Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Optical-Mass Water Gaussians for Per-Scene Underwater Rendering

Jun Zhao
Hangzhou Dianzi University
Jiafei Cao
Shanghai Artificial Intelligence Laboratory
Rongfeng Lu
Lishui University
Longjian Zeng
245060073@hdu.edu.cn
Hangzhou Dianzi University
Yiming Cui
Hangzhou Dianzi University
Zhangyi Yang
Hangzhou Dianzi University
Xichun Sheng
Macao Polytechnic University
Yahong Chen
Lishui University

Research Article

Keywords: Underwater scene reconstruction, Novel-view synthesis, 3D Gaussian splatting, Neural rendering, Participating media, Optical transport

Posted Date: September 30th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10938586/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Optical-Mass Water Gaussians for Per-Scene Underwater Rendering

Jun Zhao[1†], Jiafei Cao[2†], Rongfeng Lu[4,5], Longjian Zeng[1*], Yiming Cui[1], Zhangyi Yang[1], Xichun Sheng[3], Yahong Chen[4,5]

[1]Hangzhou Dianzi University, Hangzhou, 310018, China.
[2]Shanghai Artificial Intelligence Laboratory, Shanghai, 200232, China.
[3]Macao Polytechnic University, Macao, 999078, China.
[4]Lishui University, Lishui, 323000, Zhejiang, China.
[5]Lishui Key Laboratory of Low-Altitude Multimodal Sensing and Intelligent Computing, Lishui University, Lishui, 323000, Zhejiang, China.

*Corresponding author(s). E-mail(s): 245060073@hdu.edu.cn;
†These authors contributed equally to this work.

Abstract

Per-scene underwater reconstruction supports marine archaeology, seabed survey, and offshore inspection, but faces two coupled problems: attenuation and backscatter mix water effects with object appearance, and common 3D Gaussian Splatting renderers encode the medium globally, directionally, or on object surfaces, limiting spatial queries after camera motion. We propose AquaMass-GS (AM-GS), which couples an object-Gaussian scene model to normalized free-space water Gaussians. Cell-volume weighting gives direct-extinction and backscatter optical-mass semantics; analytic finite-path integration with coverage normalization queries water from camera to visible surface. Separate optimization and querying keep object appearance scene-tied while representing water by one equivalent coefficient per finite ray segment. On four SeaThru-NeRF development scenes, AM-GS provides competitive end-to-end novel-view quality, with the highest mean peak signal-to-noise ratio (PSNR) and the lowest learned perceptual image patch similarity (LPIPS) among six methods; the observed margins over WaterSplatting (0.282 dB PSNR and 0.0002 LPIPS) are not statistically decisive under paired uncertainty, and its mean structural similarity (SSIM) is slightly lower. On paired simulation with hidden clean targets it improves medium-disabled SSIM by 0.1139 over WaterSplatting, and, in a controlled known-optics test using a compact $\boldsymbol{\sigma/\Delta = 0.30}$ carrier, it reduces unseen-origin transmittance RMSE by 52.7% relative to a trilinear voxel carrier.

Keywords: Underwater scene reconstruction; Novel-view synthesis; 3D Gaussian splatting; Neural rendering; Participating media; Optical transport

1 Introduction

High-fidelity underwater reconstruction can turn a sparse camera survey into a navigable three-dimensional map of a submerged site. For seabed mineral exploration, offshore construction, and archaeological excavation, such a map supports route planning, equipment placement, and risk assessment before vehicles and tools enter the water. Better virtual inspection can reduce repeated deployments, improve operational accuracy, and lower the cost of work in environments where every dive is expensive. Recent work in visual computing has applied Gaussian scene representations to degraded imagery and scene-aware rendering [1, 2]. We study per-scene underwater novel-view synthesis, where a renderer is optimized from posed images of one submerged site and synthesizes held-out views of that same site [3]. This task has two coupled challenges: first, wavelength-dependent attenuation weakens the object signal, while suspended particles add accumulated backscatter [4]; these water effects are entangled with geometry, reflectance, lighting, and camera response. Second, camera motion changes the three-dimensional water path to each visible surface, so the medium must be spatially queryable independently of the reconstructed object.

The emergence of 3D Gaussian Splatting (3DGS) [5] has brought high-quality novel-view synthesis (NVS) to underwater scenes. Recent systems combine surface Gaussians with physically motivated attenuation and backscatter [6–10]. Their medium is typically a scene-global depth law, a direction-conditioned neural prediction, an optical variable stored on reconstructed surfaces, or a positionally interpolated directional feature grid. Such carriers can explain captured appearance efficiently, but they do not combine a water representation independent of object sampling with an explicit query over the finite camera-to-visible-surface path. Consequently, the second challenge remains unresolved: the renderer lacks a free-space transport carrier whose optical contribution can be evaluated when the camera origin and traversed path change. We refer to this as the missing free-space transport query.

To bridge this gap and address both challenges, we propose AquaMass-GS (AM-GS), shown in Fig. 1. At the representation level, the method couples an object-Gaussian scene model with a distinct field of normalized free-space water Gaussians whose cell-volume-weighted parameters encode optical mass for direct extinction and backscatter. At the rendering level, the object branch supplies a detached alpha-weighted surface endpoint; analytic finite-path integration and coverage normalization then query the water field from the camera to that visible surface and convert its optical depth into one per-pixel equivalent transport coefficient for the finite segment. Separating the water carrier from object sampling reduces the representational entanglement between object sampling and water transport, while evaluating only the traversed finite segment handles changes in camera origin and path. In contrast to global or direction-conditioned predictors [6, 8] and surface-attached optical variables [10], AM-GS keeps water transport spatially queryable in free space.

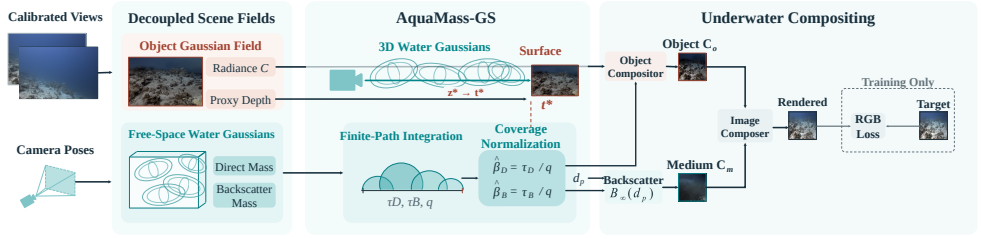


Fig. 1 Overview of AquaMass-GS. Calibrated underwater observations jointly optimize an object-Gaussian field and separate free-space water Gaussians carrying direct-extinction and backscatter optical mass. The detached axial proxy depth z^* is converted to ray distance t^*, and analytic integration over $[0, t^*]$ produces optical depths $\boldsymbol{\tau}_D, \boldsymbol{\tau}_B$ and unit-density coverage q. Coverage normalization yields the per-pixel equivalent coefficients $\widehat{\boldsymbol{\beta}}_D = \boldsymbol{\tau}_D/q$ and $\widehat{\boldsymbol{\beta}}_B = \boldsymbol{\tau}_B/q$. The retained depth-aware object compositor uses $\widehat{\boldsymbol{\beta}}_D$, while $\widehat{\boldsymbol{\beta}}_B$ and the direction-conditioned $\mathbf{B}^{\infty}(\mathbf{d}_p)$ head form the medium term. The image composer combines the object and medium terms into the rendered underwater view; the captured target participates only in the training loss. Image tiles are real inputs or renderer outputs, whereas the water lattice, ray geometry, and integration profiles are schematic.

The contributions are threefold:

- We formulate a normalized free-space water-Gaussian field whose cell-volume-weighted direct-extinction and backscatter parameters have per-primitive optical-mass semantics independent of object sampling and lattice resolution.
- We introduce a detached finite-surface query that combines analytic Gaussian integration with partition-of-unity coverage normalization, recovering the exact equivalent coefficient for a constant underlying field on every positive-coverage ray while retaining an existing object renderer.
- We provide matched end-to-end and controlled representation evidence: on four SeaThru-NeRF development scenes, AM-GS delivers competitive quality, ranking first in mean PSNR and LPIPS among six methods while WaterSplatting is marginally higher in mean SSIM; on paired simulation, it improves hidden-clean SSIM by 0.1139 over WaterSplatting; and in a controlled known-optics test with a compact $\sigma/\Delta = 0.30$ carrier, it reduces unseen-origin transmittance RMSE by 52.7% relative to a trilinear voxel carrier.

2 Related Work

2.1 Underwater Formation and Restoration

Physics-based restoration methods estimate how wavelength-dependent attenuation and backscatter alter a captured image. The revised underwater image-formation model separates the coefficients governing direct signal and backscatter [4]; Sea-thru combines range with dark-pixel evidence for water removal [11]. Haze-Lines extends color restoration across different water types and introduces a natural-image quantitative dataset [12]. Recent learned restoration models use stronger priors and explicit degradation

decomposition; for instance, SLURPP combines latent diffusion with transmission and backscatter prediction [13]. Related visual-computing work also treats adverse illumination as a restoration problem [14]. These methods restore one image or video frame at a time and do not maintain a view-consistent 3D water volume. In contrast, AM-GS optimizes one 3D scene from posed multi-view images, renders novel underwater views, and exposes a medium-disabled object diagnostic.

