Plenodium: UnderWater 3D Scene Reconstruction with Plenoptic Medium Representation - Supplementary Material -

We thank the reviewers for taking the time to review our supplementary material. We first present the implementation details of our method in Sec. A. We then detail the construction of our simulated dataset and analyze the effect of degradation on COLMAP initialization in Sec. B. In Sec. C, we conduct further ablation studies and provide an in-depth analysis of our findings. Section D provides additional qualitative visualizations to better illustrate the performance of our method. Finally, we consider our work's broader implications and potential societal impact in Sec. E.

A Implementation Detail

This section outlines our implementation details, including the training settings (i.e., Sec. A.1), supplementary information of the pseudo-depth Gaussian complementation (i.e., Sec. A.2), and the depth gradient computation under scattering media (i.e., Sec. A.3).

A.1 Training Settings

We train our model using a volumetric extension of 3D Gaussian Splatting. For reconstruction tasks, we train for 15,000 steps, while for restoration tasks, which require higher accuracy, we extend training to 30,000 steps. Following the progressive training strategy introduced in 3DGS [1], training begins at $1/4$ resolution and gradually doubles every 3,000 steps to increase spatial detail. To prevent unstable updates in the early training phase, we apply a 500-step warm-up before the Gaussian refinement. After warm-up, Gaussian refinement (including densification and culling) is performed every 100 steps. Densification is triggered for a Gaussian primitive when its gradient norm exceeds 0.0008. In this case, if the Gaussian scale is below 0.001, it is copied to expand coverage; otherwise, it is split into two samples to preserve fine-grained structure. In parallel, culling is applied at each refinement step to remove Gaussians with opacity below 0.5. To prevent opacity saturation and encourage stable convergence, all opacities are reset to 0.5 every five refinement steps. Together, these refinement steps first densify to improve coverage, then cull to remove floaters, ensuring a compact and effective representation.

We employ the Depth Anything Model [2, 3] as an external image depth estimator to generate the pseudo-depth maps. We use the newest version, V2, and the largest model variant, ViT-L, which is pretrained on diverse datasets and applied in inference mode without further fine-tuning. Following the official implementation, each image is first resized to a fixed resolution (518×518) before passing through the model. This resizing ensures compatibility with the model's ViT backbone, which performs best under fixed input sizes due to its patch-based architecture. The predicted depth

Table 7: Optimizer and scheduler configurations for each parameter group.

Parameter Group	Initial LR	Final LR	Notes
Means	1.6e-4	5e-5	Position updates
DC Features	2.5e-3	2.5e-4	Direct color channels
Rest Features	1.25e-4	1.25e-5	Non-DC channels
Opacities	5.0e-2	5.0e-2	No decay
Scales	5.0e-3	5.0e-3	No decay
Quaternions	1.0e-3	1.0e-3	Rotation parameters
Medium DC Features	2.5e-3	2.5e-4	For volumetric medium
Medium Rest Features	1.25e-4	1.25e-5	For anisotropic scattering

Algorithm 1 Pseudo-Depth Gaussian Complementation

Input:
The set of the input cameras $\mathcal{V}$, and corresponded images $\mathcal{C}$;
The COLMAP initialized Gaussian primitives, $\mathbf{G}$;
Output:
The final Gaussian primitives, $\mathbf{G}'$;

```
G' = ∅
for V ∈ 𝒱, C ∈ 𝒞 do
    D̂, T^{obj}_{N+1} ← render from G for V using Eqn. 8 of the main manuscript.
    Ω_p ← {(x, y) | T^{obj}_{N+1}(x, y) ≥ τ_w}
    get D̃ by Depth Anything Model with image input C
    Ω_n ← {(x, y) | D̃(x, y) < τ_n ear max(D̃)}
    get D̃' from D̃ using Eqn. 9 of the main manuscript.
    for (x, y) ∈ Ω_n ∪ Ω_p do
        get μ, A using Eqn. 14
        get Σ using Eqn. 15
        σ ← 0.1
        𝒢 ← {μ, Σ, A, σ}
        G' ← G' ∪ {𝒢}
    end for
end for
G' ← G ∪ G'
return G'
```

map is then upsampled via bilinear interpolation to match the original image resolution and stored as a dense pseudo-depth prior for further use in our pipeline.

