SketchSplat: 3D Edge Reconstruction via Differentiable Multi-view Sketch Splatting

Supplementary Material

Summary. In this supplementary material, we elaborate on the following topics:

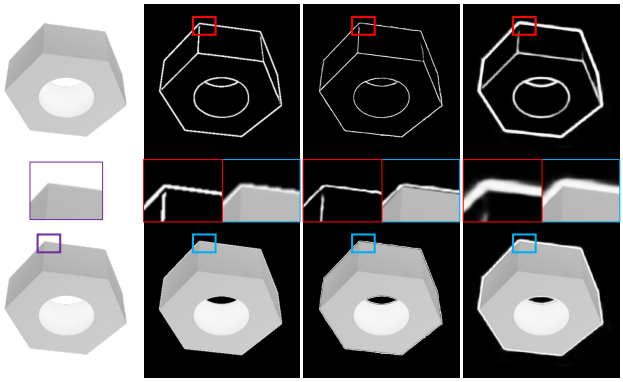
- Sec. 7 provides details of our proposed edge detector 2DGS-SN, along with visual comparisons against existing 2D edge detectors.
- Sec. 8 presents an alternative depth estimator for 2D edge detection.
- Sec. 9 evaluates the robustness of our method under noisy initialization.
- Sec. 10 discusses the necessity of the sketch filtering operation.
- Metric definitions used in the main paper are detailed in Sec. 11.
- Sec. 12 presents additional qualitative results, including 20 scenes from the ABC-NEF dataset.
- In Sec. 13, we report quantitative results on the DTU dataset.
- Sec. 14 compares our method with a feed-forward method.
- Finally, Sec. 15 discusses the limitations of our approach.

7. Details of 2D Edge Detector

We show the accuracy of 2DGS-SN by comparing with PiDiNet [9] and DexiNed [8]. From Fig. 5 (b), we can see 2DGS-SN produces accurate edges, where the object boundary locates at the center of the predicted edge pixels. Both PiDiNet and DexiNed (Fig. 5 (c, d)) detect edges with slight offsets, positioning them far from the object boundaries. According to Tab. 1 in the main paper, this slight offset introduces significant multi-view inconsistency and ambiguity, resulting in degraded performance. This also explains why EdgeGS [3] achieve much higher F10 and F20 scores than EMAP [6], but lower F5 scores.

8. Alternative 2D Edge Detector

The core insight of 2DGS-SN is that geometric cues (e.g., depth and normal) offer higher pixel-level accuracy and are thus more robust than pure learning-based edge detectors [8, 9]. Based on this observation, we demonstrate that other geometry estimation methods such as DepthPro [2] can be alternative components of our 2DGS-SN. We replace 2DGS depth maps with the ones from DepthPro and test its performance. DepthPro is significantly faster than 2DGS, as it avoids per-scene optimization. However, its accuracy is lower due to unstable edge detection caused by multi-view inconsistent depth scales (see Tab. 6). In contrast,



(a)RGB (b)2DGS-SN (c) DexiNed (d) PiDiNet

Figure 5. Visual comparison of different edge detection methods. Our 2DGS-SN provides more accurate edge detections results that align well with the object boundaries.

Method	Detector	A↓	C↓	R5↑	P5↑	F5↑
EdgeGS	2DGS-SN	7.4	7.2	75.7	86.9	80.3
Ours	2DGS-SN	**6.8**	5.8	**90.8**	**92.9**	**91.3**
EdgeGS	DepthPro-SN	10.6	7.0	75.5	81.8	77.9
Ours	DepthPro-SN	9.2	**5.4**	90.3	87.5	87.8

Table 6. **Ablation on 2D edge detector**. We ablate the depth prediction methods (2DGS [4], DepthPro [2]). 2DGS shows better performance since it produces multi-view consistent depth maps. All experiments use EdgeGS as initialization method.

2DGS [4] provides depth maps with a multi-view consistent scale, therefore a single threshold t_d in Eq. 3 is sufficient to detect consistent edges. Choosing 2DGS (higher accuracy) vs. DepthPro (faster) is therefore a trade-off between efficiency and quality.

9. Robustness to Initialization

To illustrate the robustness of our method, we inject Gaussian noise $\delta \sim \mathcal{N}(0, \sigma^2)$ to the parameters of the edges initialized from EdgeGS. Fig. 6 and Tab. 7 show that our method performs robust even with noisy initializations and only degrades when initial edge lengths become excessively large relative to the scene size, which rarely occurs in practice. Exploring alternative initialization methods would be an interesting topic for future work.

