Supplementary Materials for
Critical Regularizations for Neural Surface Reconstruction in the Wild

1. Additional Implementation Details

Distance of Boundary points. In Sec. 3.2, we calculate nearest neighbor distances $d(\mathbf{x}_B) = |(\mathbf{x}_B - \mathbf{x}_B^{nn}) \cdot \mathbf{n}_B^{nn}|$ for the boundary points $\mathbf{x}_B \in \mathcal{B}$, which may suffer from missing geometry or inaccurate estimation of normals. In practice, we draw k nearest neighbors and take the average of each distances. Also, we consider the angle between $(\mathbf{x}_B - \mathbf{x}_B^{nn})$ and $\mathbf{n}_B^{nn}$, and exclude the neighbors with large angle because they are very likely not the nearest neighbors. These two techniques enhance the robustness against missing points and noisy normal.

Computation of Hessian. In Sec. 3.3, we introduced that the Hessian matrices can be computed by auto-differentiation, which is similar to computing gradient of the SDF. This process is done by the service provided by Pytorch that a scalar function f can be differentiated with respect to each input (x, y, z), and the calculated gradient $(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z})$ can be added as a part of the computation graph. By further differentiating each entry of the gradient as $(\nabla\frac{\partial f}{\partial x}, \nabla\frac{\partial f}{\partial y}, \nabla\frac{\partial f}{\partial z})$ and concatenate the vectors together, we can get the Hessian matrix $\mathbf{H}f(x, y, z)$.

Parameter Tuning. In Sec. 4.1, we introduced the weights of the losses. This configuration can handle most of the cases when the input point cloud does not contain severe noise or incompleteness. Otherwise, we can tune the weights to get better reconstruction. If the geometry is complex and the render loss is not stable, we scale the gradient back-propagated from the differentiable intersection by 0.1. The three scenes *Church*, *Courthouse* and *Meetingroom* of Tanks and Temples dataset [4] are reconstructed under this setting. If the input point cloud is noisy, we change the weight of the normal data term λ_n to 0.1 in the second half of the training process, and let the render loss correct the surface normal. The four scenes *Barn*, *Caterpillar*, *Ignatius* and *Truck* of Tanks and Temples dataset are reconstructed under this setting.

2. Baselines

In this section, we provide more details on how to conduct experiments for the baseline methods on BlendedMVS and Tanks and Temples datasets.

SPSR and SSD. We test sPSR [3] and SSD [1] by their official implementation[1] and default setting. The resolution is the same as other experiments, which is described in Sec. 4.1.

NeuS. We use an unofficial implementation[2] for NeuS [6]. The system is trained for a fix number of 300k iterations, which takes 20 hours for each scene.

SIREN. We use the official implementation[3] for SIREN [5] The number of training steps is proportional to the number of points in the input point cloud. We add an upper bound of 50k steps. The whole training process takes at most 8 hours for each scene.

3. Details of Evaluation Metrics

In Sec. 4.1, we introduced the metric for evaluating the similarity between meshes, which is used in the comparison on BlendedMVS [7] and Tanks and Temples [4] datasets. Specifically, we first sample points from both meshes with uniform density and preserve their normal. The target density depends on the size of each scene. Then for each point in one point cloud, we find its nearest neighbor in the other one, and check the distance and the normal consistency. The threshold for distances is set to be three times as the downsample density, and the threshold for angular differences is 30°.

We run the same evaluation script, which is self-implemented, on both datasets. For Tanks and Temples, our script is different from the official one. First, we only run ICP between the camera trajectories to align the predicted mesh and the ground truth, so the alignments for all the methods are the same. Second, instead of voxel downsampling, we calculate nearest neighbor distances within the point cloud and discard one of the points in the pair with small distance.

[1] https://github.com/mkazhdan/PoissonRecon
[2] https://github.com/ventusff/neurecon
[3] https://github.com/vsitzmann/siren

4. Sensitivity to the Quality of Point Clouds

We conduct an experiment to examine the sensitivity to the quality of the input point clouds and the results are shown in Fig. 1. Assuming the point cloud from vis-MVSNet is state-of-the-art, we degrade its accuracy by randomly jittering the point positions within a certain radius.