2.2 Underwater Neural Rendering

SeaThru-NeRF transfers underwater decomposition to neural radiance fields [3]. Underwater Gaussian methods improve rendering speed but encode water in different places. SeaSplat and Gaussian Splashing apply learned, depth-conditioned attenuation and backscatter [6, 7], while WaterSplatting and AquaSplatting pair explicit scene Gaussians with auxiliary neural medium fields [8, 15]. 3D-UIR combines pose-conditioned appearance embeddings, optical parameters, and pseudo-depth supervision [9]. Plenodium introduces a positional medium grid whose spherical-harmonic features depend on viewing direction [16]; WaterClear-GS stores local spectral optical parameters on object Gaussians [10]; and UW-3DGS predicts spatially varying attenuation and backscatter from a factorized voxel volume [17]. UW-GS and OceanSplat focus on distractor suppression and medium-aware geometry [18, 19]. Recent visual-computing studies broaden this line of work to degraded-image quality assessment, scene-aware rendering, and joint depth–image reconstruction [1, 2, 20]. DualPhys-GS evaluates feature-guided attenuation and scattering on both SeaThru-NeRF and SaltPond [21]; Li and Fu's Gaussian splat-field framework targets few-shot underwater synthesis [22]; and HydroGaussian adds hierarchical depth regularization and explicit ray–Gaussian intersections [23]. Unlike these geometry-, quality-, and semantics-oriented extensions, AM-GS isolates the medium-carrier question: it retains the object renderer and requires neither monocular-depth nor semantic priors, while assigning optical mass to normalized free-space water Gaussians and querying them only along the finite camera-to-surface path.

2.3 Gaussian Scene Representations

3DGS represents radiance with anisotropic surface splats and a depth-sorted tile rasterizer [5]. Mip-Splatting improves scale consistency [24], while 2D Gaussian Splatting emphasizes geometrically accurate surface elements [25]. These methods target opaque-scene appearance rather than participating media. Vol3DGS analytically integrates 3D Gaussian density to obtain volumetrically consistent alpha values [26]. Building on this volumetric principle, AM-GS applies analytic integration specifically to normalized water primitives: object splatting remains unchanged, water mass is assigned by a finite-volume rule, and finite-path optical depth is mapped to the retained underwater compositor.

3 Method

3.1 Overview

AquaMass-GS consists of a conventional object-Gaussian set $\mathcal{G}_o$ and a separate free-space water set $\mathcal{G}_w$. To isolate the proposed water representation, the object branch, direction encoding, training losses, and underwater compositor follow the released WaterSplatting implementation [8]. Our change has four stages: a normalized water field, finite integration to a detached object-surface proxy, unit-density coverage normalization, and equivalent-coefficient composition. Figure 1 states this contract without depicting water primitives as depth-sorted object events.

3.2 Hybrid Scene Representation

For every scene, both Gaussian sets are optimized from the same posed underwater images. Object Gaussian j retains the standard mean, anisotropic covariance, opacity α_j, and spherical-harmonic radiance $\mathbf{c}_j$ of 3DGS [5]. The retained object branch uses the reference parameterization and densification schedule; our changes begin only when its splatted surface statistics are queried by the water field.

The water centers are fixed on a face-centered-cubic (FCC) lattice inside the axis-aligned bounds of the structure-from-motion points plus a fixed world-space margin. For N cells per axis, four offsets per cell produce $4N^3$ centers. The axis-aligned covariance is also fixed, with standard deviation $s\Delta$ along each dimension. The compact reported configuration uses $s = 1.3$; the initial $s = 0.7$ layout is retained as an equal-capacity control in Table 5. Only two nonnegative RGB density vectors per center are optimized: one for direct attenuation and one for backscatter attenuation. Keeping position and covariance fixed prevents transport mass from changing through geometric resizing.

3.3 Normalized Free-Space Water Field

Water Gaussian i defines a normalized basis

$$\phi_i(\mathbf{x}) = \frac{\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right]}{(2\pi)^{3/2}|\boldsymbol{\Sigma}_i|^{1/2}}. \tag{1}$$

$$\int_{\mathbb{R}^3} \phi_i(\mathbf{x})\, d\mathbf{x} = 1. \tag{2}$$

For cell volume $\Delta V = \Delta_x \Delta_y \Delta_z$, every lattice point is assigned $V_i = \Delta V/4$. Water point i receives RGB direct-extinction and backscatter-attenuation optical mass

$$\mathbf{m}_{D,i} = V_i \,\mathrm{softplus}(\boldsymbol{\rho}_{D,i}), \qquad \mathbf{m}_{B,i} = V_i \,\mathrm{softplus}(\boldsymbol{\rho}_{B,i}). \tag{3}$$

The continuous optical fields are

$$\boldsymbol{\sigma}_D(\mathbf{x}) = \sum_i \mathbf{m}_{D,i}\phi_i(\mathbf{x}), \quad \boldsymbol{\sigma}_B(\mathbf{x}) = \sum_i \mathbf{m}_{B,i}\phi_i(\mathbf{x}). \tag{4}$$

Unlike peak opacity, these parameters have an integral meaning over $\mathbb{R}^3$ and scale with the assigned volume under lattice refinement. The Gaussian bases are not truncated at the declared bounds; Section 4.9 therefore distinguishes assigned full-space mass from mass inside the hard box.

3.4 Detached Finite-Surface Proxy

The water field needs a finite endpoint because the target medium contribution depends on how far the visible scene lies from the camera. We first render object alpha and alpha-weighted camera-space depth without water. For pixel p, let $w_j(p) = T_j(p)\alpha_j(p)$, where T_j is the standard object visibility before Gaussian j. The proxy depth is

$$z^*(p) = \mathrm{stopgrad}\left(\frac{\sum_j w_j(p) z_j}{\max(\sum_j w_j(p), \epsilon)}\right). \tag{5}$$

Pixels with negligible accumulated alpha use the farthest visible object depth in that camera. Stopping gradients prevents the medium integral from moving object geometry solely to alter its integration interval.

Let the unit world-space ray be $\mathbf{r}_p(t) = \mathbf{o} + t\mathbf{d}_p$, and let $d_{p,z}^{\mathrm{cam}}$ denote the corresponding unit ray's camera-forward component. Because z^* is camera-space axial depth, the integration endpoint in ray-distance units is $t^* = z^*/\max(d_{p,z}^{\mathrm{cam}}, \epsilon)$; this conversion is performed inside the analytic CUDA rasterizer. For water Gaussian i, define

$$L_i(p, z^*) = \int_0^{t^*} \phi_i(\mathbf{r}_p(t))\, dt. \tag{6}$$

Let $\mathbf{A}_i = \boldsymbol{\Sigma}_i^{-1}$, $\mathbf{q}_i = \mathbf{o} - \boldsymbol{\mu}_i$,

$$a = \mathbf{d}_p^\top \mathbf{A}_i \mathbf{d}_p, \quad b = \mathbf{d}_p^\top \mathbf{A}_i \mathbf{q}_i, \quad c = \mathbf{q}_i^\top \mathbf{A}_i \mathbf{q}_i. \tag{7}$$

Completing the square gives a closed form:

$$\begin{aligned} L_i(p) = {} & \frac{\exp[-\frac{1}{2}(c - b^2/a)]}{(2\pi)^{3/2}|\boldsymbol{\Sigma}_i|^{1/2}} \sqrt{\frac{\pi}{2a}} \\ & \times \left[\mathrm{erf}\left(\frac{at + b}{\sqrt{2a}}\right)\right]_0^{t^*}. \end{aligned} \tag{8}$$

The finite-path RGB optical depths are

$$\boldsymbol{\tau}_k(p, z^*) = \sum_i \mathbf{m}_{k,i} L_i(p, z^*), \qquad k \in \{D, B\}. \tag{9}$$

3.5 Coverage-Normalized Composition

Let $V_0 = \Delta V/4$ be the common volume assigned to each FCC center. The unit-density coverage length of the same finite support is

$$q(p, z^*) = V_0 \sum_i L_i(p, z^*). \tag{10}$$

At a finite box boundary or under numerical support culling, q need not equal the geometric ray length represented by z^*. Dividing by q preserves the relative contributions of all intersected Gaussians while removing this zeroth-order coverage bias. The production renderer obtains two outputs from the same analytic operator. For compatibility with the rasterizer interface, the learned-mass and unit-density passes return $\boldsymbol{\tau}_k/z^*$ and q/z^*; their ratio gives the finite-segment equivalent coefficient:

$$\widehat{\boldsymbol{\beta}}_k(p) = \frac{\boldsymbol{\tau}_k(p, z^*)}{\max(q(p, z^*), \epsilon)}, \qquad k \in \{D, B\}. \tag{11}$$

This is a per-pixel coverage-normalized homogeneous equivalent of the explicit field over the proxy segment; it is not a global water coefficient. For a constant density $\boldsymbol{\beta}_k$, Eq. 3 gives $\mathbf{m}_{k,i} = V_0\boldsymbol{\beta}_k$, so Eqs. 9–11 return $\widehat{\boldsymbol{\beta}}_k = \boldsymbol{\beta}_k$ exactly for every positive-coverage ray. The query deliberately collapses the depth profile on that segment to one coefficient per channel. Two profiles with the same normalized segment integral therefore produce the same $\widehat{\boldsymbol{\beta}}_k$, which is then reused at every object depth on the ray; this is an explicit carrier query, not depth-resolved medium–object transport. The coefficient is constant only along the queried ray: it still varies across pixels and camera origins because each ray samples a different spatial subset of the free-space field. This cross-ray spatial variation is the purpose of the carrier and is evaluated directly in the controlled unseen-origin experiment. Moreover, $\mathbf{m}_{B,i}$ controls attenuation inside the inherited backscatter formation term, while the direction-conditioned $\mathbf{B}^\infty$ head supplies scattering radiance. It is not itself emitted-radiance mass. An independent production audit evaluates both passes on all eight final native-resolution views and verifies $\widehat{\boldsymbol{\beta}}_k = (\boldsymbol{\tau}_k/z^*)/(q/z^*) = \boldsymbol{\tau}_k/q$ to a maximum absolute discrepancy below 4.6×10^{-7}. This numerical identity audits the coefficient query; geometric-distance composition is audited separately in Section 4.9.