10. Necessity of Sketch Filtering

In the main paper, the proposed filtering operations seem to have little effect on the metrics (Tab. 3), as the edges initial-

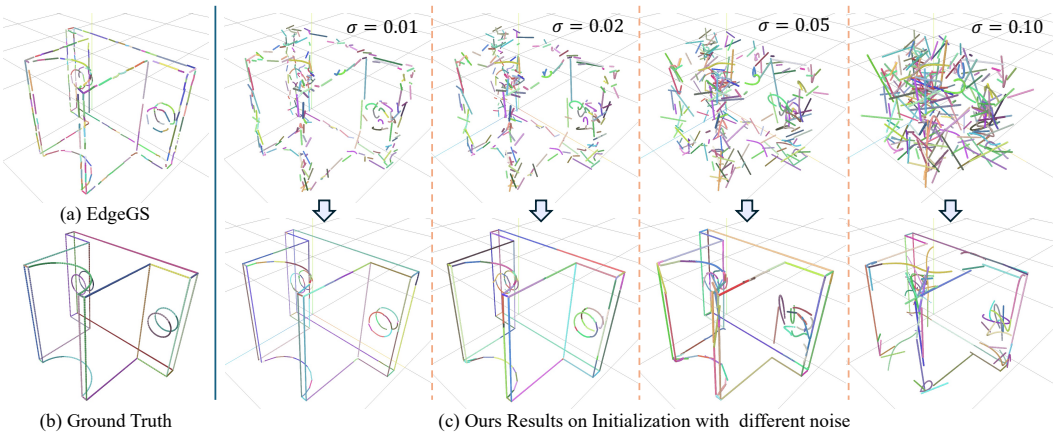


Figure 6. **Performance under noisy initialization**. We add Gaussian noise $\delta \sim \mathcal{N}(0, \sigma^2)$ to our initialization edges (from EdgeGS). On ABC-NEF, our method remains robust within reasonable noise levels but degrades when initial edge lengths become excessively large relative to the scene size $1m^3 (\sigma > 0.02)$, which is rare in our experiments.

Method	Noise (σ)	A↓	C↓	R5↑	P5↑	F5↑
EdgeGS	-	7.4	7.2	75.7	86.9	80.3
Ours	-	6.8	5.8	90.8	92.9	91.3
Ours	0.01	6.7	5.7	91.0	92.9	91.4
Ours	0.02	6.6	6.2	91.0	92.4	91.2
Ours	0.05	7.4	8.4	87.3	84.3	85.2
Ours	0.10	21.4	17.5	65.8	54.1	58.3

Table 7. **Performance under Noisy Initialization**. We present results with noisy initialization by adding Gaussian noise to the input edge parameters (on ABC-NEF dataset). Compared to the baseline EdgeGS, our method performs significantly better within a reasonable noise level ($\sigma < 0.05$ for a $1m^3$ scene). All results use 2DGS-SN as edge detector.

Method	Noise (σ)	Topo	A↓	C↓	R5↑	P5↑	F5↑
Ours	-	all	6.8	5.8	90.8	92.9	91.3
Ours	0.02	all	6.6	6.2	91.0	92.4	91.2
Ours	0.02	no filter	7.0	5.9	91.0	90.7	90.4

Table 8. **Effectiveness of Filtering Operations under Noisy Initialization**. We evaluate performance under noisy initialization by adding Gaussian noise to the input edge parameters on the ABC-NEF dataset. Without filtering, the method shows a clear drop in accuracy.

ized from EdgeGS are already clean. However, under poor initialization, filtering becomes essential for improving accuracy. We demonstrate this using a pseudo noisy initialization scenario, where we add Gaussian noise to the initialized edge parameters with $\sigma = 0.02$ and examine the effect of filtering. Tab. 8 shows that without filtering, the method suffers a noticeable accuracy drop and slightly higher completeness, confirming our assumption.

11. Details of Metrics

We provide the details of the used metrics in experiments. The ABC-NEF dataset [5, 11] provides ground-truth edges, which can be used for quantitative evaluation of 3D parametric edge reconstruction. To compute the metrics, we sample points along the predicted edges and compare them with points sampled at the same resolution from the ground-truth edges.