As can be seen, the sPSR method is sensitive as the noise level goes up, but our method is robust against such noises and provides more consistent results both quantitatively and qualitatively. This again supports the argument that proposed regularizations are robust to random noises.

5. Ablation and Sensitivity Study of Regularizations

In Tab. 1, we provide a more comprehensive quantitative ablation and sensitivity study on DTU and Tanks and Temples datasets. Generally, different hyperparameters do have influence on the result. For DTU, we can lower the weight of Hessian or even disable it because the point cloud quality is relatively high. For Tanks and Temples, however, changing the weight of Hessian loss do have notable influence on the results. Apart from the Hessian loss, other configurations are worse than or comparable with the default setting. Overall, the default setting achieves good results in all the datasets (which is shown in the main paper). And we believe the default setting will perform well for other new data.

6. More Results

We provide more results of DTU [2], BlendedMVS [7] and Tanks and Temples [4] datasets in Fig. 2, 3, 4.

In the supplementary video, we show novel view rendering results, and the geometry with respect to training steps.

References

[1] Fatih Calakli and Gabriel Taubin. Ssd: Smooth signed distance surface reconstruction. In *Computer Graphics Forum*, volume 30, pages 1993–2002. Wiley Online Library, 2011. 1

[2] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *CVPR*, 2014. 2, 4, 5

[3] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013. 1

[4] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):78, 2017. 1, 2, 6

[5] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020. 1

[6] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. 2021. 1

[7] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *CVPR*, 2020. 1, 2, 6

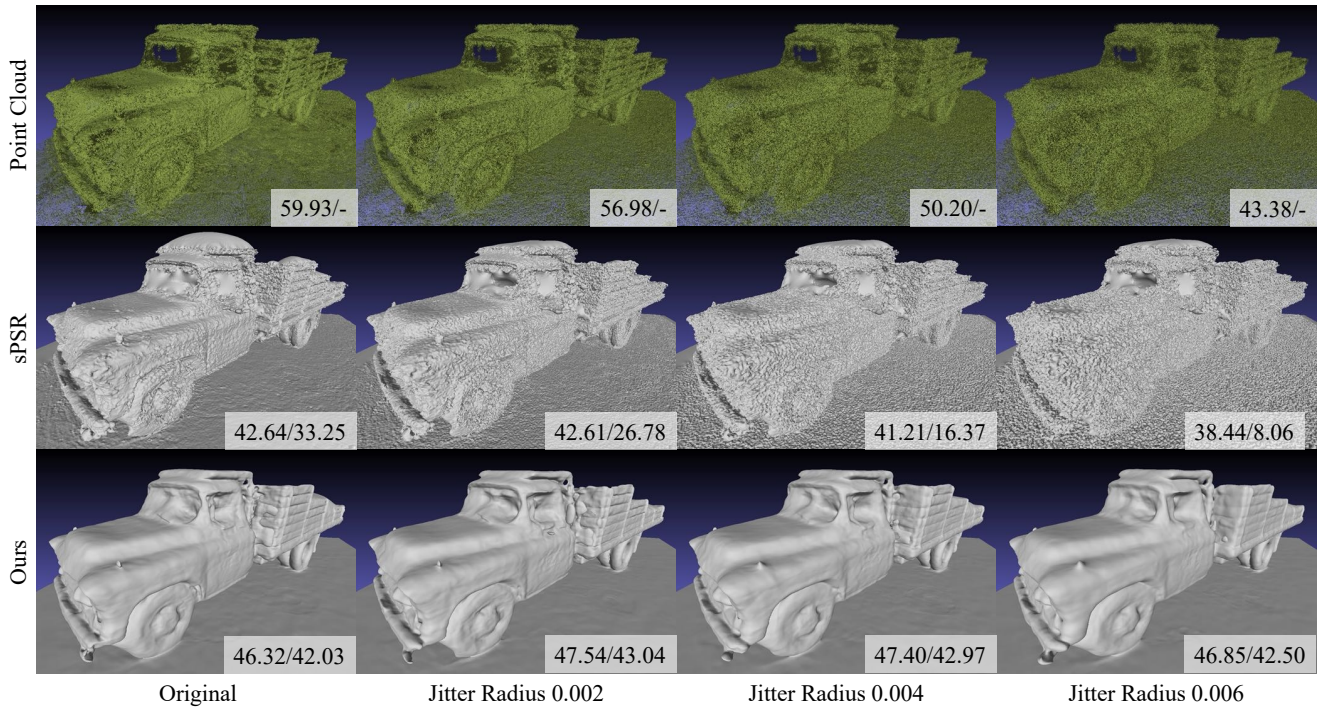