Let object Gaussians be ordered by camera-space axial depth z_j, with $z_0 = 0$, and let $d^{\mathrm{cam}}_{p,z} \in (0, 1]$ be the camera-forward component of pixel p's unit world-space ray. Their geometric camera-ray distances and accumulated object visibility are

$$t_j(p) = \frac{z_j}{\max(d^{\mathrm{cam}}_{p,z}, \epsilon)}, \qquad t_0 = 0, \qquad T_j = \prod_{\ell<j}(1 - \alpha_\ell). \tag{12}$$

The implementation equivalently multiplies each per-pixel coefficient by $1/d^{\mathrm{cam}}_{p,z}$ before invoking the inherited axial-depth compositor. The retained compositor evaluates the

attenuated object term

$$\mathbf{C}_o(p) = \sum_j T_j \alpha_j \, \mathbf{c}_j \odot \exp[-\widehat{\boldsymbol{\beta}}_D(p) t_j], \tag{13}$$

and accumulates backscatter over each object-depth interval,

$$\begin{aligned} \Delta\mathbf{C}_{m,j}(p) &= T_j \mathbf{B}^{\infty}(\mathbf{d}_p) \odot \\ &\qquad \left(e^{-\widehat{\boldsymbol{\beta}}_B t_{j-1}} - e^{-\widehat{\boldsymbol{\beta}}_B t_j} \right), \qquad (14) \\ \mathbf{C}_m(p) &= \sum_j \Delta\mathbf{C}_{m,j}(p) \\ &\qquad + T_{J+1} \mathbf{B}^{\infty}(\mathbf{d}_p) \odot e^{-\widehat{\boldsymbol{\beta}}_B t_J}. \qquad (15) \end{aligned}$$

The final underwater image is $\mathbf{C}_o + \mathbf{C}_m$. The asymptotic color $\mathbf{B}^{\infty}(\mathbf{d}_p)$ remains the inherited direction-conditioned neural nuisance head; consequently, the full medium appearance is not represented exclusively by explicit Gaussians. For one opaque surface at z^*, Eqs. 11–15 reduce to

$$\mathbf{C} = \mathbf{J} \odot e^{-\boldsymbol{\beta}_D t} + \mathbf{B}^{\infty} \odot (1 - e^{-\boldsymbol{\beta}_B t}), \tag{16}$$

the standard direct attenuation and backscatter model [4], using the normalized coefficients from Eq. 11. With multiple semi-transparent surfaces, one proxy endpoint is shared by all object contributions, so the conversion is an approximation whose error is measured in Section 4.9.

3.6 Optimization and Outputs

Training retains the reference loss implementation with the configured `reg_l2` and `reg_ssim` branches. Specifically, the RGB term is the mean squared residual after division by stopgrad($\widehat{\mathbf{I}}$) + 10^{-3}, and the structural term is one minus SSIM [27] after applying the same prediction-dependent normalization to both images. Their convex combination supervises only captured underwater RGB. The RGB and structural weights are 0.8 and 0.2. The water densities receive a spatial total-variation penalty of 10^{-4}, formed from mean absolute first differences along the three lattice axes plus the within-cell deviation of the four FCC sites. No clean images, calibrated optical coefficients, or medium masks are used in the real-scene experiments. The reported compact configuration uses $N = 8$, hence 2,048 fixed water Gaussians and 12,288 learned RGB transport parameters, and optimizes both transport densities jointly with the object set for 10k iterations. Adam uses learning rate 2.5×10^{-3} for each transport-density tensor and 10^{-3} for the retained directional color head, with optimizer $\epsilon = 10^{-15}$; the density rates are constant, while the directional head decays to 1.5×10^{-4} over the configured 15k horizon. The grid spans the COLMAP point bounds plus a two-world-unit margin before multiplication by the archived coordinate scale. Direct and backscatter-attenuation densities initialize respectively to

$(0.035, 0.020, 0.012)/s_c$ and $(0.012, 0.018, 0.025)/s_c$, where s_c is that coordinate scale; $\mathbf{B}^\infty$ initializes from $(0.08, 0.32, 0.48)$. The initially frozen $N = 16$, $s = 0.7$ configuration remains in the component, stochastic, numerical, and efficiency audits so the post-hoc capacity decision is visible. Optimizer decay horizons remain configured for 15k steps. The real-scene WaterSplatting and Ours runs terminate at 10k as stated in Section 4.1; the paired-simulation runs use the separately declared 15k protocol.

The renderer exports the supervised underwater image, its attenuated object term, its accumulated medium term, and a medium-disabled object image. The last output is obtained by bypassing the learned attenuation and backscatter during rendering, but its object colors were still optimized only through underwater images. We therefore call it an *object diagnostic*; it is neither trained nor evaluated as physical clear-radiance restoration.

3.7 Properties and Scope

Homogeneous-grid optical-mass identity

For homogeneous channel-wise coefficient $\boldsymbol{\beta}$ in a box of volume V, an N^3 FCC grid has $4N^3$ points and cell volume $\Delta V = V/N^3$. Equation 3 gives

$$\sum_i \mathbf{m}_{D,i} = 4N^3 \left(\boldsymbol{\beta}\frac{\Delta V}{4}\right) = \boldsymbol{\beta}V, \tag{17}$$

independent of N, with the same identity for $\mathbf{m}_{B,i}$. This identity is over $\mathbb{R}^3$ because each ϕ_i is normalized there. It does not assert exact mass inside the finite center box: boundary Gaussians have nonzero outside tails. The CUDA renderer also uses finite projected support for culling, so the analytic identity characterizes the parameterization rather than guaranteeing pixelwise invariance across lattice resolutions.

Coordinate scaling

If scene coordinates are scaled by s, Gaussian means and standard deviations scale by s while density scales by $1/s$. Equation 9 is then invariant. All real-scene runs use the coordinate scale already stored by the shared data parser; no image or test metric is used to choose it.

Coverage and single-surface identity

For any positive-coverage ray and constant density, Eqs. 10–11 reproduce that density exactly even when the finite Gaussian sum does not integrate to the geometric path length. For a non-uniform field and one opaque surface at axial proxy depth z^*, with geometric distance $t^* = z^*/d_{p,z}^{\mathrm{cam}}$, the compositor reproduces the endpoint transmittance of the partition-of-unity-corrected integral, $\boldsymbol{\tau}_k t^*/q$, rather than the unnormalized finite-lattice integral $\boldsymbol{\tau}_k$. For multiple object depths, it distributes this endpoint-equivalent coefficient along the same ray and is generally approximate. The approximation is small when visibility is concentrated near one surface and can increase at occlusion boundaries or separated opaque layers.

4 Experiments

4.1 Experimental Setup

Real scenes and split

We use Curasao, IUI3-RedSea, Japanese Gardens, and Panama from SeaThru-NeRF [3]. Every method is optimized independently for each scene. Two filename-selected views per scene are withheld from all training runs, yielding eight full-resolution test images. All methods are scored against byte-identical decoded targets using one implementation of PSNR [28], SSIM [27], and AlexNet-LPIPS [29]. PSNR measures pixelwise fidelity, SSIM emphasizes local structure, and LPIPS measures perceptual feature distance; higher PSNR/SSIM and lower LPIPS are better. Because the compact water-field capacity was selected while these scenes were available, the comparison is a controlled development-set study rather than an independent leaderboard.

Independent RGB-blind confirmation

We separately reserve all three Underwater360 scenes as an independently held-out evaluation set [30]. The official even/odd split supplies 53/53, 91/90, and 76/76 train/test panoramas for Scenes 1, 3, and 4; every panorama is bilinearly projected to four fixed 480×480, 90° perspective faces. The final AM-GS configuration and WaterSplatting are trained for 10k iterations from the same complete-capture camera poses and official point cloud. No held-out RGB is decoded before all six checkpoints are hashed, after which the four faces are averaged per panorama, panoramas are averaged per scene, and scenes receive equal macro weight. Thus the test is RGB-blind but not geometry-blind; it measures transfer to new captures, not calibrated water removal or native spherical rendering.

Compared methods

We retrain the released 3D-UIR [9], SeaSplat [6], UW-GS [18], Gaussian Splashing [7], and WaterSplatting [8] pipelines on the same scene files and test lists as Ours. Method-native schedules are retained: 30k iterations for 3D-UIR, SeaSplat, and Gaussian Splashing, 15k for UW-GS, and 10k for WaterSplatting and Ours. Released preprocessing is retained, but no held-out RGB enters training, depth normalization, model selection, or pseudo-depth estimation. WaterClear-GS [10], Plenodium [16], and OceanSplat [19] are discussed as protocol-mismatched context rather than inserted into this ranking without common predictions and decoded targets. Plenodium is nonetheless reproduced in the paired simulation of Section 4.5, and the reason the two studies differ is the comparison protocol rather than a selective omission: its official implementation requires a vendored Nerfstudio tree with its own `gsplat` and PyTorch pins and was compiled for a different GPU architecture, so it cannot be retrained inside the single environment this table uses, whereas the paired simulation compares lossless exports rescored in one frozen metric environment and therefore tolerates per-method training stacks.

Implementation

AM-GS holds the inherited object Gaussians, optimizer, loss, and direction-conditioned asymptotic-color head fixed so that only the spatial water representation and query change. The final configuration uses an $N = 8$ fixed lattice, $\sigma/\Delta = 1.3$, coverage normalization, and geometric camera-ray distance, corresponding to 2,048 water Gaussians and 12,288 RGB transport parameters. All real-scene results use full input resolution and seed 0 unless a multi-seed result is explicitly identified. Eighteen analytic-renderer unit tests covering diagonal rays, finite segments, anisotropic bases, culling, and transformed cameras agree with independent float64 integration within 2.90×10^{-6}. All reported differences, intervals, and percentages are computed from archived full-precision per-view or per-field arrays; table cells are rounded only for display.