Accuracy (A) is defined as the average distance from predicted points to their nearest ground-truth points, while completeness (C) measures the average distance from ground-truth points to the closest predicted points. Lower values indicate better performance for these metrics.

For a given distance threshold τ, precision P(τ) represents the proportion of predicted points with at least one corresponding ground-truth point within τ. Conversely, recall R(τ) indicates the percentage of ground-truth points that have a predicted counterpart within the same threshold. The F-score F(τ) is the harmonic mean of precision and recall, formulated as:

$$F(\tau) = \frac{2P(\tau)R(\tau)}{P(\tau) + R(\tau)} \tag{1}$$

For precision, recall, and F-score, higher values indicate better performance. We report these metrics for thresholds of 5, 10, and 20 millimeters (mm). All of the CAD objects are scaled such that the longest edge of the bounding box is normalized to 1 meter (m).

12. More Visual Results

We provide more visualization results (20 scenes) on ABC-NEF dataset [5, 11] in Fig. 7, Fig. 8, Fig. 9, and Fig. 10. These figures show that our method achieves state-of-the-art accuracy and completeness compared to existing approaches.

Scan	LIMAP [7]		NEF [11]		NEAT [10]		EMAP [6]		EdgeGS* [3]		Ours		Ours (Topo)	
	R5↑	P5↑	R5↑	P5↑	R5↑	P5↑	R5↑	P5↑	R5↑	P5↑	R5↑	P5↑	R5↑	P5↑
37	75.8	74.3	39.5	51.0	63.9	**85.1**	62.7	83.9	**79.4**	76.6	78.7	80.2	78.2	78.9
83	75.7	50.7	32.0	21.8	72.3	52.4	72.3	61.5	**77.8**	62.6	76.5	**64.0**	74.2	62.8
105	**79.1**	64.9	30.3	32.0	68.9	73.3	78.5	**78.0**	72.6	68.8	74.0	72.0	71.6	72.3
110	79.7	65.3	31.2	40.2	64.3	**79.6**	**90.9**	68.3	83.4	60.3	85.0	66.8	83.5	66.1
118	59.4	62.0	15.3	25.2	59.0	71.1	**75.3**	**78.1**	74.5	68.6	73.5	70.5	71.2	70.1
122	79.9	79.2	15.1	29.1	70.0	82.0	85.3	82.9	85.0	82.7	**86.8**	**83.0**	83.6	81.9
mean	74.9	66.1	27.2	33.2	66.4	73.9	77.5	**75.4**	78.8	70.0	**79.1**	72.8	77.0	72.0

Table 9. 3D Edge Reconstruction on the DTU dataset [1]. Our SketchSplat performs on par with EMAP [6] and EdgeGS* [3] (EdgeGS* denotes the results reproduced with the code released by the authors [3]). Ours(Topo): ours with topology control.

Method	Input	A↓	C↓	R5↑	P5↑	F5↑
EMAP	2DGS-SN	8.8	7.9	63.5	70.4	66.3
EdgeGS	2DGS-SN	7.4	7.2	75.7	86.9	80.3
Ours	2DGS-SN	**6.8**	**5.8**	**90.8**	**92.9**	**91.3**
NerVE (PWL)	2DGS	18.0	35.7	40.5	62.3	48.4
NerVE (CAD)	2DGS	15.3	117.3	24.0	57.7	32.3

Table 10. **Comparison with feed-forward method NerVE on ABC-NEF dataset**. For NerVE, we train a 2DGS and extract surface points as input. It turns out that NerVE is sensitive to noise in the input and produces incomplete and inaccurate edges. Our method use EdgeGS as initialization.

13. Quantitative Evaluation on DTU dataset

Tab. 9 shows the evaluation results on six scenes in the DTU dataset [1]. We reproduce EdgeGS [3] using its released code and configuration (denoted as EdgeGS*) to initialize sketches for our method. As shown in Tab. 9, SketchSplat without topological operations achieves comparable results to EMAP [6] and slightly outperforms EdgeGS, demonstrating its effectiveness in correcting misalignment between 3D edges and 2D edge images. However, incorporating topological operations slightly reduces recall, as merging reduces redundant edges, whereas the pseudo ground-truth favors duplicated edges in P5 and R5 due to the thickness of the GT annotations (see Fig. 4 in the main paper).