Each parameter group is optimized using the Adam optimizer with $\epsilon = 10^{-15}$ and exponential decay scheduling. For instance, the 3D means are trained with an initial learning rate of 1.6×10^{-4}, which decays to 5×10^{-5} over time, while opacities are optimized using a fixed learning rate of 0.05. Additional learning rates and scheduler configurations details are provided in Tab. 7.

A.2 More Detail of the Pseudo-Depth Gaussian Complementation

In this section, we detail the procedure of our Pseudo-Depth Gaussian Complementation (PDGC), as summarized in Alg. 1.

Based on the pixel regions selected by Ω_n and Ω_p (as defined in Sec. 4.2 of the main manuscript), we determine where new Gaussians should be inserted. For each selected pixel (x, y), we project it into 3D space as a Gaussian using its calibrated pseudo-depth $\tilde{D}'(x, y)$. The 3D mean position μ and spherical harmonics–encoded color feature A are computed as:

$$\mu = W^T \cdot \begin{bmatrix} \tilde{D}'(x,y) \cdot x \\ \tilde{D}'(x,y) \cdot y \\ \tilde{D}'(x,y) \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}, \quad A = \text{RGB2SH}(C(x,y)), \quad \text{where } (x,y) \in \Omega_n \cap \Omega_p, \tag{14}$$

here, W is the intrinsic matrix, and $[x_c, y_c, z_c]^T$ is the camera position. The function RGB2SH maps RGB values to 0th-order spherical harmonics coefficients for a compact color representation.

To represent the shape and orientation of each Gaussian, we define its covariance matrix Σ via isotropic scaling S and a random rotation R:

$$\Sigma = RSS^TR^T, \quad S = \text{diag}(s, s, s), \quad s = \frac{\tilde{D}'(x,y) \cdot (f_x + f_y)}{h + w}, \tag{15}$$

where diag(s, s, s) constructs a diagonal matrix S that uniformly scales the Gaussian along all three spatial axes, resulting in an isotropic shape. The scalar s adapts the Gaussian size to the scene depth, while considering focal lengths (f_x, f_y) and image dimensions (h, w). The rotation matrix R is randomly initialized to promote diversity in orientation and mitigate optimization bias.

A.3 Backward Pass

Unlike standard 3DGS, where the depth z_i mainly affects the rendered depth $\hat{D}$, in our medium-aware formulation, z_i also influences the final rendered color $\hat{C}$ through scattering and attenuation. The loss $\mathcal{L}$ gradient concerning z_i becomes:

$$\frac{\partial \mathcal{L}}{\partial z_i} = \frac{\partial \mathcal{L}}{\partial \hat{D}} \cdot \frac{\partial \hat{D}}{\partial z_i} + \frac{\partial \mathcal{L}}{\partial \hat{C}} \cdot \frac{\partial \hat{C}}{\partial z_i}. \tag{16}$$

The first term corresponds to the direct contribution of z_i to the depth rendering, which follows the standard 3DGS formulation:

$$\frac{\partial \hat{D}}{\partial z_i} = \alpha_i T_i^{\text{obj}}. \tag{17}$$

The second term accounts for the influence of z_i on color rendering, which stems from the medium-aware compositing process:

$$\hat{C} = \sum_i^N c_i \alpha_i T_i^{\text{obj}} e^{-\sigma^{\text{att}} z_i} + \sum_i^N c^{\text{med}} T_i^{\text{obj}} \left(e^{-\sigma^{\text{bs}} z_{i-1}} - e^{-\sigma^{\text{bs}} z_i} \right) + c^{\text{med}} T_{N+1}^{\text{obj}} e^{-\sigma^{\text{bs}} z_N}. \tag{18}$$

Thus, z_i appears in the following terms of the color computation:

- $\hat{C}_i^{\text{obj}} = c_i \alpha_i T_i^{\text{obj}} e^{-\sigma^{\text{att}} z_i}$, where z_i affects attenuation of the object.
- $\hat{C}_i^{\text{med}} = c^{\text{med}} T_i^{\text{obj}} (e^{-\sigma^{\text{bs}} z_{i-1}} - e^{-\sigma^{\text{bs}} z_i})$, where z_i appears in the second exponential term.
- $\hat{C}_{i+1}^{\text{med}} = c^{\text{med}} T_{i+1}^{\text{obj}} (e^{-\sigma^{\text{bs}} z_i} - e^{-\sigma^{\text{bs}} z_{i+1}})$, where z_i appears in the first exponential term.