Paired simulation protocol

We additionally use the official Plenodium Beach and Street simulations at easy, medium, and hard water levels [16]. Each subset contains 100 posed 512×512 images and a pose-matched clean counterpart; explicit filenames freeze 50 degraded training views and 50 held-out views before any full run. The clean images are inaccessible to training and are read only by the final evaluator. We train Nerfstudio 3DGS [5] for 30k iterations, the official WaterSplatting [8] and Ours implementations for 15k, and the official depth-assisted Plenodium implementation [16] for 15,001 iterations with independently inferred Depth Anything V2 priors [31]. Every method receives the same degraded images, cameras, subset-specific sparse point cloud, and frozen test poses; no clean target, test image, or test metric selects a checkpoint or hyperparameter. We report the same PSNR, SSIM, and AlexNet-LPIPS implementations for (i) the rendered underwater image against its degraded target and (ii) the raw medium-disabled branch against the clean target. Ordinary 3DGS has no medium-separation branch, so its underwater RGB is retained as a clearly labeled no-separation reference in the second test. To remove training-environment differences, all methods export lossless 8-bit RGB PNGs and the paper tables rescore those exports in one frozen metric environment; method-native float metrics are retained only as a render audit.

4.2 Real-Scene Rendering Quality

Table 1 reports every scene and all three image metrics instead of hiding structural or perceptual losses behind macro averages. Ours has the highest average PSNR (27.871 dB) and lowest average LPIPS (0.1389) among the six reproduced methods, while WaterSplatting has the highest average SSIM by 0.0005. Relative to that baseline, Ours improves PSNR on Curasao, IUI3, and Japanese Gardens by 1.041, 0.230, and 0.212 dB, respectively, and is 0.357 dB lower on Panama. Its SSIM is higher on Curasao and IUI3 but lower on Japanese Gardens and Panama; LPIPS follows the same two-win/two-loss pattern. Thus the main table supports competitive real-scene reconstruction and a modest average advantage, not uniform dominance over the strongest matched baseline. The paired hierarchical-bootstrap PSNR interval [32], $[-0.213, +0.952]$ dB, includes zero and is reported as descriptive uncertainty over four reused scenes. Compared with 3D-UIR, SeaSplat, UW-GS, and Gaussian Splashing, Ours improves average PSNR by

Table 1 Complete per-scene SeaThru-NeRF comparison [3] using two common filename-selected holdouts, byte-identical targets, and shared PSNR/SSIM/AlexNet-LPIPS code [27–29]. Method-native schedules are retained. Ours was selected on these scenes, so this is development evidence. The first and second values are highlighted; best and second-best values remain bold and underlined for grayscale readability.

(a) PSNR↑

Method	Curasao	IUI3	Japanese	Panama	Avg.
3D-UIR [9]	28.579	23.330	25.123	29.144	26.544
SeaSplat [6]	26.722	22.923	24.172	22.411	24.057
UW-GS [18]	27.840	22.873	24.235	28.344	25.823
Gaussian Splashing [7]	27.105	22.183	24.088	27.933	25.327
WaterSplatting [8]	29.454	24.852	26.134	**29.917**	27.589
Ours	**30.495**	**25.083**	**26.346**	29.560	**27.871**

(b) SSIM↑

Method	Curasao	IUI3	Japanese	Panama	Avg.
3D-UIR [9]	0.8578	0.7794	0.9281	0.8812	0.8616
SeaSplat [6]	0.8423	0.7674	0.9166	0.8354	0.8404
UW-GS [18]	0.8435	0.7736	0.9224	0.8724	0.8530
Gaussian Splashing [7]	0.8319	0.7331	0.9101	0.8528	0.8320
WaterSplatting [8]	0.9104	0.8713	**0.9393**	**0.9049**	**0.9065**
Ours	**0.9128**	**0.8732**	0.9379	0.8998	0.9059

(c) LPIPS↓

Method	Curasao	IUI3	Japanese	Panama	Avg.
3D-UIR [9]	0.2025	0.2267	0.1122	0.1417	0.1707
SeaSplat [6]	0.2260	0.2339	0.1333	0.2272	0.2051
UW-GS [18]	0.2696	0.2335	0.1384	0.1516	0.1983
Gaussian Splashing [7]	0.3297	0.3076	0.1711	0.1930	0.2503
WaterSplatting [8]	0.1670	0.1903	**0.0921**	**0.1068**	0.1391
Ours	**0.1666**	**0.1874**	0.0928	0.1089	**0.1389**

1.327, 3.814, 2.048, and 2.544 dB, respectively; all four scene-level PSNR values are also higher than the reproduced 3D-UIR and Gaussian Splashing rows.

4.3 Independent Real-Scene Confirmation

On the three independently held-out scenes, Ours and WaterSplatting are visually and numerically close: the macro change is −0.056 dB PSNR, −0.000064 SSIM, and +0.00181 LPIPS. The predeclared 20,000-sample hierarchical bootstrap (seed 164) resamples scenes and then held-out panoramas after four-face averaging. Its PSNR interval, $[-0.133, +0.054]$ dB, crosses zero and the −0.1 dB non-inferiority margin; the corresponding LPIPS interval, $[+0.00024, +0.00335]$, is entirely unfavorable because lower LPIPS is better. On these independent captures Ours is therefore statistically

Table 2 RGB-blind, but not geometry-blind, confirmation on three independently held-out Underwater360 evaluation scenes [30]. Rows average four fixed faces per held-out panorama; the macro weights scenes equally. No held-out RGB is decoded before six checkpoint hashes. Common PSNR/SSIM/AlexNet-LPIPS code is used [27–29]. Within each scene row and metric, the better and worse values across WaterSplatting and Ours are highlighted as first and second, respectively. The predeclared Ours-minus-WaterSplatting [8] PSNR interval is $[-0.133, +0.054]$ dB; superiority/non-inferiority fail/fail.

	WaterSplatting [8]			Ours			Δ PSNR
Scene	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	Ours−WS
Scene 1	24.426	0.7362	**0.3194**	**24.488**	**0.7379**	0.3196	+0.062
Scene 3	**23.744**	**0.7434**	**0.3752**	23.646	0.7424	0.3786	-0.098
Scene 4	**24.362**	**0.7576**	**0.3711**	24.230	0.7566	0.3730	-0.132
Macro	**24.178**	**0.7457**	**0.3553**	24.121	0.7457	0.3571	-0.056

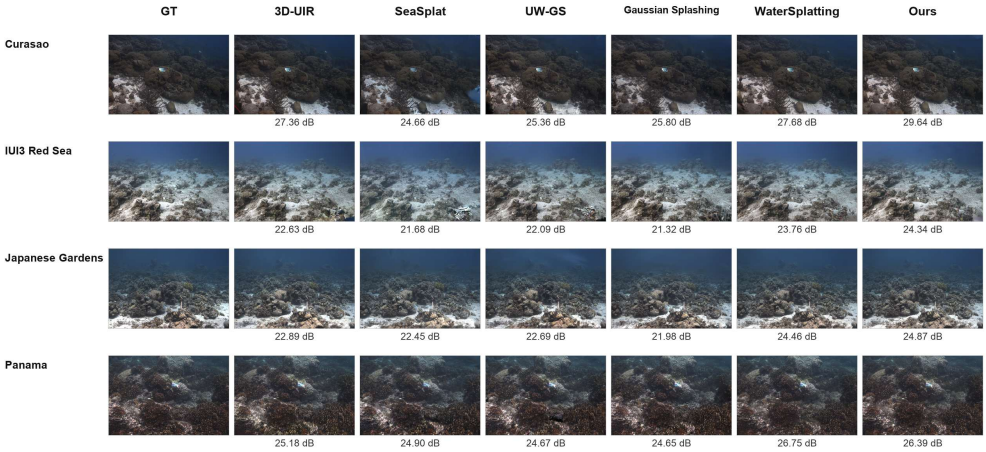


Fig. 2 Complete held-out frames from all four SeaThru-NeRF scenes [3]. Columns show ground truth and protocol-matched 3D-UIR [9], SeaSplat [6], UW-GS [18], Gaussian Splashing [7], WaterSplatting [8], and Ours. The immutable evaluator-index-0 frame is used for every scene. Every complete image preserves its native aspect ratio and receives identical resizing without cropping, color adjustment, sharpening, or method-specific processing. Labels report full-frame PSNR [28].

indistinguishable from the strongest baseline in PSNR, with a small but consistently unfavorable perceptual difference. This confirmation is retained because it bounds the claim to the water representation rather than a broad NVS advantage.

4.4 Full-Image and Local Qualitative Comparison

Figure 2 shows complete frames before any crop is considered. It exposes both the three PSNR-favorable scenes and the Panama loss, preventing a selected detail from standing in for whole-image behavior.

