Appendix for "Eve3D: Elevating Vision Models for Enhanced 3D Surface Reconstruction via Gaussian Splatting"

Overview

This appendix contains supplementary material that supports and extends the findings presented in the main paper. We begin in Sec. A with detail descriptions of our experimental setup. Sec. B provides further analysis, including ablation studies to better understand different components of our approach. Additional qualitative results of 3D reconstructions produced by Eve3D are presented in Sec. C. Finally, in Sec. D, we reflect on the broader impact of our methodology.

A Details of Experimental Setting

A.1 Datasets

DTU. The DTU dataset provides ground-truth point clouds for evaluating object-level reconstruction quality. Following prior works [18, 55, 6], we use 15 scans (24, 37, 40, 55, 63, 65, 69, 83, 94, 102, 106, 110, 114, 118, and 122) to assess surface reconstruction performance. In our experiments, all images from each scan are used, downsampled to half resolution for training.

Tanks and Temples. The Tanks and Temples dataset includes ground-truth points for evaluating surface reconstruction in both indoor and outdoor scenes. In line with previous studies [18, 55, 6], we conduct experiments on six scenes: Barn, Caterpillar, Courthouse, Ignatius, Meetingroom, Truck. For each scene, we use all available images, downsampled to half resolution for training.

Mip-NeRF360. Since Mip-NeRF360 does not provide ground-truth points for surface reconstruction evaluation, we instead use it to evaluate novel view synthesis performance. We adopt the standard train/test splits from prior works [18, 55, 6]. For outdoor scenes (bicycle, flowers, garden, stump, treehill), images are downsampled to quarter resolution. For indoor scenes (bonsai, counter, kitchen, room), images are downsampled to half resolution, consistent with previous studies [18, 55, 6]. For mesh reconstruction visualizations, we train models using only the training split images.

A.2 Implementations

Hyperparameters. Our base model adopts the plane depth definitions [6] to render depth. We constrain the shortest axis scale of Gaussians to zero to make Gaussians as close to planes. We adopt a depth-normal consistency loss [18, 55, 6] to encourage the consistent representations of rendered depth and normal vectors. The learnable prior depth maps are initialized using predictions from a depth estimation model and optimized with a learning rate of (1×10^{-4}). For depth map initialization, we sample 500,000 points for DTU scans and 1,000,000 points for both Tanks and Temples scenes and the Mip-NeRF360 dataset.

Eve3D. We train Eve3D for a total of 30,000 iterations. Prior depth supervision is introduced starting from iteration 500. We set the T_{joint} to 7000. The shortest axis scale loss is applied from the beginning of training. The depth-normal consistency is activated starting at iteration 7000. The densification process for 3D Gaussians begins at iteration 500 and ends at iteration 15,000.

Eve3D-*fast*. We train Eve3D-*fast* for a total of 5,000 iterations. Prior depth supervision is introduced starting from iteration 500. We set $T_{joint} = 1000$. The shortest axis scale loss is applied from the beginning of training. The depth-normal consistency is enabled from iteration 1000. The densification of 3D Gaussians begins at the 500 iteration and concludes at iteration 4000.

Mesh Extraction. We render depth maps from the 3D Gaussians and apply Truncated Signed Distance Function (TSDF) fusion to extract surface meshes. For scenes captured with front-facing cameras (DTU), we use unbounded mesh extraction and set the voxel size to 0.002. For scenes captured by surround-view cameras (e.g., Tanks and Temples, Mip-NeRF360), we use bounded mesh extraction, where the voxel size is set to the maximum scene extent divided by 2048. For indoor scenes, scene bounds are estimated from camera trajectories, while for outdoor scenes, they are estimated from the reconstructed point clouds.

Table 7: **Direct Comparisons between Eve3D and GS2Mesh.** Methods are trained with mini-splatting2 to render stereo views and FoundationStereo to predict depth maps.