14. Comparison to Feed-forward Baseline

We also compare against a feed-forward 3D line detection method. To evaluate methods like NerVE [12], we first reconstruct a point cloud from the input images (e.g., using 2DGS), then input it to the pretrained NerVE model to predict 3D edges. As shown in Tab. 10, NerVE performs significantly worse than our method and the baselines, suggesting that optimization-based methods remain more robust and generalize better, while feed-forward approaches are more sensitive to noisy inputs.

15. Limitations

Objects with thin structures. Though our method performs well on small structure (like scene 7551 in ABC-NEF dataset), our merge mechanism may fail when edges with distance close to or lower than our merging threshold (such as the case shown in the first raw of Fig. 11).

Wrong supervision. Like previous methods [3, 6, 11], our approach also encounters the challenge of incorrect edge supervision. During rendering, both visible and invisible edges from a given view are projected onto 2D images. As a result, some invisible edges may receive positive supervision, leading to misplaced edges that are difficult to filter out. This issue is illustrated in the second row of Fig. 11.

Edge detection Although the proposed edge detector 2DGS-SN is accurate, we found when both depth reconstruction and normal estimation fail, some small structure such as small circles will not be detected in 2D edge images. This leads to edge missing in the reconstructed 3D edges. Further improving the edge detection method may resolve this problem.

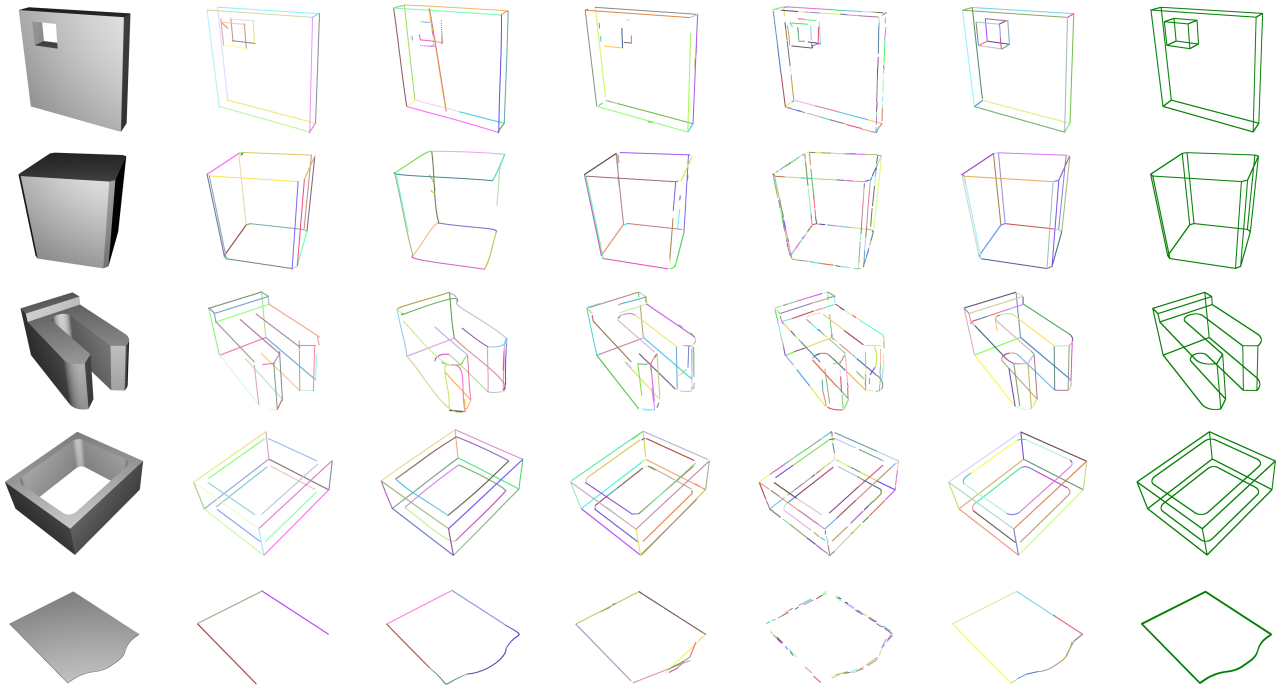
References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120:153–168, 2016. 3

[2] Aleksei Bochkovskii, AmaÃĢl Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024. 1