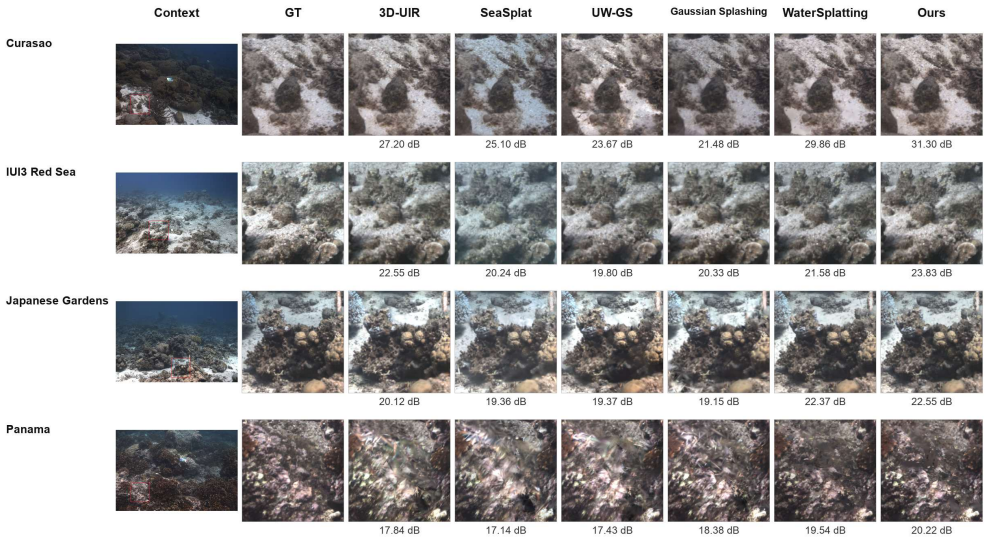


Fig. 3 Complete four-scene local-detail disclosure for the same views and methods as Fig. 2. One square per scene is selected only from low-pass, percentile-clipped luminance-gradient structure in the decoded ground truth, outside a fixed border; no prediction selects the view or crop. The source square occupies 24% of image height and is copied to every method before aspect-preserving enlargement. Labels give crop PSNR [28]; no crop receives color, sharpness, or geometric modification.

Relative to WaterSplatting, local PSNR changes by +1.43, +2.24, +0.18, and +0.68 dB on Curasao, IUI3, Japanese Gardens, and Panama, respectively. The corresponding MAE changes from 0.0229 to 0.0196, 0.0703 to 0.0539, 0.0502 to 0.0486, and 0.0696 to 0.0608. These identical ground-truth-selected regions make the strongest improvement visible on IUI3 while retaining the smaller Japanese Gardens gain and the full-frame Panama loss from Fig. 2.

The shared error scale reveals where the numeric gain comes from: Ours reduces broad high-error regions around the pale seabed in IUI3 and suppresses local structure errors in the Curasao and Panama crops. The complete four-scene crop panel remains Fig. 3; Fig. 4 is not presented as an unbiased scene subset.

Figure 5 confirms that the explicit water branch participates in every scene. It also shows why a paired hidden-clean simulation is required: the jointly optimized object branch can compensate for water appearance, so disabling the medium in an unpaired real capture is not by itself proof of correct water removal.

4.5 Paired Underwater and Hidden-Clean Agreement

Table 3 separates two questions that cannot be answered by the unpaired real captures. The underwater panel measures end-to-end novel-view reconstruction under the observed medium, whereas the hidden-clean panel tests how closely each method's raw medium-disabled branch approaches the pose-matched clean scene that was hidden throughout training. The six water–scene subsets and all 50 held-out views per subset

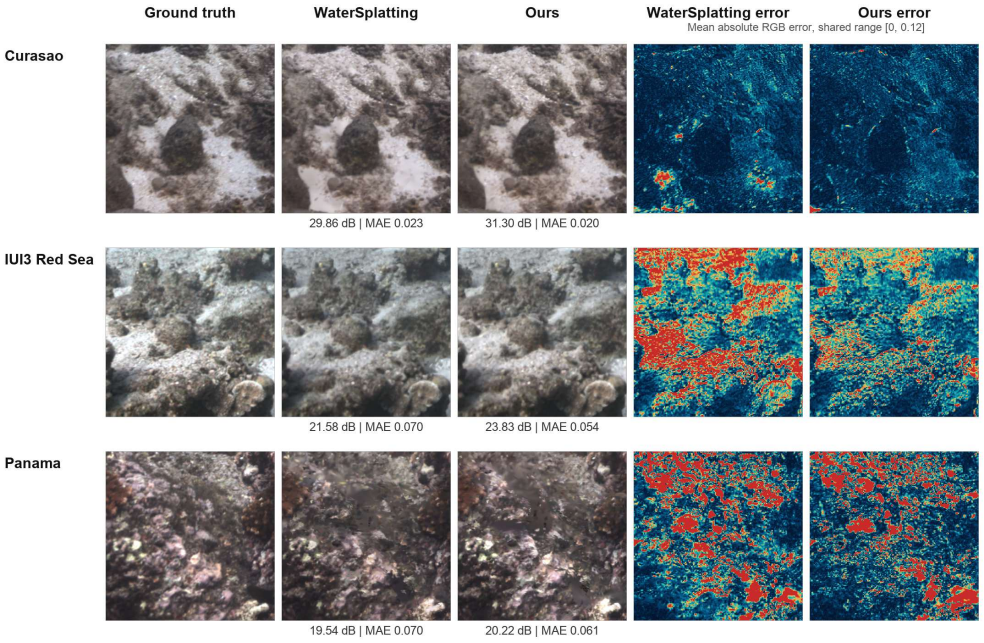


Fig. 4 Explanatory error view for the three clearest local Ours-versus-WaterSplatting gains in Fig. 3. This three-scene subset was selected after inspecting the complete four-scene measurements and is therefore used only to explain, not to estimate, a win rate. Each pair uses the same ground-truth-only crop as Fig. 3; per-pixel mean absolute RGB error is displayed under one shared $[0, 0.12]$ scale. Darker maps, higher crop PSNR [28], and lower MAE indicate closer agreement.

Table 3 Official paired Plenodium simulation [16]: Beach (B) and Street (S), each at easy (E), medium (M), and hard (H) water. Methods share 50 degraded training and 50 frozen test poses per subset. Columns give subset PSNR and six-subset macro PSNR/SSIM/AlexNet-LPIPS [27–29]. Panel (a) targets underwater RGB; (b) scores the raw medium-disabled branch against pose-matched clean RGB withheld from training. 3DGS uses ordinary RGB as a disclosed no-separation reference. To separate the primary medium-disentanglement result from reconstruction context, color highlights are used only in panel (b); best and second-best values remain bold and underlined in both panels.

Method	B-E	B-M	B-H	S-E	S-M	S-H	Mean PSNR↑	Mean SSIM↑	Mean LPIPS↓
(a) Held-out underwater reconstruction									
3DGS [5]	41.168	42.690	44.752	37.896	39.480	40.604	41.098	0.9784	0.0211
WaterSplatting [8]	41.323	42.542	43.869	**37.949**	39.856	**41.205**	41.124	0.9824	0.0224
Plenodium [16]	**41.907**	**43.825**	**45.474**	37.911	**39.965**	41.026	**41.685**	**0.9826**	**0.0177**
Ours	39.861	41.759	44.042	37.263	38.997	40.391	40.385	0.9824	0.0208
(b) Raw medium-disabled branch versus hidden clean target									
3DGS [5]	19.239	14.560	12.019	22.659	19.693	18.353	17.754	0.7215	0.2550
WaterSplatting [8]	13.581	10.874	10.191	26.165	**23.189**	19.966	17.328	0.7160	0.1848
Plenodium [16]	24.265	20.412	15.058	**26.739**	22.542	19.606	**21.437**	0.8089	**0.1129**
Ours	**27.095**	**21.441**	**15.545**	20.260	21.759	**20.846**	21.158	**0.8299**	0.1281

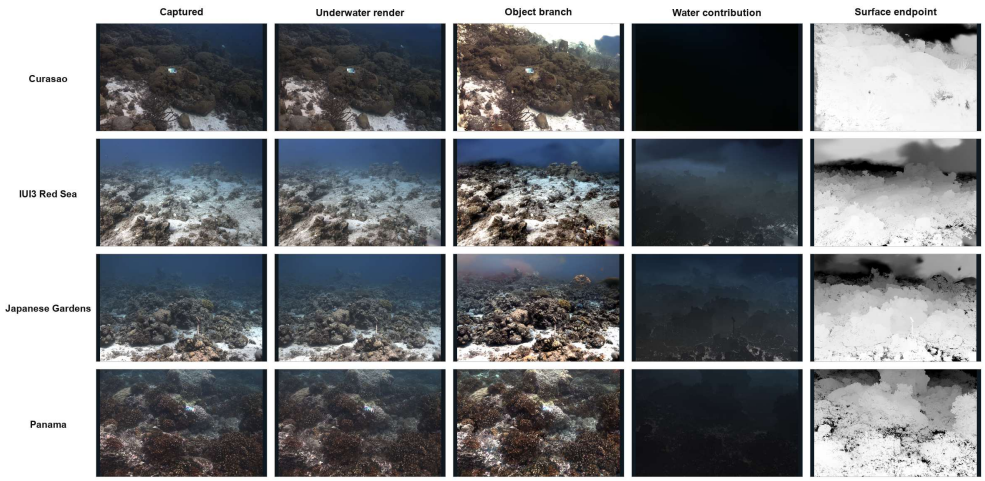


Fig. 5 Diagnostic branches from the final model on all four real scenes [3]: captured image, supervised underwater render, medium-disabled object branch, raw accumulated-water contribution, and the detached endpoint used by the finite-path query. Images preserve their full aspect ratio, and the water contribution receives no display gain. Because real captures provide no calibrated clear target, the object-branch column is a decomposition diagnostic rather than a claimed restored image.

contribute to the macro equally; no result is selected from a favorable severity or camera. For underwater reconstruction, official Plenodium ranks first in macro PSNR, SSIM, and LPIPS at 41.685 dB, 0.9826, and 0.0177, respectively; Ours reaches 40.385 dB, 0.9824, and 0.0208. The deficit is concentrated in PSNR, while the SSIM gap is only 0.0002. The ordinary 3DGS reference also reaches 41.098 dB in this panel, so Ours is not presented as the best degraded-RGB renderer on this synthetic protocol. This pattern is consistent with the scope of our intervention: Ours retains the WaterSplatting object branch, directional color head, and compositor and changes the water carrier, whereas Plenodium jointly redesigns spatial medium features, depth-assisted initialization, and geometry regularization for this paired synthetic protocol. The result therefore shows that a stronger carrier alone does not guarantee the best degraded-image reconstruction when the remaining renderer and geometry pipeline are held fixed. The ranking changes when the raw medium branch is disabled and the hidden clean target is used. Ours gives the highest SSIM (0.8299), the second-highest PSNR (21.158 dB, 0.279 dB below Plenodium), and the second-lowest LPIPS (0.1281). Relative to WaterSplatting, it raises PSNR by 3.830 dB and SSIM by 0.1139 while lowering LPIPS by 0.0567; it wins 4/6, 5/6, and 4/6 scene–severity subsets on those metrics. Relative to Plenodium, Ours wins 4/6 PSNR and 5/6 SSIM subsets, but Plenodium wins 5/6 LPIPS subsets and the macro PSNR/LPIPS values. The result therefore supports stronger structural separation than the inherited direction-conditioned coefficient carrier, not calibrated recovery or perceptual-color leadership.

