\mathbb{M}$ can be expressed as follows:

$$[x, y, z] = \mathbb{M}(u, v, d). \tag{10}$$

Specifically, this projection includes the following two steps:

Step 1: Mapping each 2D pixel to an individual 3D point based on the imaging information including an intrinsic camera matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$ and an extrinsic camera matrix $[\mathbf{R}|\mathbf{t}] \in \mathbb{R}^{3\times4}$. The mapping satisfies the following equation:

$$[\mathbf{R}|\mathbf{t}][x_p, y_p, z_p, 1]^\top = \mathbf{K}^{-1}([u, v, 1]^\top \cdot d). \tag{11}$$

Based on Eq. Eq. (11), we can solve the $[x_p, y_p, z_p]$, i.e., the 3D position of the corresponding point.

Step 2: Discretizing the point position into a grid voxel with a given unit voxel size g,

$$[x, y, z] = [\lfloor x_p/g{+}0.5 \rfloor, \lfloor y_p/g{+}0.5 \rfloor, \lfloor z_p/g{+}0.5 \rfloor], \tag{12}$$

where $\lfloor \cdot \rfloor$ is the floor rounding. The unit grid size g is $0.08m$ and the resultant 3D voxel size is $(60, 36, 60)$. The 2D-3D projection is used for two purposes in this work: 1) getting the class labels of 2D pixels (resulting $Y(d)$ in Section 3 and $\mathbf{Y}_{2D}$ in Section 4.3); 2) translating a 2D feature into a 3D feature (translating from $\mathbf{F}_r$ to $\mathbf{V}_r$ in Section 4.1).

B. Implementation Details of 3D-TSDFNet and 3D-SSCNet

This supplementary is for Section 4.1 of the main paper. We introduce the details of the 3D-TSDFNet and 3D-SSCNet in both student and teacher networks, as presented in Figure S1. Following 3D-Sketch [4], the 3D-TSDFNet includes 3 layers of 3D convolutions to encode the input TSDF into high dimensional features, 8 DDR blocks with different dilations and a downsample rate of 4 to enlarge receptive fields and reduce computation costs, and 2 layers of 3D deconvolutions to upsample features to have the same volume size as the input TSDF volume. Besides, a skip connection is added between each pair of DDR block and deconvolution layer for efficient gradient backpropagation. The 3D-SSCNet uses the same architecture as the 3D-TSDFNet except that it removes the first 3 layers of 3D convolutions.

Methods	Inputs	2DNet	SC-IoU	SSC-mIoU
Baseline	RGB+TSDF	ResNet50	70.6%	43.2%
Teacher	RGB+TSDF-CAD		85.1% (↑ 14.5%)	57.1% (↑ 13.9%)
CleanerS	RGB+TSDF		73.1% (↑ 2.5%)	45.2% (↑ 2.0%)
Baseline	RGB+TSDF	Segformer -B2	71.9%	45.5%
Teacher	RGB+TSDF-CAD		85.3% (↑ 13.4%)	59.4% (↑ 13.9%)
CleanerS	RGB+TSDF		75.0% (↑ 3.1%)	47.7% (↑ 2.2%)

Table S1. The ablation study results for using different 2DNets (ResNet50 [21] *vs*. Segformer-B2 [59]) in our CleanerS on the test set of NYU [51].

C. Ablation Results with Different 2DNet

This supplementary is for Section 5.4 of the main paper. We supplement the ablation study for CleanerS-Res50 and compare the results with those of CleanerS in Table S1. From the table, we can observe that: 1) For all the methods, including baseline, teacher, and CleanerS, better performances are achieved by using Segformer-B2 (than using ResNet50), especially on the metric of SSC mIoU. The reason is that image features extracted from the transformer-based Segformer-B2 encode better global (i.e., longer-range) context information. 2) With different 2DNet architectures (Segformer-B2 or ResNet50), the proposed CleanerS can consistently improve over baseline by large margins, *i.e*., over 2.5% for SC-IoU and over 2.0% for SSC-mIoU. In addition, using better 2DNets (Segformer-B2), our CleanerS achieves even higher improvement (over baseline). This demonstrates the effectiveness and robustness of the proposed CleanerS for resolving the problem of depth noise in SSC.

D. Ablation Results with Different Resolutions

This supplementary is for Section 5.4 of the main paper. We conduct experiments with a lower resolution under the 3D size of (30, 18, 30). Experimental results are given in Table S2, where "HR/LR" denotes high/low resolution. Furthermore, "HR2LR" is a variant by distilling from an HR teacher to an LR student, which is meant to verify if an HR teacher will enable better knowledge distillation. We can observe that: 1) compared to HR, our CleanerS with