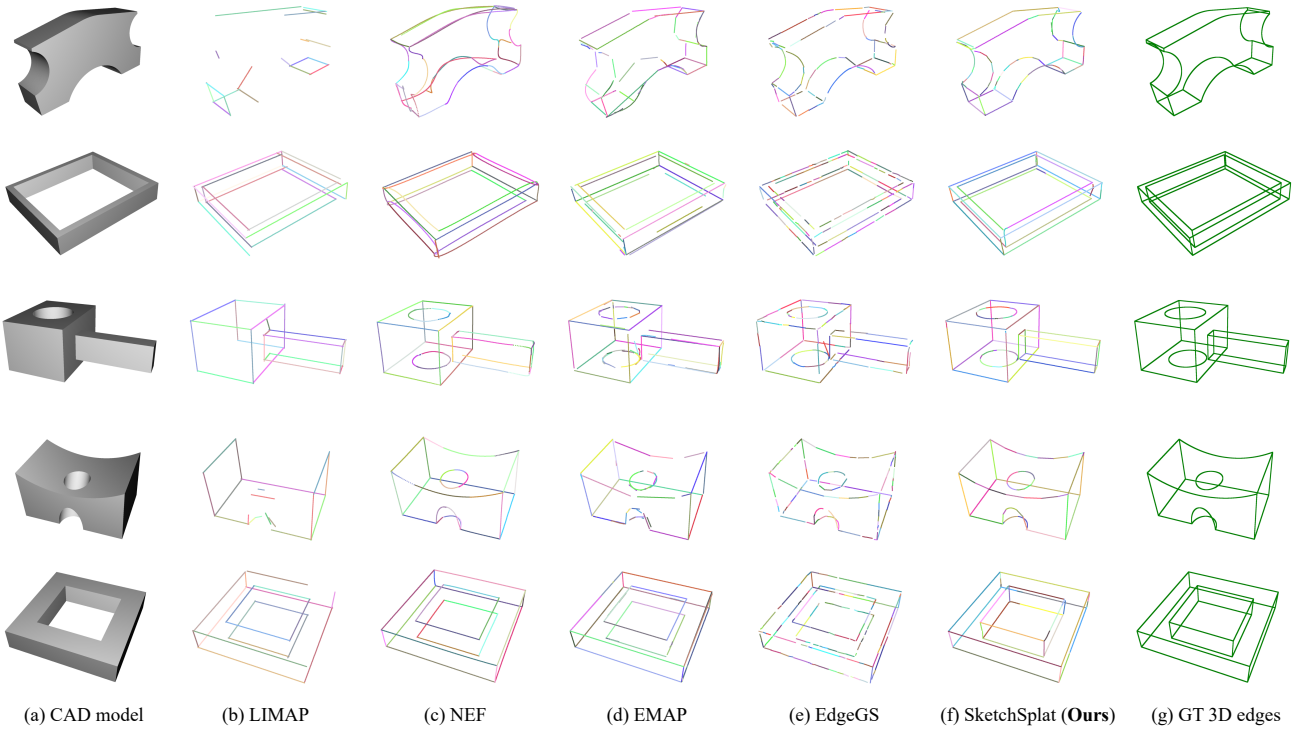
[3] Kunal Chelani, Assia Benbihi, Torsten Sattler, and Fredrik Kahl. Edgegaussians–3d edge mapping via gaussian splatting. *arXiv preprint arXiv:2409.12886*, 2024. 1, 3