Combining these, we get:

$$\frac{\partial \hat{C}}{\partial z_i} = -\sigma^{\text{att}} c_i \alpha_i T_i^{\text{obj}} e^{-\sigma^{\text{att}} z_i} + \sigma^{\text{bs}} c^{\text{med}} e^{-\sigma^{\text{bs}} z_i} \left(T_i^{\text{obj}} - T_{i+1}^{\text{obj}} \right), \tag{19}$$

where, the difference in transmittance simplifies as $T_i^{\text{obj}} - T_{i+1}^{\text{obj}} = T_i^{\text{obj}} - (1 - \alpha_i) T_i^{\text{obj}} = \alpha_i T_i^{\text{obj}}$.

Then, we substitute it back into the gradient of the loss:

$$\frac{\partial \mathcal{L}}{\partial z_i} = \frac{\partial \mathcal{L}}{\partial \hat{D}} \cdot \alpha_i T_i^{obj} + \frac{\partial \mathcal{L}}{\partial \hat{C}} (\sigma^{bs} e^{-\sigma^{bs} z_i} c^{med} - \sigma^{att} e^{-\sigma^{att} z_i} c^{\text{obj}}) \cdot \alpha_i T_i^{obj}, \tag{20}$$

This formulation captures depth's dual role in geometry and appearance, enabling more informative gradient flow in scattering environments.

B More Details of Our Simulated Dataset

In this section, we present additional details about our simulated dataset. We first describe the dataset construction process (i.e., Sec. B.1), including medium configurations and rendering settings. We then analyze the impact of different degradation levels on COLMAP-based initialization (i.e., Sec. B.2).

B.1 Dataset Construction

As shown in Fig. 8, we simulate scattering medium using Blender' `Principled Volume` shader, rendered with the `Cycles` engine to achieve high-fidelity light transport. A vertical density gradient is introduced along the Z-axis by combining the `Texture Coordinate`, `Mapping`, and `Separate XYZ` nodes, followed by a `ColorRamp` node to control the falloff. For fog, we adopt a white absorption color and low anisotropy (0.001) to simulate uniform scattering. For water, we use a bluish absorption tint and increased anisotropy to better approximate underwater light propagation with enhanced forward scattering. Three degradation levels (easy, medium, and hard) are realized by scaling the base density using adjustable `Multipliers` (e.g., 0.005, 0.01, 0.02). All images in our dataset are rendered with linear color management to allow for accurate exposure adjustments during post-processing.

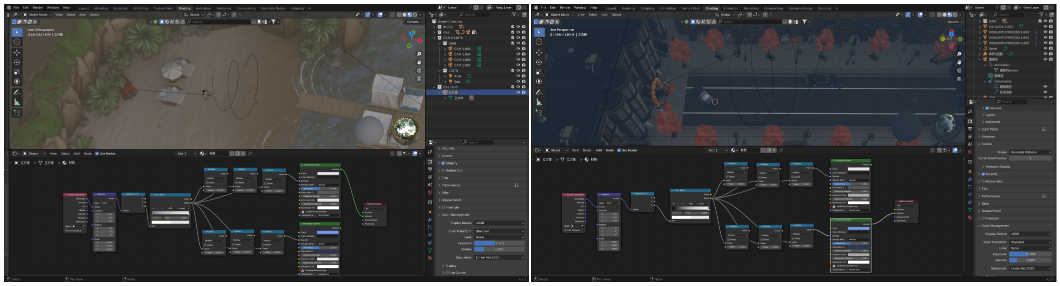


Figure 8: Blender interface used for dataset rendering.

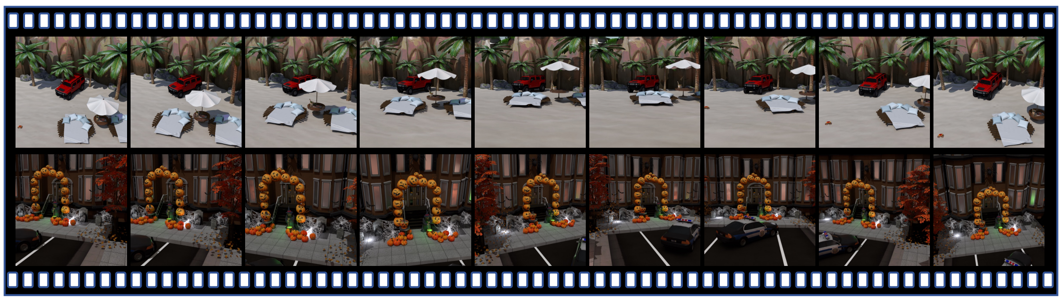


Figure 9: Sampled images from our simulated dataset.