Methods	Barn	Caterpillar	Courthouse	Ignatius	Meetingroom	Truck	Mean ↑	Time
GS2Mesh [45]	0.51	0.27	0.08	0.61	0.19	0.41	0.35	12 m
Eve3D-*fast* (Ours)	0.69	0.44	0.34	0.82	0.41	0.62	0.56	20 m
Eve3D (Ours)	0.70	0.48	0.35	0.83	0.46	0.66	0.58	1.2 h

Table 8: **Comparisons to PGSR with Depth Priors.** We train PGSR with the FoundationStereo initialization and supervision, which is the same to the supervision used in Eve3D. The difference is that Eve3D uses the Prior-involved bundle adjustment with joint optimization, while PGSR uses multi-view consistency between neighbor-view rendering results to maintain the multi-view consistency. With the same depth prior, Eve3D shows significantly better convergence than PGSR.

Total	PGSR + FoundationStereo			Eve3D (Ours)		
Iterations	Precision ↑	Recall ↑	F1 Score ↑	Precision ↑	Recall ↑	F1 Score ↑
3k	0.494	0.531	0.485	0.500	0.570	0.525
5k	0.492	0.574	0.519	0.532	0.600	0.555
10k	0.505	0.586	0.532	0.546	0.611	0.568
30k	0.552	0.609	0.571	0.553	0.631	0.581

Overlapping Score. Following [51], for a reference view $\boldsymbol{V}_i$, we compute the overlapping score $s(i,j) = \sum_{\boldsymbol{X}} \eta(\theta_{ij}(\boldsymbol{X}))$ for its neighboring view $\boldsymbol{V}_j$, and $\boldsymbol{X}$ is a 3D point which is observed by both views $\boldsymbol{V}_i$ and $\boldsymbol{V}_j$. In detail, $\theta_{ij}(\boldsymbol{X}) = (180/\pi)\arccos((\boldsymbol{t}_i - \boldsymbol{X}) \cdot (\boldsymbol{t}_j - \boldsymbol{X}))$ is the baseline angle and $\boldsymbol{t}$ represents the camera center. $\eta(\cdot)$ is piece-wise Gaussian function that favors a certain baseline angle θ_0:

$$\eta(\theta) = \begin{cases} \exp\left(-\frac{(\theta-\theta_0)^2}{2\sigma_1^2}\right), & \text{if } \theta \leq \theta_0 \\ \exp\left(-\frac{(\theta-\theta_0)^2}{2\sigma_2^2}\right), & \text{if } \theta > \theta_0 \end{cases} . \tag{16}$$

where θ_0, σ_0 and σ_1 are hyper-parameters and are set to 5, 1, and 10 respectively.

B Additional Analysis

B.1 Direct Comparisons with Improved GS2Mesh

Eve3D leverages the rendering capabilities of 3DGS to generate stereo pairs and infer depth priors—similar in spirit to GS2Mesh [45]. However, while GS2Mesh directly uses stereo depth maps to reconstruct the meshes, our apporoach treats stereo depth maps as priors, which are then jointly optimized along with the 3D Gaussian via our proposed framework.

Although the original GS2Mesh significantly underperforms compared to Eve3D in terms of reconstruction accuracy (see Tab. 2 in the main paper), the reader might argue that the discrepancy could be due to the use of a different stereo backbone-DLNR [59]. Therefore, to fully assess the superiority of our methodology over the direct fusion of stereo priors, we re-implement GS2Mesh using the same settings as Eve3D.

Accordingly, we provide a new comparison between GS2Mesh and Eve3D in Tab. 7 on the Tanks and Temples dataset. In this experiment, both methods use Mini-Splatting [13] to render pseudo stereo views and FoundationStereo [44] to estimate depth maps. Even with such high-quality priors, GS2Mesh struggles to reconstruct accurate surfaces—particularly in complex scenes like Meetingroom and Courthouse. In contrast, our Eve3D-fast, with only eight minutes of additional optimization, achieves significantly better reconstruction quality.

Table 9: **Ablation Study.** Impact of vision model choice.

Prior Source	Eve3D (Ours)	Precision ↑	Recall ↑	F1 Score ↑
FoundationStereo [44]	✗	0.431	0.519	0.463
FoundationStereo [44]	✓	0.553	0.631	0.581
Stereo Anywhere [3]	✗	0.410	0.470	0.431
Stereo Anywhere [3]	✓	0.533	0.600	0.555
MVSAnywhere [19]	✗	0.430	0.517	0.462
MVSAnywhere [19]	✓	0.506	0.578	0.532
MVSFormer [4]	✗	0.450	0.548	0.483
MVSFormer [4]	✓	0.493	0.598	0.528
OMNI-DC [64]	✗	0.298	0.390	0.330
OMNI-DC [64]	✓	0.448	0.540	0.479