[4] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024. 1



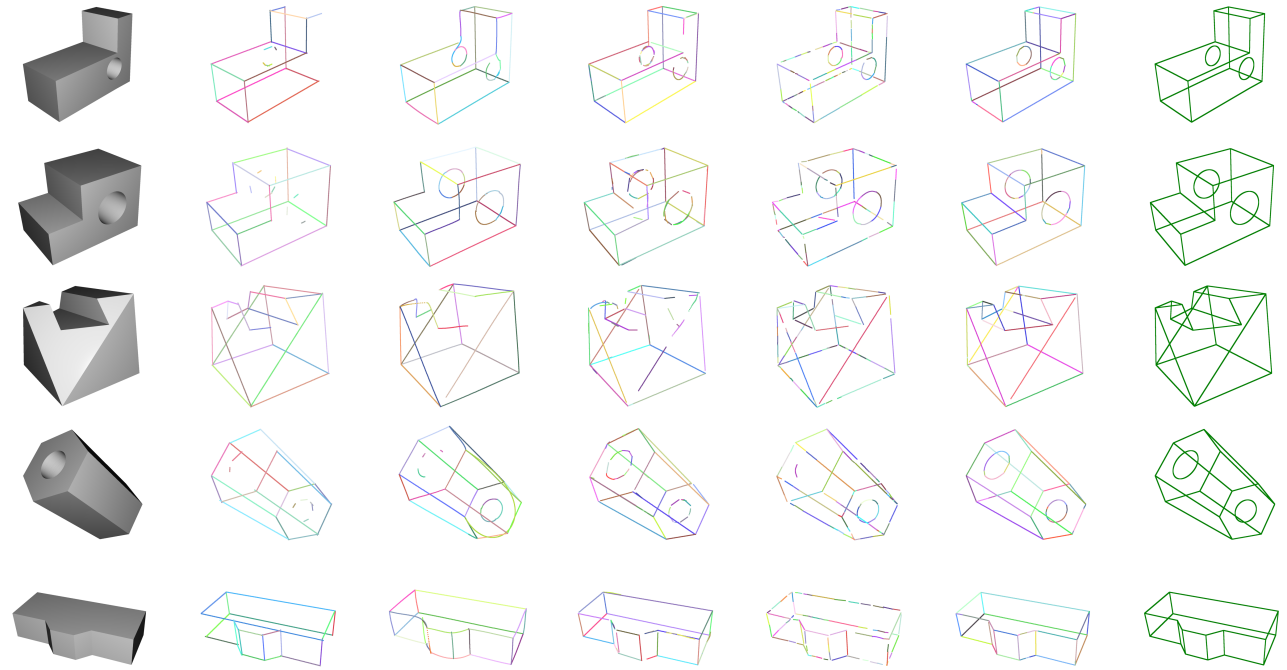
(a) CAD model (b) LIMAP (c) NEF (d) EMAP (e) EdgeGS (f) SketchSplat (**Ours**) (g) GT 3D edges

Figure 7. More qualitative results on ABC-NEF dataset [11].



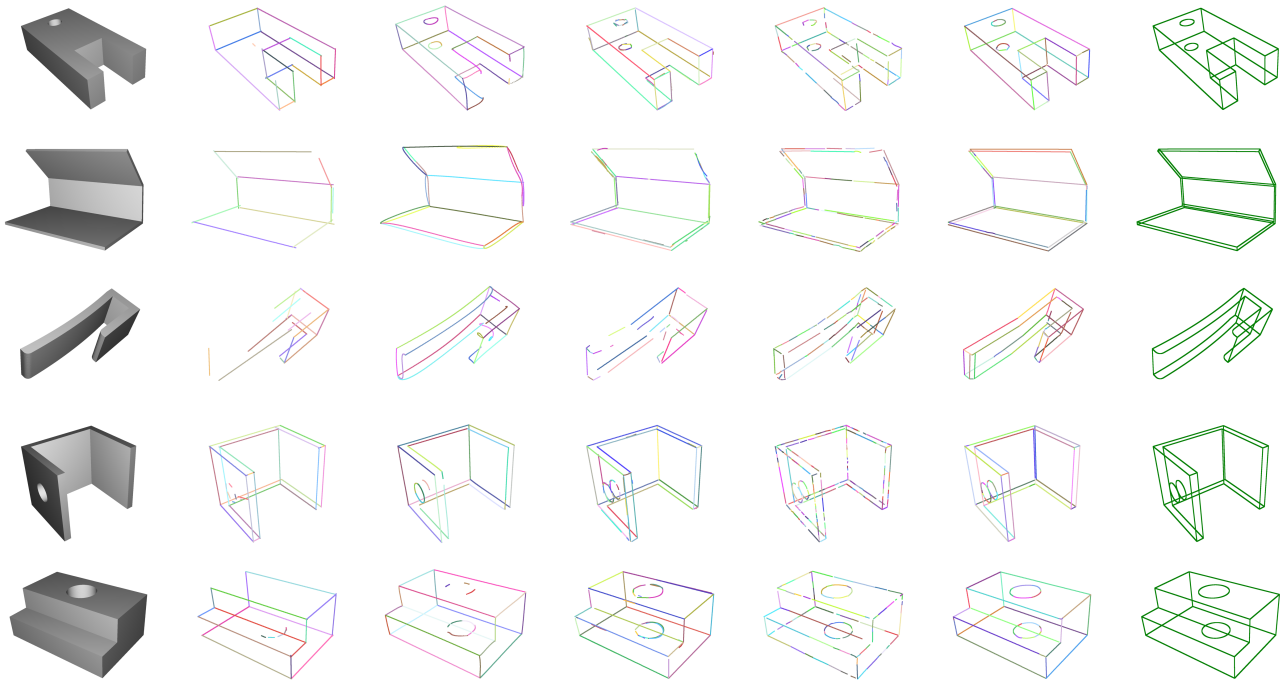
(a) CAD model (b) LIMAP (c) NEF (d) EMAP (e) EdgeGS (f) SketchSplat (**Ours**) (g) GT 3D edges