Specifically, we set the view transform to Standard and turn off gamma correction (gamma = 1.0). We do not use user-defined curve adjustments, ensuring no tone mapping or nonlinear operations alter the image. This enables consistent and physically meaningful exposure control during post-processing. The dataset comprises two distinct scenes (Beach and Street), as illustrated in the ground truth (GT) visualizations shown in Fig. 9, supporting robust and comprehensive benchmarking. Additional details, including exact shader setups and scene configurations, are provided in the supplementary Blender source files.

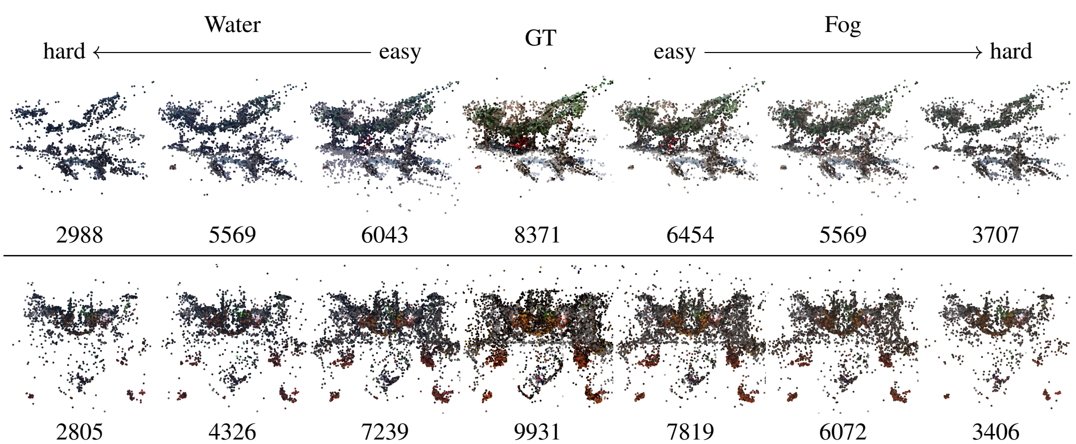


Figure 10: Sparse point clouds obtained by COLMAP under varying degradation conditions. The numbers below each image indicate the number of 3D points. For both the Beach (top) and Street (bottom) scenes, we show the impact of different levels of fog and water degradation (from easy to hard) compared to the clean ground truth. Severe degradation results in significantly sparser points, illustrating the challenge of reliable initialization of 3DGS.