Figure 1. Influence of point cloud quality on mesh accuracy. The two numbers are the evaluation results with and without normal (details are in Sec. 4.1).

	No	Hessian L_H			Minimal Surface L_M			Sharpness of L_M		Default
	L_H and L_M	λ_H=1e-1	λ_H=1e-3	No L_H	λ_M=1e-1	λ_M=1e-3	No L_M	ϵ=1	ϵ=100	Config.
24	0.808	0.777	0.586	**0.573**	0.587	0.631	0.682	0.687	0.595	0.597
37	1.780	1.818	**1.326**	1.516	1.584	1.572	1.600	1.696	1.441	1.410
40	0.721	0.865	0.623	**0.610**	1.132	0.659	0.675	0.671	0.633	0.637
55	0.480	0.483	**0.390**	0.401	0.412	0.487	0.452	0.462	0.410	0.428
63	1.287	1.703	1.143	**1.108**	1.440	1.144	1.201	1.371	1.431	1.342
65	0.615	1.005	**0.611**	0.620	1.575	0.742	0.742	0.673	0.632	0.623
69	0.765	0.771	0.576	**0.551**	0.783	0.674	0.636	0.708	0.566	0.599
83	1.247	1.042	0.904	0.900	**0.890**	1.154	1.229	1.145	0.897	0.895
97	0.986	1.372	**0.848**	0.866	1.206	0.909	0.895	0.948	0.942	0.919
105	1.324	1.196	0.976	**0.940**	1.205	1.172	1.230	1.223	0.970	1.020
106	0.645	0.947	**0.535**	0.564	0.844	0.686	0.726	0.698	0.611	0.600
110	0.748	0.635	0.677	0.572	**0.537**	0.720	0.786	0.673	0.580	0.594
114	0.335	0.360	**0.293**	0.302	0.315	0.297	0.300	0.299	0.302	0.297
118	0.438	0.591	0.398	0.392	0.513	0.373	0.406	**0.370**	0.376	0.406
122	0.443	0.525	0.383	0.379	0.387	0.438	0.465	0.394	**0.379**	0.389
Mean	0.841	0.939	**0.685**	0.686	0.894	0.777	0.802	0.801	0.718	0.717
Caterpillar	14.33%	9.93%	16.67%	16.47%	14.26%	16.92%	16.64%	16.92%	**17.62%**	16.75%
Truck	33.47%	36.01%	40.78%	40.55%	36.40%	41.42%	41.86%	41.99%	41.24%	**42.03%**
Mean	23.90%	22.97%	28.72%	28.51%	25.33%	29.17%	29.25%	**29.45%**	29.43%	29.39%

Table 1. Ablation and sensitivity study on DTU and Tanks and Temples datasets. The value reported for DTU is the overall Chamfer distance (the lower the better), and for Tanks and Temples is the inlier percentage with normal criterion (the higher the better). The default configuration achieves overall good results.