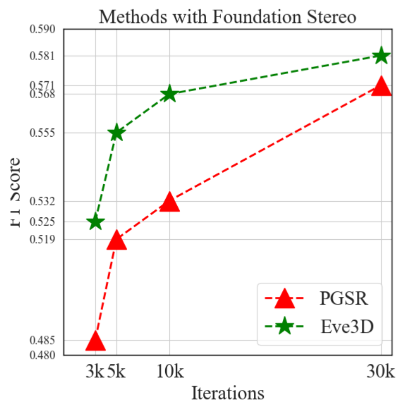

Figure 8: **Comparisons of PGSR and Eve3D with FoundationStereo priors – convergence speed.**
.

B.2 Comparisons to PGSR with Depth Prior

PGSR [6] sets a strong baseline for surface reconstruction with the proposed multi-view consistency based on rendered results. Compared to PGSR, Eve3D uses the prior-depth involved bundle adjustment to enhance the multi-view consistency, incorporating more than one neighbor view in one training loop. Moreover, Eve3D leverages bundle adjustment not only to refine the 3D Gaussians but also to optimize the depth priors themselves. When the initial priors are reasonably accurate at a coarse level, they can be quickly refined into multi-view consistent priors through local bundle adjustment. This leads to better convergence behavior compared to enforcing multi-view constraints directly on rendering outputs, as done in PGSR. As shown in Table8 and Figure8, when using the same FoundationStereo priors, Eve3D achieves significantly faster and more stable convergence than PGSR.

Table 10: **Ablation Study.** Impacts of baseline length.

Baseline Length	3 % camera extent	7 % camera extent	10 % camera extent
F1 score ↑	0.580	0.581	0.581

B.3 Additional Ablation Study on Method Components

Prior Depth Types. We evaluate the generalizability of our method across alternative sources for depth priors, replacing those obtained from FoundationStereo applied to rendered stereo images with different approaches: i) using the Stereo Anywhere model [3]; ii) using depth maps from multi-view stereo (MVSAnywhere [19] and MVSFormer [4]) or iii) a depth completion network (OMNI-DC [64]) applied to sparse depth points extracted from COLMAP. All these methods predict depth maps

Table 11: **Ablation study.** Impact of the neighbors in local bundle adjustment.

Number of Neighbors	Precision ↑	Recall ↑	F1 Score ↑	Training Time
1	0.544	0.625	0.573	50 m
2	0.551	0.627	0.578	1 h
4	0.553	0.631	0.581	1.2 h
8	0.554	0.631	0.582	1.5 h

at the correct metric scale, although through different working principles: stereo models estimate disparity maps from stereo images rendered using intrinsics and extrinsics at the same scale as the pretrained 3DGS, then triangulate depth using the known focal length and baseline; multi-view stereo methods exploit the same camera poses used to optimize 3DGS, thus predicting depth at consistent scale; depth completion models densify sparse COLMAP points used to initialize 3DGS, maintaining their metric scale. Despite variations in depth accuracy across these sources, our joint optimization consistently improves reconstruction performance (Table 9), demonstrating robustness to different depth initializations and strong generalization. We emphasize that all foundation models used in our experiments are applied zero-shot without fine-tuning.

We also highlight how, at the current stage, rendering stereo images to extract priors through FoundationStereo [44] represents the optimal choice; nonetheless, we don't exclude that future advances in multi-view stereo or depth completion may lead to stronger models, thus making Eve3D no longer require rendering stereo images to get priors. Confirming this hypothesis in future research would allow for further improve Eve3D performance – as it can be seamlessly integrated even with future, more advanced networks.

Virtual Camera Baseline Length. For stereo pair generation, we set the baseline length to 7% of the scene radius across all experiments. To assess the sensitivity of our method to this choice, we evaluate different baseline lengths in Table 10. The results demonstrate that our method is robust to baseline selection, with F1 scores remaining consistent (0.580-0.581) across baseline lengths ranging from 3% to 10% of the camera extent. This robustness stems from our joint optimization and local bundle adjustment, which enforce multi-view consistency constraints that naturally compensate for variations in initial stereo depth estimates.