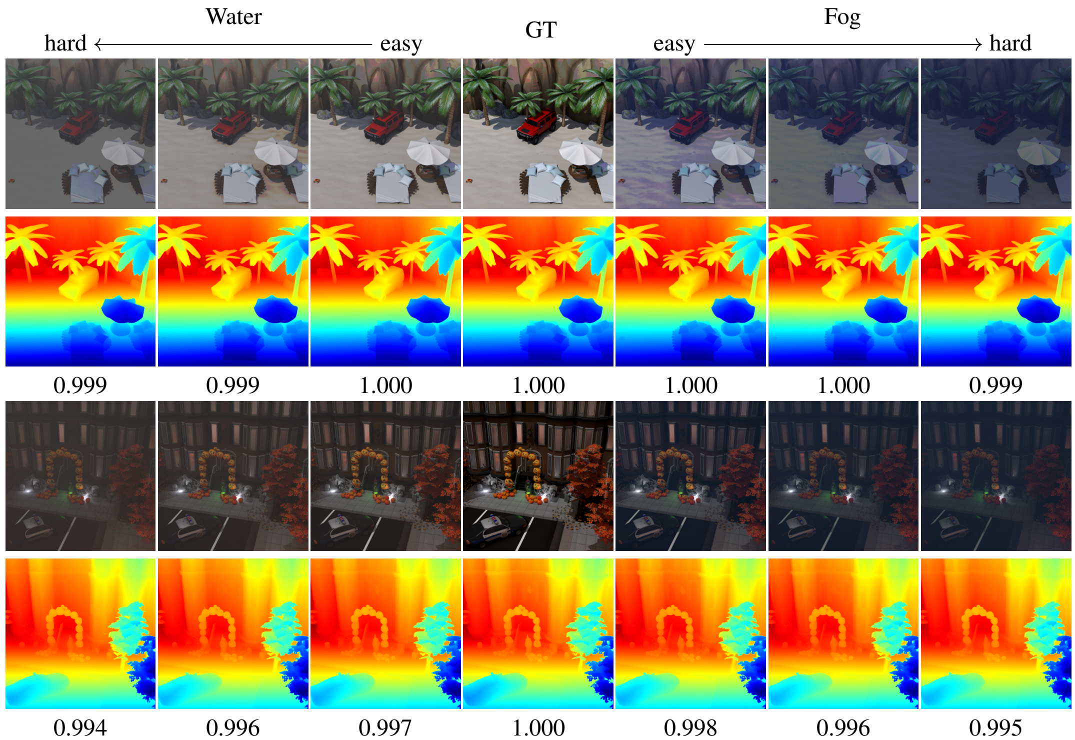


Figure 11: Pseudo-Depth estimated from various degraded images by [3]. Numbers below each map denote the Pearson correlation coefficient concerning the GT-based pseudo-depth. The consistently high values (close to 1.0) validate its effectiveness as a reliable depth in different environments.

B.2 Dataset Analysis

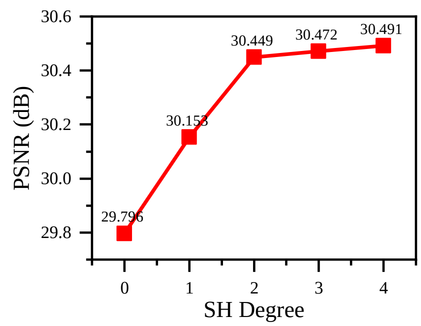
To evaluate the impact of image degradation caused by scattering media on the structure-from-motion (SfM) [4] initialization process in COLMAP [4, 5], we analyze the density and completeness of the generated sparse point clouds under degraded imaging conditions. When image quality is compromised due to fog or water, COLMAP struggles with reliable feature extraction and matching, resulting in significantly sparser and less accurate point clouds. As visualized in Fig. 10, specific regions, particularly those with strong scattering effects, exhibit apparent gaps or absences in the geometry. This degradation-induced sparsity directly hinders the quality of subsequent reconstruction stages, especially for methods relying on accurate geometry priors, such as 3DGS. These findings highlight the sensitivity of COLMAP-based initialization pipelines to visibility degradation, underscoring the need for complementary initialization strategies to recover missing geometry in severely degraded scenes.

C More Analysis and Discussion

In this section, we provide a comprehensive analysis of our method under various settings. We first verify the robustness of our pseudo-depth under diverse degradation types (i.e., Sec. C.1) and examine how critical hyperparameters affect performance(i.e., Sec. C.2). We then assess the impact of COLMAP initialization (i.e., Sec. C.3) and our depth ranking regularized loss (i.e., Sec. C.4), demonstrating their importance for stable geometry learning. Furthermore, we analyze statistical variance across different runs and degradation levels to establish result consistency (i.e., Sec. C.5). Finally, we analyze the limitations of our method with respect to the LPIPS metric (i.e., Sec. C.6).

C.1 Robustness of Pseudo-Depth

Our pipeline leverages the Depth Anything Model [2, 3], a state-of-the-art monocular depth estimator, to compute robust pseudo-depth maps from media-degraded images. These maps serve as essential



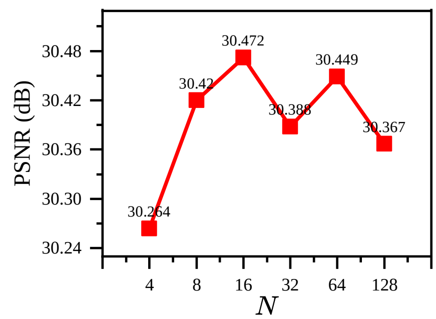


Figure 12: Effect of varying the maximum SH degree used for the plenoptic medium representation.

Figure 13: Effect of varying the number N of patches used in the depth ranking regularized loss.

Table 8: Effect of the COLMAP initialization. We compare it with a random initialization with 50,000 points.

Initialzation	PSNR	SSIM	LPIPS	FPS	Time
Random	25.198	0.7983	0.2235	116	6.4min
COLMAP	30.388	0.9207	0.1274	237	7.0min
COLMAP & PDGC	30.472	0.9225	0.1276	249	7.0min

Table 9: Effect of the depth ranking regularized loss. We compare it with the Pearson correlation loss from [6].