Figures 6 and 7 provide whole-image and local evidence under the same frozen selection rule. In Beach, Ours avoids the severe dark residual retained by WaterSplatting and preserves the table, bottle, and umbrella support more closely than the other raw

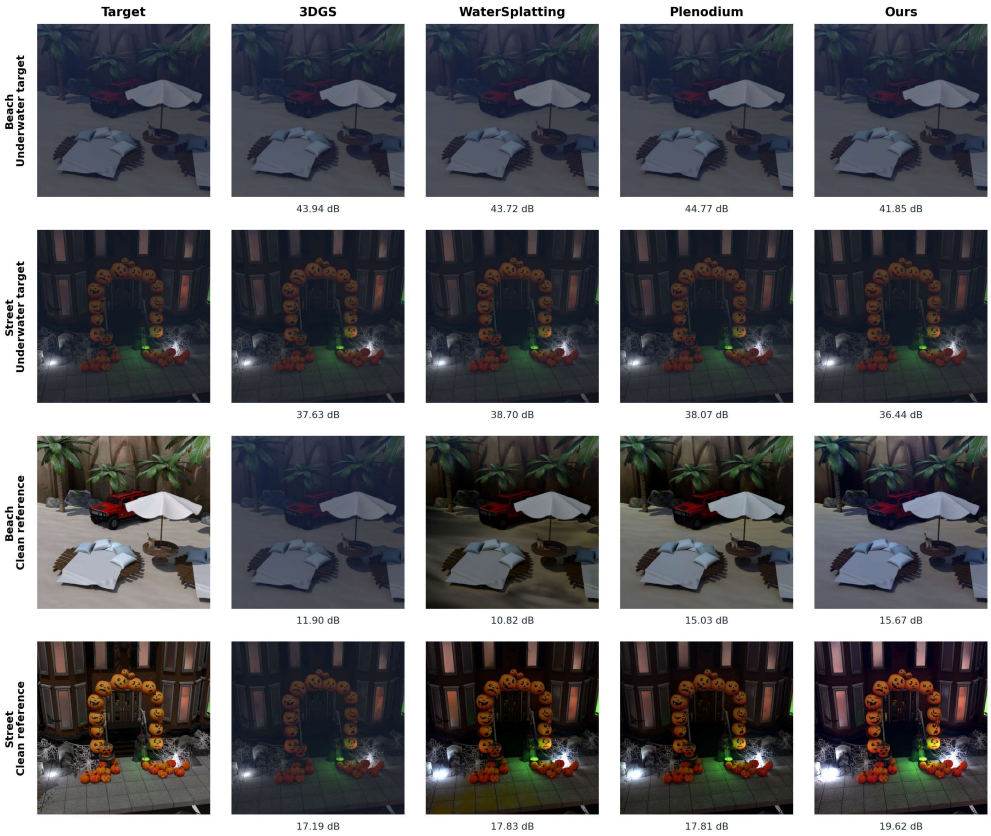


Fig. 6 Complete held-out frames on the official Plenodium simulation [16] under hard water. Columns compare the target with Nerfstudio 3DGS [5], WaterSplatting [8], the official Plenodium implementation [16], and Ours. The upper two rows target the degraded underwater image; the lower two target the pose-matched clean image using each method's raw medium-disabled branch. Since 3DGS has no medium-separation branch, its ordinary RGB is retained as a disclosed no-separation reference. Beach and Street use one ground-truth-only view fixed before formal method predictions were read. Every panel is the complete 512×512 export, shown without crop, geometric deformation, color adjustment, or sharpening. Labels report full-frame PSNR [28].

outputs at medium and hard water. In Street, the ranking is mixed: Ours gives the best full-frame hard-water PSNR, whereas the two baselines are closer on the selected medium-water detail. These examples are interpreted together with all-level quantitative results rather than as a replacement for them. Complete per-subset three-metric tables and all six full hidden-clean comparisons appear in the Appendix.

4.6 Controlled Water-Medium Representation

Real images can score a rendered view but cannot reveal the unknown three-dimensional attenuation and backscatter field. We therefore use 24 procedurally generated local water fields selected by a frozen support-only rule from a 4,096-field pool before

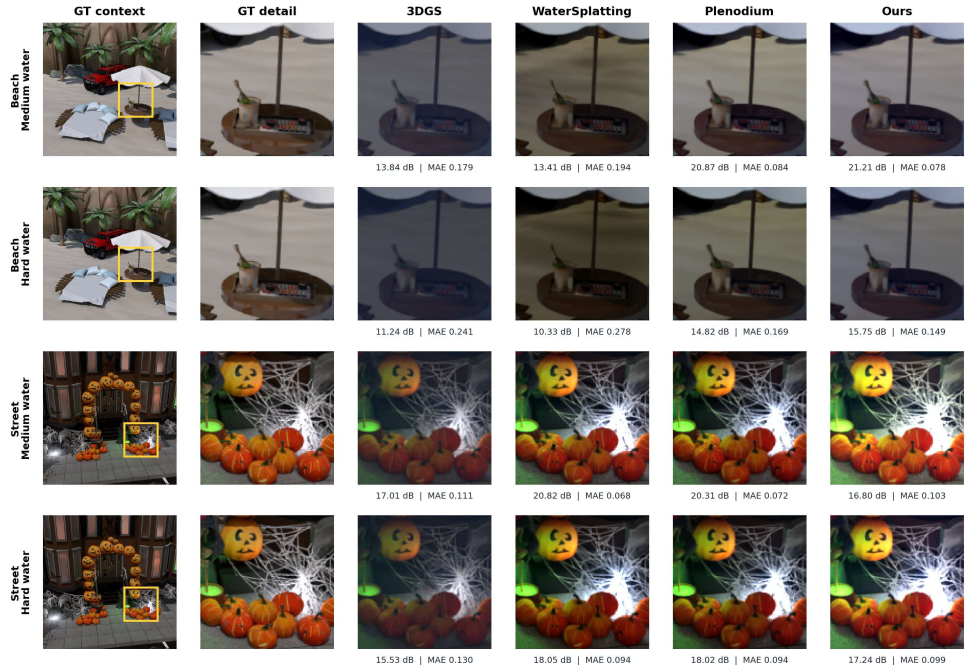


Fig. 7 Medium-disabled details against the hidden clean target on Beach and Street at medium and hard water [16]. Columns show Nerfstudio 3DGS [5], WaterSplatting [8], official Plenodium, and Ours. Each scene's view and 128×128 square are selected once from clean ground-truth structure before predictions are read, then copied unchanged to every method and severity. The context marks that square; detail pixels receive no geometric or color enhancement. Since 3DGS has no separation branch, its ordinary RGB is the disclosed reference. Labels give crop PSNR [28] and RGB MAE; crops explain differences but do not enter Table 3.

any carrier basis or prediction is built. Each field contains two to four randomly rotated, optionally hollow compact superquadrics inside a declared water box, with no observation noise; Appendix D gives the complete distributions, camera geometry, quadrature, objective, and stopping rule. The primary split preserves ray directions while moving to five unseen origins; a secondary split changes origin and direction jointly. The positive 13^3 trilinear grid has 13,182 RGB transport parameters, while each $N = 8$ FCC Gaussian carrier has 12,288, so the voxel comparison is approximate rather than exact capacity matching. All carriers use one deterministic nonnegative solver and the same nominal ℓ_2 coefficient, not an assumption of equal effective smoothness. This experiment follows the controlled-phantom logic used to validate participating-media image formation [4, 16], but evaluates a different endpoint: whether a local optical change remains spatially localized after the camera moves.

Edited-transmittance RMSE measures the pixelwise error after removing the known local component; lower is better. Support PR-AUC asks whether rays that truly cross the changed component receive the largest predicted change, and soft IoU measures overlap between the predicted and true affected-ray regions; higher is better. Let e_i^G and e_i^V be the compact-Gaussian and voxel RMSE for field i. The estimand was prospectively frozen before evaluation in the supplementary R146 protocol lock

Table 4 Prospectively frozen positive-basis test on 24 independently generated procedural field instances. Rows fit identical noiseless finite-path observations with nonnegative projected FISTA and the same nominal ℓ_2 coefficient; the voxel is only approximately parameter-matched, and equal effective smoothness is not assumed. "Origin" changes camera position only; "joint" also changes direction. PR-AUC [33] and soft IoU [34] score affected-ray localization. The compact width is a mechanism setting, not the final model width. The first and second values are highlighted; best values remain bold.