Number of Neighbors in Local Bundle Adjustment. We study the impact of the number of neighboring views used in local bundle adjustment in Table 11. As the number of neighbors increases, the training time also grows due to the additional computation. At the same time, the inclusion of more diverse viewing angles in each optimization step enhances the geometric accuracy of both the 3DGS representation and the optimized depth priors. However, beyond a certain point, the benefit of adding more neighbors saturates. This is because additional views with weaker co-visibility relationships contribute limited new information, resulting in diminishing returns in geometric improvement.

B.4 Training Time Breakdown

All training times reported in the main paper include the complete pipeline: 3DGS pretraining, stereo pair rendering, FoundationStereo predictions, and final 3DGS training. We provide a detailed breakdown of preprocessing and training times for transparency.

Tanks and Temples: On average, it takes 4 minutes to pretrain 3DGS with Mini-Splatting [13], 8 minutes for stereo pair rendering and FoundationStereo depth predictions, and 60 minutes for final Eve3D training (8 minutes for Eve3D-*fast*). The total time is therefore 1.2 hours for Eve3D and 20 minutes for Eve3D-*fast*.

DTU: On average, it takes 3 minutes to pretrain 3DGS with Mini-Splatting, 4 minutes for stereo pair rendering and FoundationStereo predictions, and 8 minutes for final training. The total time is 15 minutes.

C Additional Visualization Results

We provide additional qualitative results of Eve3D in Fig. 9, 10, and 11, which illustrate the surface reconstructions on the Tanks and Temples, DTU, and Mip-NeRF360 datasets, respectively.