Loss	PSNR	SSIM	LPIPS	FPS	Time
w/o $\mathcal{L}_{\text{depth}}$	30.305	0.9212	0.1272	252	7.0min
w/ $\mathcal{L}'_{\text{depth}}$ [6]	30.384	0.9209	0.1292	246	7.7min
w/ $\mathcal{L}_{\text{depth}}$	30.472	0.9225	0.1276	249	7.0min

guidance for both our Pseudo-Depth Gaussian Complementation (PDCG) and the depth ranking regularized loss. As shown in Fig. 11, a key advantage of this approach is its robustness to medium-induced degradations. Despite varying levels of scattering and absorption in both water and fog, the pseudo-depth maps remain visually consistent across different input conditions and align well with those derived from clean ground-truth images. To quantitatively support this observation, we report the Pearson [6] correlation coefficient below each depth map, comparing each pseudo-depth to the one predicted from the clean (GT) image. The consistently high correlation values (e.g., >0.99) validate the robustness and medium-agnostic nature of the predictions by [3], making it well-suited for initialization and supervision in degraded scenes.

C.2 Effect of Hyperparameters

In our experiments, we investigate two critical hyperparameters that affect the performance of our plenoptic medium representation and the efficacy of the depth ranking regularized loss.

First, in Fig. 12, we control the maximum spherical harmonics (SH) degree for our plenoptic representation in our method. Adjusting this parameter determines the level of angular complexity captured in the medium field, thereby influencing the fidelity of volumetric effects such as scattering and color absorption. A higher maximum SH degree can model more detailed angular variations. Still, it may also increase computational cost and risk of overfitting, whereas a lower degree results in a smoother but potentially oversimplified medium representation. To achieve an optimal trade-off between computational efficiency and representational fidelity, we fix the SH degree to 3.

Second, we vary the number N of downsampled patches used in the depth ranking regularized loss for our method in Fig. 13. This loss plays a crucial role in enforcing depth consistency during training. A larger N provides finer granularity for capturing local depth variations, but it also introduces more noise and increases computational overhead, even out-of-memory issues during training. In contrast, a smaller N simplifies the loss calculation but may not capture sufficient spatial detail. Empirically, setting $N = 16$ yields the best performance while maintaining a reasonable computational load.

C.3 Effect of the COLMAP Initialization

To evaluate the role of COLMAP-based initialization within our framework, we compare three variants: (1) our method with COLMAP initialization but without PDGC, (2) our method with random initialization using 50,000 uniformly sampled 3D points, and (3) our full pipeline combining

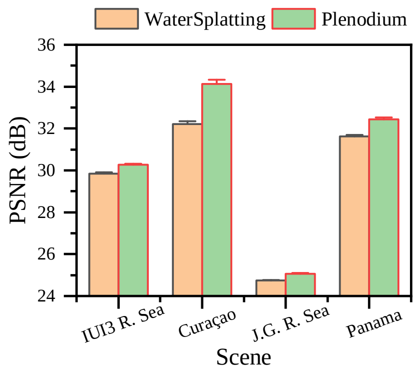


Figure 14: Mean and variance of reconstruction quality over four runs on real-world scenes.

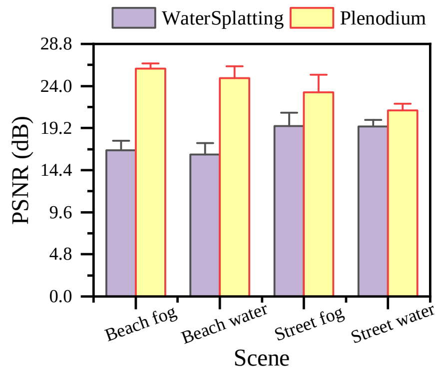


Figure 15: Performance variation on simulated data across different degradation levels.

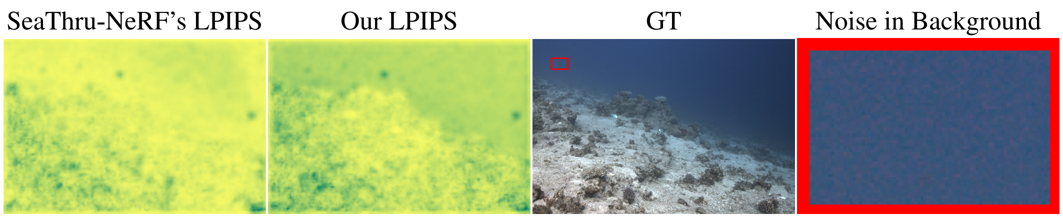


Figure 16: Visual comparison of LPIPS maps between SeaThru-NeRF and our Plenodium. The primary difference appears in background regions corrupted by GT noise.