Figure 8. More qualitative results on ABC-NEF dataset [11].



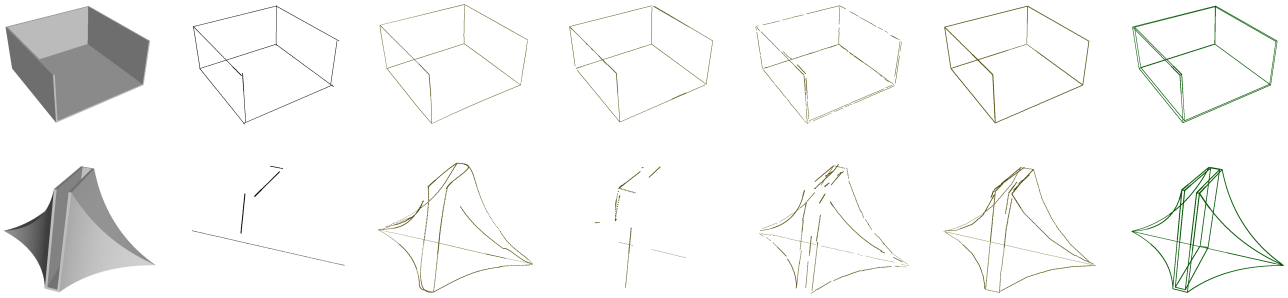
(a) CAD model (b) LIMAP (c) NEF (d) EMAP (e) EdgeGS (f) SketchSplat (**Ours**) (g) GT 3D edges

Figure 9. More qualitative results on ABC-NEF dataset [11].



(a) CAD model (b) LIMAP (c) NEF (d) EMAP (e) EdgeGS (f) SketchSplat (**Ours**) (g) GT 3D edges

Figure 10. More qualitative results on ABC-NEF dataset [11].



(a) CAD model (b) LIMAP (c) NEF (d) EMAP (e) EdgeGS (f) **Ours** (g) GT 3D edges

Figure 11. Failure cases. Our method struggles with very thin structures and the wrong edges.

[5] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9601–9611, 2019. 2

[6] Lei Li, Songyou Peng, Zehao Yu, Shaohui Liu, Rémi Pautrat, Xiaochuan Yin, and Marc Pollefeys. 3d neural edge reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21219–21229, 2024. 1, 3

[7] Shaohui Liu, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. 3d line mapping revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21445–21455, 2023. 3

[8] Xavier Soria Poma, Edgar Riba, and Angel Sappa. Dense extreme inception network: Towards a robust cnn model for edge detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1923–1932, 2020. 1

[9] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikäinen, and Li Liu. Pixel difference networks for efficient edge detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5117–5127, 2021. 1

[10] Nan Xue, Bin Tan, Yuxi Xiao, Liang Dong, Gui-Song Xia, Tianfu Wu, and Yujun Shen. Neat: Distilling 3d wireframes from neural attraction fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19968–19977, 2024. 3

[11] Yunfan Ye, Renjiao Yi, Zhirui Gao, Chenyang Zhu, Zhiping Cai, and Kai Xu. Nef: Neural edge fields for 3d parametric curve reconstruction from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8486–8495, 2023. 2, 3, 4, 5

[12] Xiangyu Zhu, Dong Du, Weikai Chen, Zhiyou Zhao, Yinyu Nie, and Xiaoguang Han. Nerve: Neural volumetric edges for parametric curve extraction from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13601–13610, 2023. 3