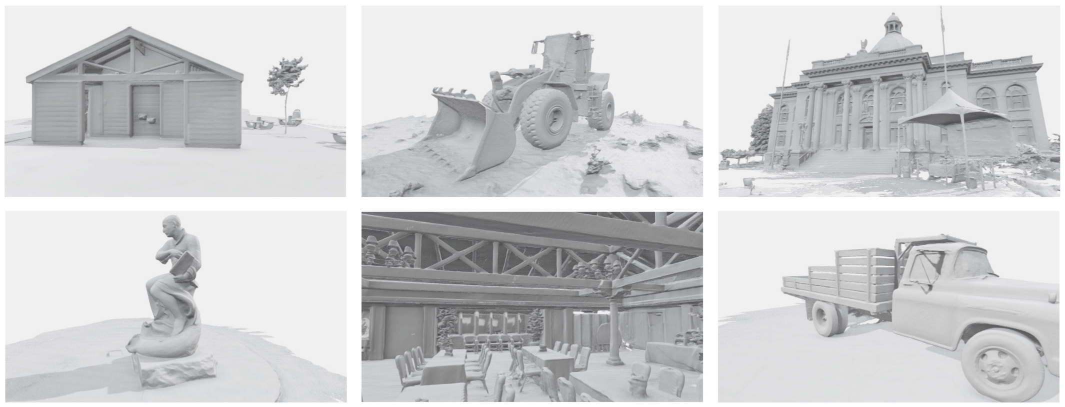

Figure 9: **Qualitative Visualizations on the Tanks and Temples Dataset.**

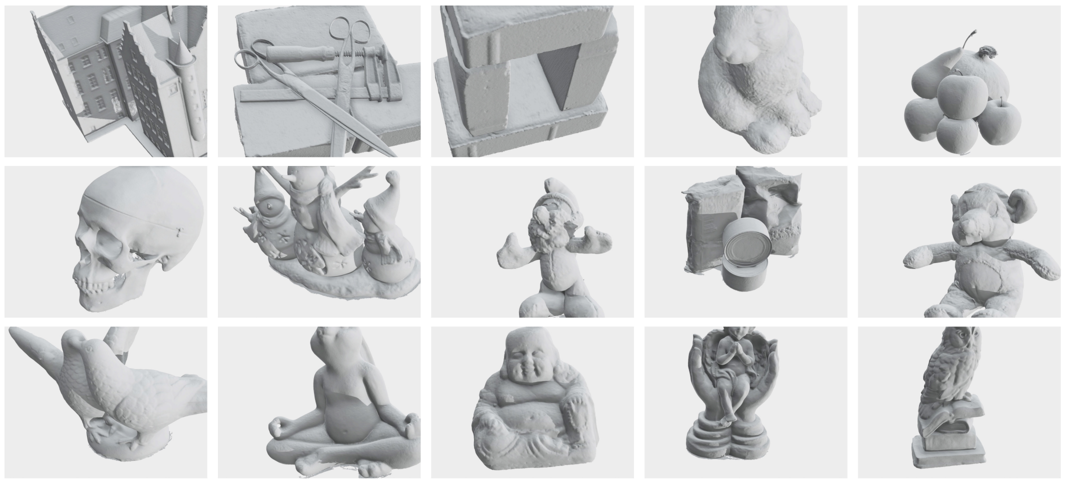

Figure 10: **Qualitative Visualizations on the DTU Dataset.**

D Broader Impact Statement

Eve3D sets a new state-of-the-art in 3D surface reconstruction, achieving unprecedented accuracy with a very low time and hardware budget.

On the one hand, Eve3D has the potential to accelerate progress across several high-level applicative domains, including augmented/virtual reality, robotics, autonomous navigation/interaction with the environment, and 3D content creation. The accuracy-speed trade-off achieved by Eve3D could represent a strong opportunity to democratize access to high-quality 3D modeling, by significantly lowering the entry barriers for researchers, educators, or any independent developers. Furthermore, a faster convergence speed also translates into a reduced carbon footprint associated to 3D reconstruction.

On the other hand, the possibility of producing higher-quality 3D models also comes with ethical considerations. These latter could be misused for applications such as surveillance or other privacy-infringement purposes. However, we argue Eve3D is not designed to handle dynamic objects/subjects during the reconstruction process, thus making it unsuited for processing casually collected videos where subjects may appear without their explicit consent.

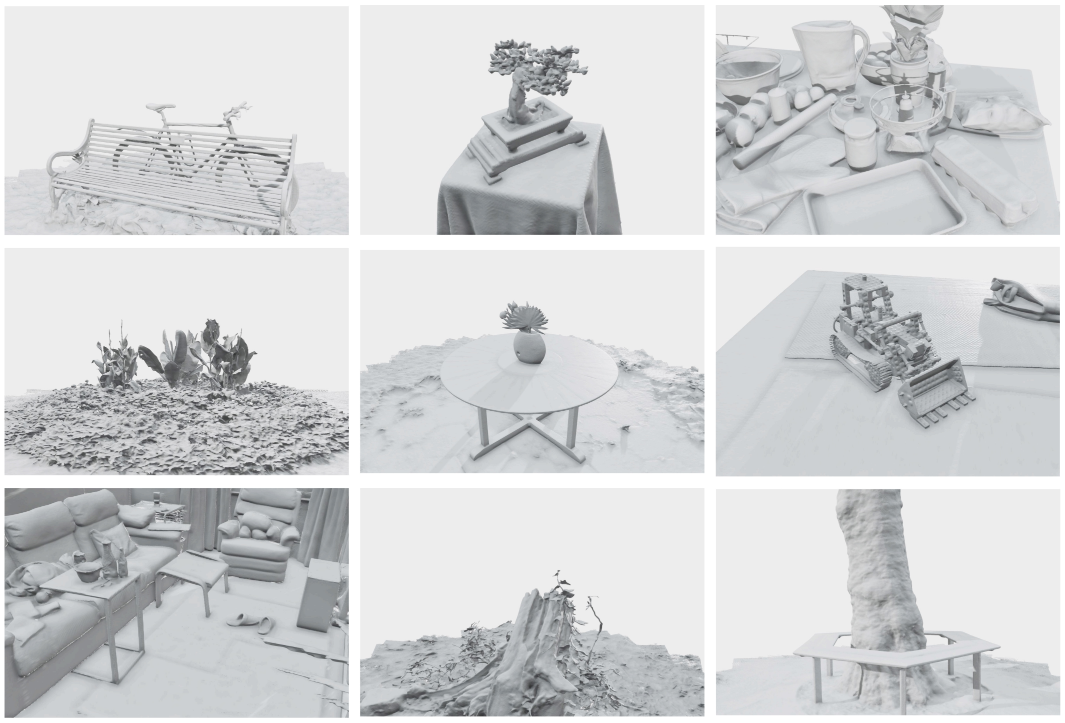
Figure 11: **Qualitative Visualizations on the Mip-NeRF360 Dataset.**