COLMAP with PDGC. COLMAP provides a strong geometric prior that aids reconstruction; however, under severe degradation (e.g., fog or water), its output often becomes sparse and contains missing regions. In contrast, random initialization does not rely on scene-specific priors but ensures uniform spatial coverage, even in areas where COLMAP fails to generate points. As shown in Tab. 8, despite the degraded visibility, COLMAP initialization still leads to better performance than random initialization, validating the utility of its geometric prior. Moreover, our full method—augmenting COLMAP with PDGC—further improves results, indicating that while COLMAP provides a solid foundation, suggesting that complementary strategies can effectively enhance geometric priors under a degraded environment.

C.4 Effect of the Depth Ranking Regularized Loss

To further evaluate the effectiveness of our proposed depth ranking regularized loss $\mathcal{L}_{\text{depth}}$, we compare our method (i.e., w/ $\mathcal{L}_{\text{detph}}$) against two baselines: one trained without any depth supervision (i.e., w/o $\mathcal{L}_{\text{depth}}$), and another using the Pearson correlation-based depth loss $\mathcal{L}'_{\text{depth}}$ adopted in FSGS [6] (i.e., w/ $\mathcal{L}'_{\text{depth}}$). As shown in Tab. 9, while $\mathcal{L}'_{\text{depth}}$ provides marginal improvements over the no-depth baseline, our method that leverages $\mathcal{L}_{\text{depth}}$ achieves superior performance, which shows that our depth ranking regularized loss offers more effective geometric supervision with imprecise pseudo-depth supervision.

C.5 Statistical Analysis

To ensure the robustness and stability of our quantitative results, we conduct four independent training runs on real-world scenes and report the average performance in Tab. 1 of the main manuscript. As shown in Fig. 14, we visualize the mean performance across runs and the corresponding variance to reflect consistency.

For our simulated dataset, we compare the average PSNR across scenes under water and fog degradation. As shown in Fig. 15, our method (Plenodium) consistently outperforms WaterSplatting across all conditions. The error bars represent the standard deviation across different degradation