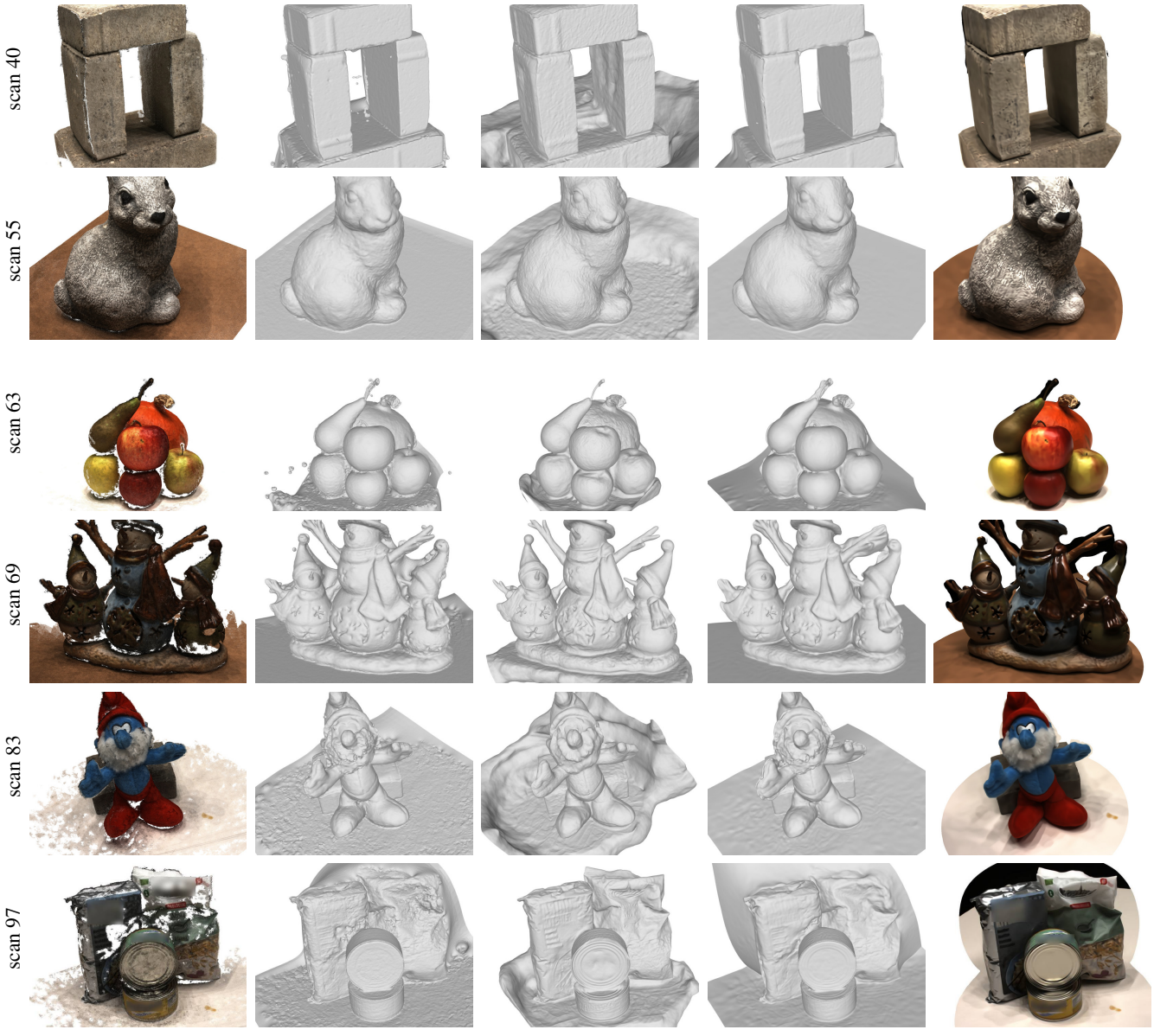

Figure 2. Qualitative results on DTU [2] dataset. Brand names are blurred.

Figure 3. Qualitative results on DTU [2] dataset (cont.).