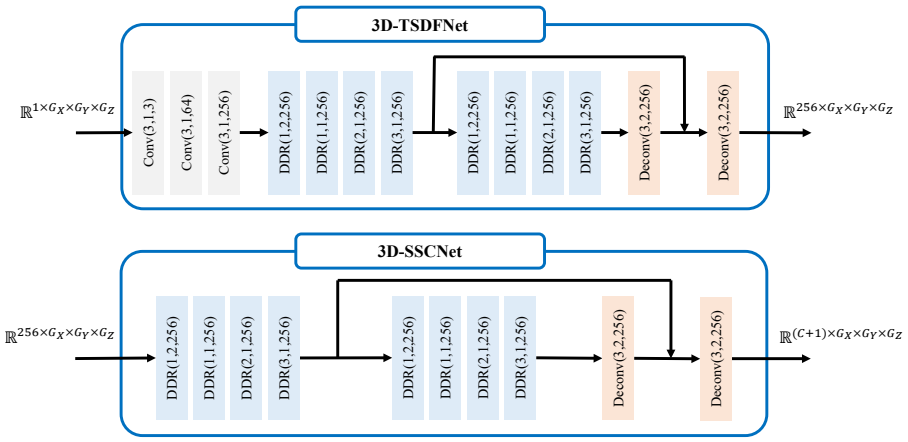

Figure S1. The architectures of 3D-TSDFNet and 3D-SSCNet. Conv(k, d, c) is a 3D convolutional layer with kernel size k, dilation d, and output channel c; DDR(d, r, c) is a DDR block [31] with dilation d, downsample rate r, and output channel c; and Deconv(k, R, c) is a 3D deconvolutional layer with kernel k, upsample rate R, and output channel c.

Methods	HR		LR		HR2LR	
	SC-IoU	SSC-mIoU	SC-IoU	SSC-mIoU	SC-IoU	SSC-mIoU
Baseline	71.9%	45.5%	80.1%	38.6%	-	-
CleanerS	**75.0%**	**47.7%**	**81.8%**	**40.6%**	81.1%	40.5%

Table S2. The ablation study results of our CleanerS with different 3D resolution on the test set of NYU [51].

LR results in a higher SC-IoU and a lower SSC-mIoU; 2) HR2LR performs even worse than LR2LR, which suggests the same resolution inputs enable a better knowledge distillation.

E. Feature-based KD *vs*. Data-based Denoise

This supplementary is for Section 5.4 of the main paper. In related works, there is a common practice to mitigate noises by using the corresponding clean data as learning targets [37, 62]. We validate here that it is not working for our cases. Specifically, we mitigate the noise in TSDF by using the noise-free TSDF-CAD input as a learning target. First, we add an extra prediction layer after 3D-TSDFNet, which with an input TSDF feature V_t^S. The prediction layer includes a DDR block [31] and a 3D convolutional layer, which outputs a 3-channel prediction (2-channel for the sign prediction and 1-channel for distance prediction). Then, we compare its results with our feature-based KD, in Table S3. As shown in Table S3, it achieves a limited performance gain (0.4% on SC-IoU and 0.4% on SSC-mIoU). In contrast, our feature-based KD significantly improves the SC-IoU. We think there are two reasons. 1) The TSDF-CAD features are task-oriented features and have a richer representation than the TSDF-CAD input. 2) Taking the TSDF-CAD input as a learning target needs extra prediction layers, which might distract the optimization.

Methods	Intermediate Supervision	SC-IoU	SSC-mIoU
Baseline	-	71.9 %	45.5%
CleanerS	TSDF-CAD input	72.3 %	45.9 %
CleanerS	TSDF-CAD feature	74.6%	46.0%

Table S3. The results of using different methods to mitigate noises in the TSDF on the test set of NYU [51]. We use either of TSDF-CAD inputs or TSDF-CAD features (output by the teacher network) to be an intermediate supervision in 3D-TSDFNet.

F. Correlation between the Noise Rate and Accuracy Degrade

This supplementary is for Section 5.4 of the main paper. To figure out the correlation between the noise rate and accuracy degradation, we perform the synthetic noise depth input by randomly adding either one or both of the zero noise and delta noise into the clean depth-CAD. The noise rate is gradually set to 20%, 50%, and 80%. Experimental results are given in Table S4. We can observe that 1) the higher the noise rate, the more degradation of the accuracy; 2) mixing both zero noise and delta noise will result in a complex noise and drops the performance drastically, especially for the SSC-mIoU.

	SC-IoU/SSC-mIoU (%)		
Noise Rate	20%	50%	80%
Zero Noise	↓2.2 / ↓1.9	↓2.4 / ↓2.6	↓2.7 / ↓3.3
Delta Noise	↓5.1 / ↓4.2	↓5.5 / ↓5.6	↓5.9 / ↓6.5
Mix Both Noises	↓8.7 / ↓5.0	↓12.6 / ↓5.5	↓14.0 / ↓6.8

Table S4. Results of synthetic depth with different noise rates.