		Unseen origin			Unseen joint 6-DoF		
Carrier	Parameters	T RMSE↓	PR-AUC↑	IoU↑	T RMSE↓	PR-AUC↑	IoU↑
Positive trilinear voxel	13,182	0.001499	0.9956	0.9219	0.002511	0.9674	0.6485
Optical-mass Gaussian, $\sigma/\Delta = 0.45$	12,288	0.001552	0.9936	0.8787	0.001697	0.9861	0.7938
Compact optical-mass Gaussian, $\sigma/\Delta = 0.30$	12,288	**0.000702**	**0.9984**	**0.9491**	**0.001228**	**0.9926**	**0.8010**

(SHA-256 prefix `3a3fbac070b8`): $24^{-1}\sum_i(1 - e_i^G/e_i^V) = 52.68\%$; a 200,000-resample paired-field bootstrap gives 48.37–56.82%, and the Gaussian wins all 24 fields. For comparison, taking the ratio only after averaging the full-precision RMSEs in Table 4 gives $1 - 0.000701736/0.001498573 = 53.17\%$; this different aggregate-ratio estimand is not used for the headline. The broader $\sigma/\Delta = 0.45$ Gaussian is slightly worse than the voxel carrier on the primary origin-only split (RMSE 0.001552 versus 0.001499, PR-AUC 0.9936 versus 0.9956, and soft IoU 0.8787 versus 0.9219), so the 52.7% result should not be read as evidence for the final $\sigma/\Delta = 1.3$ deployment width. It also has the best localization scores and retains a 51.08% RMSE advantage under the joint six-degree-of-freedom change. The equal-capacity broad and compact Gaussian rows isolate kernel width; the approximately matched voxel row compares basis families under one observation and solver protocol, without asserting equal smoothness. Because this prospective mechanism test uses $\sigma/\Delta = 0.30$ whereas the final real-scene model uses 1.3, it supports the localizability of optical-mass Gaussians but is not presented as an end-to-end proof that the learned real-water field is physically correct.

4.7 End-to-End Known-Water Audit

The controlled carrier test supplies true optical observations directly, whereas real training observes only the final RGB image. To test whether the jointly optimized system nevertheless identifies water, we transfer the exact final representation to three frozen Blender scenes with 105 underwater training views, 15 held-out views, hidden clean radiance, true depth, and spatially varying direct and backscatter fields. The only protocol adaptation is the declared linear-RGB image formation of this synthetic set; hidden clean images and optical maps remain inaccessible to training. Complete per-scene measurements and raw full-frame comparisons are reported in Table 10 and Fig. 9 of Appendix B.

Monster and Junkshop reconstruct the underwater target accurately, yet their medium-disabled and optical errors remain large; Classroom also fails the transferred underwater rendering configuration. Across all three scenes, underwater PSNR is 32.086 dB, but the medium-disabled branch reaches only 12.457 dB against hidden clean radiance and direct/backscatter transmittance RMSE is 0.2784/0.3054. Thus a physically parameterized carrier can be fitted as part of a good RGB renderer without identifying the generating water field. This result is deliberately retained: Table 4 establishes controlled representation capacity, while Table 10 shows that RGB-only

Table 5 Component analysis on the four SeaThru-NeRF scenes [3]. All rows use the same eight held-out images and metric implementations [27–29]. Panel (a) changes the medium and color carriers inside the finite-surface renderer. Panel (b) follows the compact spatial carrier through the subsequent fixed-kernel and rendering-operator choices. Rows are separately optimized unless stated otherwise; therefore, non-monotonic changes are reported rather than interpreted as isolated causal gains. The first and second values are highlighted within each panel; best values remain bold.

Variant	PSNR↑	SSIM↑	LPIPS↓
(a) Representation components			
WaterSplatting reference [8]	27.589	**0.9065**	**0.1391**
Ours + homogeneous medium + global color	27.566	0.9043	0.1421
Ours + spatial medium + global color	27.754	0.9019	0.1489
Ours + spatial medium + directional color ($N = 16$)	27.437	0.9037	0.1414
Ours + compact spatial medium + directional color ($N = 8$)	**27.759**	0.9038	0.1416
(b) Compact-carrier refinements			
Ours + compact carrier ($\sigma/\Delta = 0.7$)	27.759	0.9038	0.1416
Ours + wider kernels ($\sigma/\Delta = 1.3$)	27.770	0.9041	0.1404
Ours + coverage normalization	**27.937**	**0.9062**	0.1393
Ours + geometric ray distance (final)	27.871	0.9059	**0.1389**

object–medium optimization does not convert that capacity into calibrated physical recovery.

4.8 Ablation and Repeatability

Table 5(a) separates spatial transport from the retained directional asymptotic-color head. At $N = 16$ with global color, replacing homogeneous transport by spatial transport raises average PSNR from 27.566 to 27.754 dB but reduces SSIM and increases LPIPS, showing that real-view metrics alone do not identify the water field. Re-enabling directional color recovers structural and perceptual quality, while the compact $N = 8$ carrier gives the best PSNR in this panel. Panel (b) shows that the rendering refinements are also non-monotonic: the $\sigma/\Delta = 0.7 \rightarrow 1.3$ change improves all three point estimates, coverage normalization gives the highest PSNR and SSIM, and geometric distance gives the lowest LPIPS but slightly reduces PSNR. The width change is small relative to the three-seed variation reported below (0.011 dB versus 0.040 dB PSNR standard deviation, and 0.0012 LPIPS versus 0.0012 standard deviation), so it is not treated as statistically decisive model selection. We nevertheless use $\sigma/\Delta = 1.3$ and geometric distance in the final model because the former is the retained production configuration and the latter converts camera-space axial depth to the actual distance travelled along each ray, preserving the physical unit of optical depth; the 0.066 dB PSNR decrease relative to coverage normalization is an explicit accuracy–consistency trade-off rather than an unreported selection. Because these are independently optimized checkpoints, the rows support the final design choice but do not isolate a pure test-time causal gain or establish that the selected width is superior under stochastic variation.

The final $N = 8$, $\sigma/\Delta = 1.3$, POU, geometric-distance configuration is repeated from scratch with seeds 1 and 2 and combined with the frozen seed-0 run. Across the runs, Ours reaches 27.849±0.040 dB and 0.1388±0.0012 LPIPS versus 27.533±0.050 dB

and 0.1399±0.0007 for WaterSplatting, leading three of four scene PSNR means; the baseline is higher in SSIM by 0.00016. The repeat measures stochastic stability only; the four development scenes remain the unit of scientific generalization, so the small seed variance cannot establish cross-scene superiority.

4.9 Numerical Properties and Efficiency

The numerical tests separate analytic correctness from image quality. They verify the homogeneous optical-mass identity, coordinate-scale invariance, finite-segment integration, and expected error near separated surfaces and occlusion boundaries. For four known coefficient fields and three severities, water Gaussians reduce direct and backscatter optical-depth RMSE by 70.06% and 66.87% relative to a direction-only oracle under identical finite paths. That oracle receives only the unit ray direction and path length: it fits one RGB coefficient for each sampled image direction jointly over all camera origins, with neither camera origin nor a 3D query coordinate, and is not an official WaterSplatting reproduction. These known-coefficient tests are unit tests, not learned image restoration results. Complete numerical results are provided in Tables 11 and 12 of Appendix C.

The exact final real-scene model reaches 19.12 render-only FPS on one RTX 5090 versus 239.63 FPS for WaterSplatting, while peak allocated memory rises from 1120.2 to 1449.5 MiB. The current dense integration kernel is therefore a representation prototype rather than a speed contribution. Its cost is dominated by the per-pixel sum over water primitives, which grows with lattice size; because the reported kernel is wider than the lattice spacing, support culling cannot reduce that sum without perturbing the integral, so the practical route to real-time evaluation is quadrature on a precomputed field, as discussed in Section 5.

4.10 Reproducibility

The supplementary archive records source snapshots, commands, configurations, split manifests, checkpoint identities, decoded-target hashes, full-precision per-view metrics, and figure builders. Every main quantitative cell is generated from a machine-readable record, and qualitative crops store their source paths and coordinates.

Table 6 Three-seed repeat on four SeaThru-NeRF development scenes [3]. Ours uses the exact final configuration; WaterSplatting reuses its unchanged reference. Scene columns are seed-averaged PSNR and macro cells are mean±sample standard deviation. All rows share eight decoded targets and metric code [27–29]. Hash-matched Ours repeats ran on RTX 5090 and RTX A5000 GPUs, so variation includes seed and possible hardware nondeterminism. The first and second values are highlighted; bold marks the better mean.

Method	Curasao	IUI3	Japanese	Panama	PSNR↑	SSIM↑	LPIPS↓
WaterSplatting [8]	29.412	24.796	26.158	**29.765**	27.533±0.050	**0.9061**±0.0004	0.1399±0.0007
Ours: exact final configuration	**30.371**	**25.009**	**26.348**	29.666	**27.849**±0.040	0.9060±0.0006	**0.1388**±0.0012

Table 7 Exact-final efficiency on one RTX 5090. FPS excludes metrics, file output, and data loading; memory is peak CUDA allocation, and time is the four-scene mean to 10k. The first and second values are highlighted.