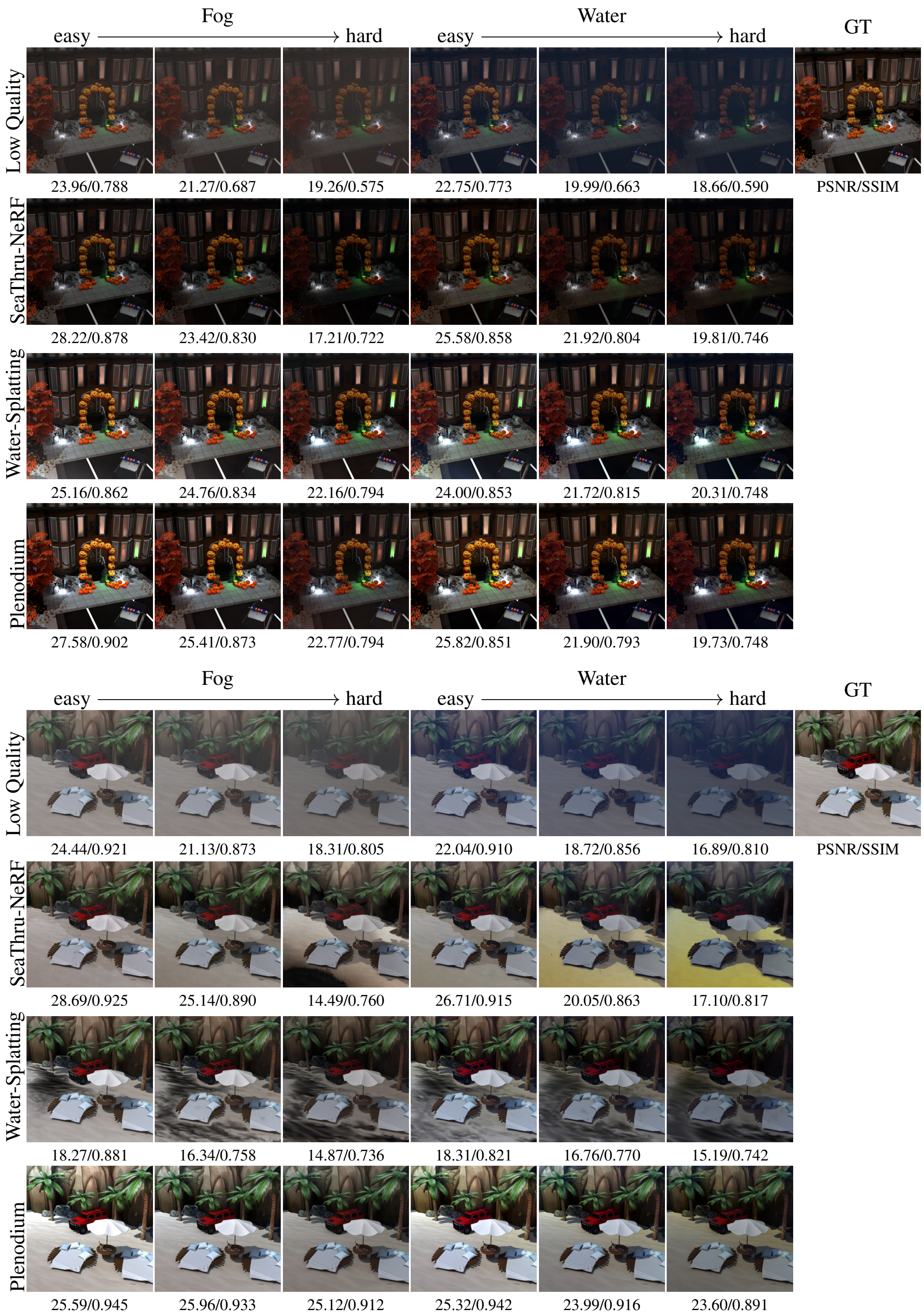


Figure 17: Visual comparison on our simulated dataset.

levels, reflecting both the effectiveness and robustness of each method under challenging visual environments.

C.6 Limitation

While our method underperforms SeaThru-NeRF in terms of LPIPS in some scenes (as reported in Tab. 1 of the main manuscript), we conduct a visual analysis to better understand this discrepancy. As shown in Fig. 16, the LPIPS maps indicate that the main difference arises in the medium regions, where our method yields higher LPIPS values. We further observe that background areas in the GT contain visible noise, which may act as a confounding factor in LPIPS evaluation, limiting its reliability in degraded scenes.

D More Visualizations

In this section, we present additional visualizations on our simulated dataset, comparing SeaThru-NeRF [7], WaterSplatting [8], and our proposed Plenodium, as shown in Fig. 17. We also include video results in the supplementary material, rendered at 24 FPS using camera trajectories interpolated from the evaluation poses with a step size of 10.

E Broader Impact

Our method offers a more accurate and efficient solution for underwater 3D reconstruction, which can positively impact fields such as marine ecology, environmental monitoring, underwater archaeology, and infrastructure inspection. By improving scene recovery in visually degraded environments, our approach may assist in documenting underwater habitats, tracking pollution effects, and preserving submerged cultural heritage. Furthermore, the proposed simulated dataset provides a benchmark for evaluating underwater image restoration methods, promoting reproducibility and transparency. However, as with any enhanced visual sensing technology, there exists potential for misuse in surveillance or unauthorized mapping. We encourage responsible use and recommend that applications of this technology follow appropriate ethical and legal guidelines.

References

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 139–1, 2023.

[2] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 371–10 381.

[3] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," *Conference on Neural Information Processing Systems*, vol. 37, pp. 21 875–21 911, 2025.

[4] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[5] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision*, 2016.

[6] Z. Zhu, Z. Fan, Y. Jiang, and Z. Wang, "Fsgs: Real-time few-shot view synthesis using gaussian splatting," in *European Conference on Computer Vision*. Springer, 2024, pp. 145–163.

[7] D. Levy, A. Peleg, N. Pearl, D. Rosenbaum, D. Akkaynak, S. Korman, and T. Treibitz, "Seathru-nerf: Neural radiance fields in scattering media," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2023, pp. 56–65.

[8] H. Li, W. Song, T. Xu, A. Elsig, and J. Kulhanek, "WaterSplatting: Fast underwater 3D scene reconstruction using gaussian splatting," *International Conference on 3D Vision*, 2025.