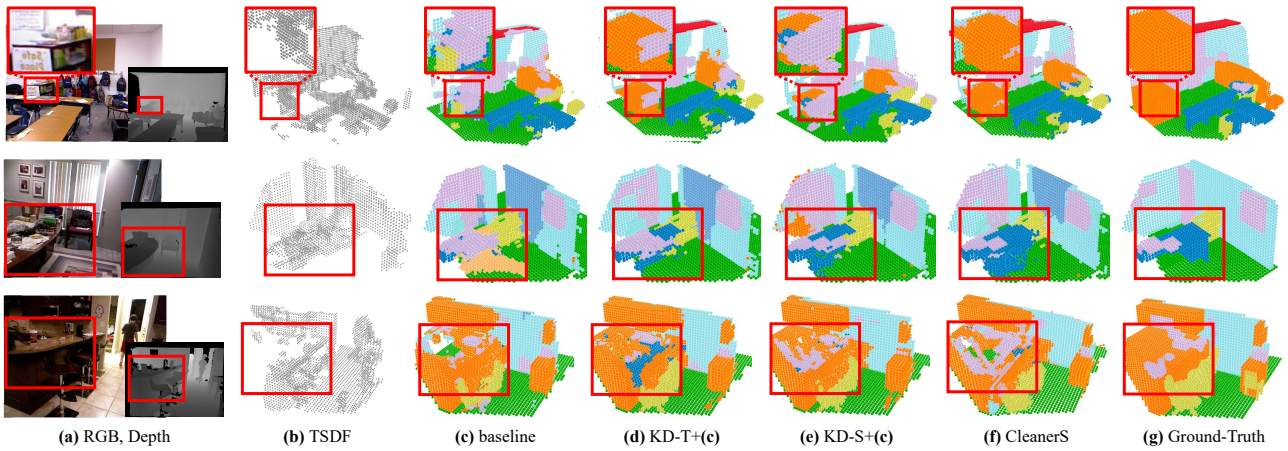

Figure S2. Visualization results for ablation study. The proposed feature-based KD (in (d)) and logit-based KD (in (e)) improve the baseline with better volumetric occupancy and semantics. Combining both (in (f)) achieves the best results.

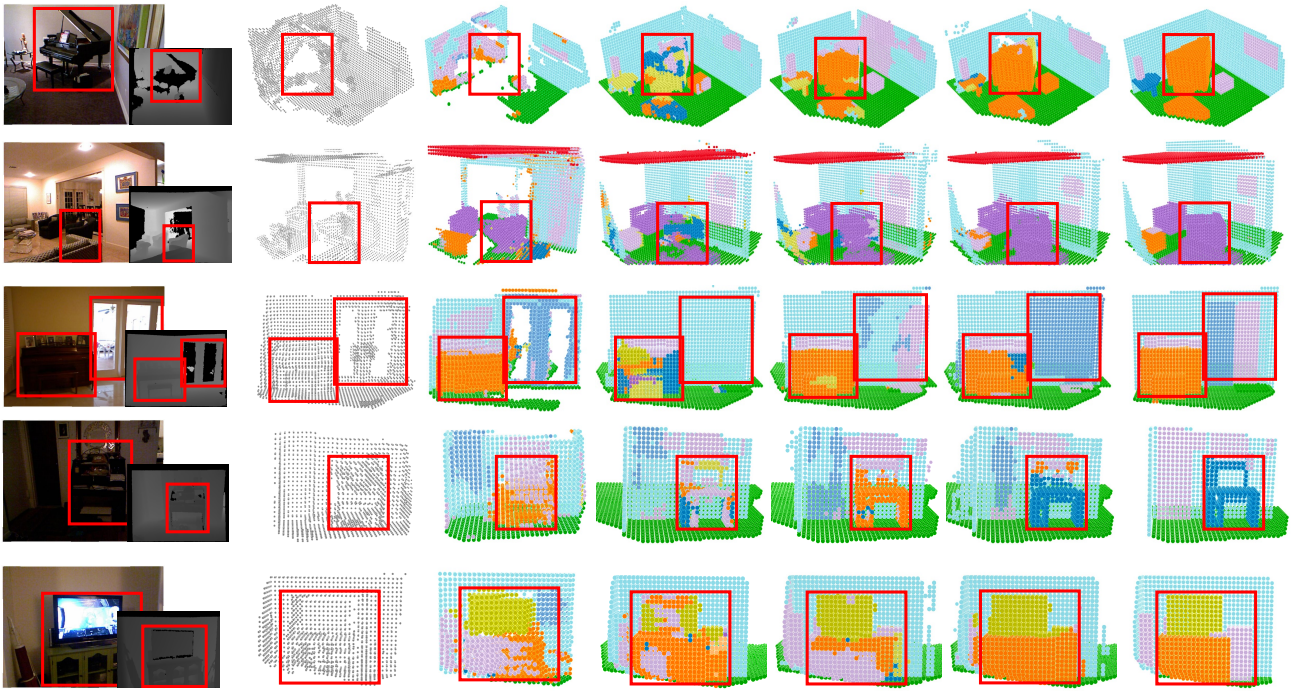

Figure S3. More qualitative comparisons with state-of-the-art methods, including SSCNet [52] and 3D-Sketch [4]. We present several challenging examples with zero noises and delta noises.

G. More Visualization Results

This supplementary is for Section 5.4 of the main paper. As shown in Figure S2, compared with the baseline method (in (c)), the cleaner surface distillation by feature-based KD (in (d)) helps to get cleaner occupancy predictions but may confuse the semantics. Combining it with the cleaner semantic distillation by logit-based KD (in (f)) can resolve this confusion.

In Figure S3, we supplement more visual examples compared to state-of-the-art methods.