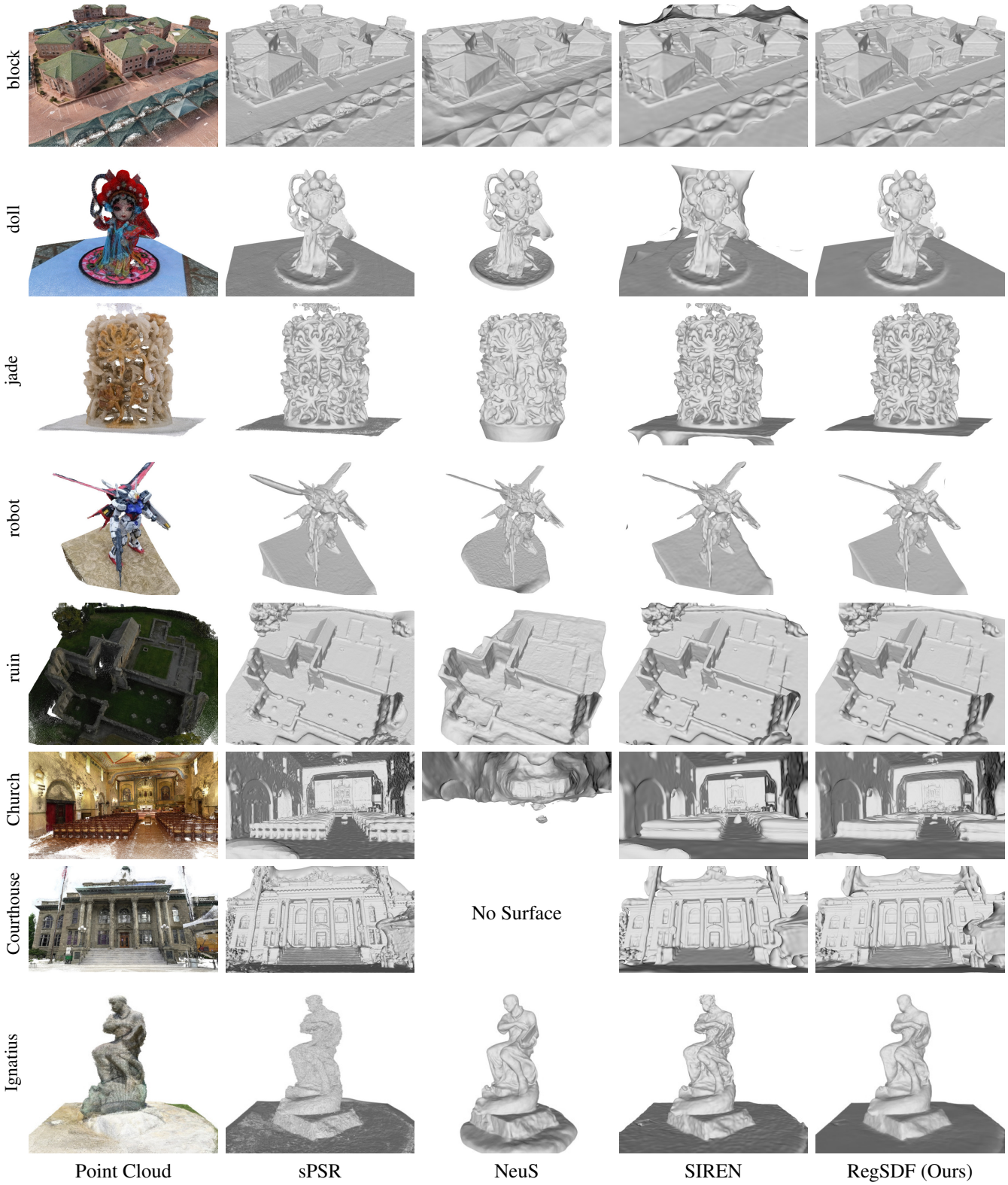

Figure 4. Qualitative results on BlendedMVS [7] and Tanks and Temples [4] dataset.