Method	Forward FPS↑	Rel.	MiB↓	Time min.↓	Rel.
WaterSplatting [8]	239.63	1.000×	1120.2	2.55	1.00×
Ours	19.12	0.080×	1449.5	51.16	20.03×

5 Scope and Limitations

AM-GS uses a single-scattering emission–absorption model and one visible-surface endpoint per pixel; multiple scattering, refraction, polarization, unknown camera response, and separated translucent layers remain outside its scope. Because training uses only underwater RGB, object radiance and water transport are not fully identifiable, so the medium-disabled output is a diagnostic rather than calibrated restoration. Evaluation covers four development scenes, three independent confirmation scenes, and controlled synthetic fields; broader environments, geometry-aware evaluation, and dynamic-scene handling remain future work [21]. Finally, the current dense integration kernel is slower than WaterSplatting, and the reported kernel width explains why support culling is not the remedy: at $\sigma/\Delta = 1.3$ the three-deviation support radius is 3.9 lattice spacings across a lattice of eight, so culling leaves most sites in play, and a tighter cutoff would perturb the integral beyond the tolerance audited in Section 4.9. That same width band-limits $\boldsymbol{\sigma}_D$ and $\boldsymbol{\sigma}_B$, which are therefore resolved by a grid only a few times finer than Δ. Precomputing both fields on such a grid by separable convolution and integrating by quadrature over the same finite segment would make the per-pixel cost independent of the number of water primitives; the analytic kernel would then be retained as the reference against which that fast path is audited.

6 Conclusion

AquaMass-GS replaces inherited predicted transport coefficients with normalized free-space water Gaussians and a finite-segment analytic query while retaining the object renderer. On four development scenes, the six-method comparison and exact-final three-seed repeat place Ours first in mean PSNR and LPIPS but second in SSIM; complete frames and fixed target-selected crops retain scene-level losses. On paired simulation, the medium-disabled branch ranks first in SSIM and second in PSNR/LPIPS, while Plenodium leads underwater rendering and hidden-clean perceptual distance. Across 24 independently generated known fields, compact Gaussians with $\sigma/\Delta = 0.30$ reduce mean fieldwise unseen-origin edited-transmittance RMSE by 52.7% versus a positive trilinear voxel carrier, although this controlled width differs from the final real-scene model. Independent-scene and known-water audits delimit this conclusion: the present evidence supports a queryable water representation and competitive per-scene rendering, while calibrated physical recovery remains an open problem.

A Complete Paired-Simulation Results

Tables 8 and 9 disclose PSNR, SSIM, and LPIPS for every scene–severity subset used in the paired simulation. Figure 8 shows the complete medium-disabled comparison against the hidden clean target for all six subsets under the same ground-truth-only view selection used in the main text.

Table 8 Complete held-out underwater reconstruction results on the official paired Plenodium simulation [16]. Beach (B) and Street (S) each use easy (E), medium (M), and hard (H) water. Every cell averages all 50 frozen held-out views with common PSNR, SSIM, and AlexNet-LPIPS implementations [27–29]. The final column is the equal-weight mean over all six subsets. The first and second values are highlighted; best and second-best values remain bold and underlined.

Method	B-E	B-M	B-H	S-E	S-M	S-H	Avg.
PSNR↑							
3DGS [5]	41.168	<u>42.690</u>	<u>44.752</u>	37.896	39.480	40.604	41.098
WaterSplatting [8]	<u>41.323</u>	42.542	43.869	**37.949**	<u>39.856</u>	**41.205**	<u>41.124</u>
Plenodium [16]	**41.907**	**43.825**	**45.474**	<u>37.911</u>	**39.965**	<u>41.026</u>	**41.685**
Ours	39.861	41.759	44.042	37.263	38.997	40.391	40.385
SSIM↑							
3DGS [5]	0.9827	0.9847	0.9873	0.9673	0.9723	0.9762	0.9784
WaterSplatting [8]	0.9836	<u>0.9863</u>	0.9885	**0.9743**	**0.9793**	<u>0.9822</u>	0.9824
Plenodium [16]	**0.9848**	**0.9869**	**0.9891**	0.9737	<u>0.9791</u>	0.9821	**0.9826**
Ours	<u>0.9842</u>	0.9862	<u>0.9886</u>	<u>0.9738</u>	0.9790	**0.9824**	<u>0.9824</u>
LPIPS↓							
3DGS [5]	<u>0.0147</u>	<u>0.0153</u>	<u>0.0145</u>	0.0242	0.0280	0.0299	0.0211
WaterSplatting [8]	0.0230	0.0213	0.0175	<u>0.0232</u>	<u>0.0240</u>	0.0257	0.0224
Plenodium [16]	**0.0125**	**0.0124**	**0.0117**	**0.0231**	**0.0228**	**0.0239**	**0.0177**
Ours	0.0170	0.0190	0.0148	0.0239	0.0251	<u>0.0247</u>	<u>0.0208</u>

Table 9 Complete raw medium-disabled branch agreement with the hidden clean target on the official paired Plenodium simulation [16]. Beach (B) and Street (S) each use easy (E), medium (M), and hard (H) water. Every cell averages all 50 frozen held-out views with common PSNR, SSIM, and AlexNet-LPIPS implementations [27–29]. The final column is the equal-weight mean over all six subsets. The first and second values are highlighted; best and second-best values remain bold and underlined.

Method	B-E	B-M	B-H	S-E	S-M	S-H	Avg.
PSNR↑							
3DGS [5]	19.239	14.560	12.019	22.659	19.693	18.353	17.754
WaterSplatting [8]	13.581	10.874	10.191	<u>26.165</u>	**23.189**	<u>19.966</u>	17.328
Plenodium [16]	<u>24.265</u>	<u>20.412</u>	<u>15.058</u>	**26.739**	<u>22.542</u>	19.606	**21.437**
Ours	**27.095**	**21.441**	**15.545**	20.260	21.759	**20.846**	<u>21.158</u>
SSIM↑							
3DGS [5]	0.8865	0.7825	0.6704	0.7614	0.6498	0.5782	0.7215
WaterSplatting [8]	0.7739	0.6546	0.5553	<u>0.8510</u>	<u>0.8043</u>	<u>0.6571</u>	0.7160
Plenodium [16]	<u>0.9351</u>	<u>0.8795</u>	<u>0.7687</u>	**0.8712**	0.7532	0.6459	<u>0.8089</u>
Ours	**0.9381**	**0.8949**	**0.8287**	0.7941	**0.8063**	**0.7177**	**0.8299**
LPIPS↓							
3DGS [5]	0.1479	0.2659	0.3605	0.1449	0.2573	0.3534	0.2550
WaterSplatting [8]	0.1993	0.2405	0.2900	<u>0.0653</u>	<u>0.1302</u>	<u>0.1835</u>	0.1848
Plenodium [16]	**0.0586**	**0.0922**	**0.1563**	**0.0632**	**0.1208**	0.1863	**0.1129**
Ours	<u>0.0647</u>	<u>0.1163</u>	<u>0.1846</u>	0.0914	0.1424	**0.1693**	<u>0.1281</u>

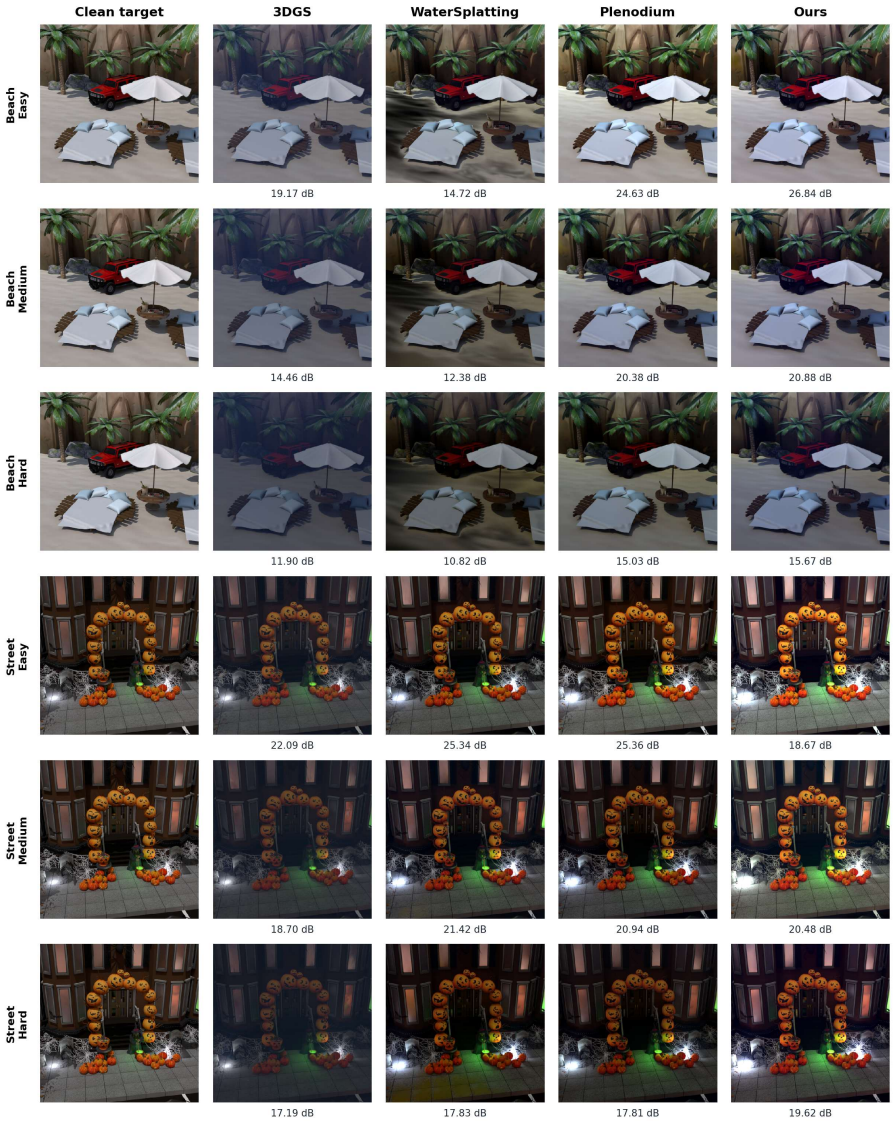


Fig. 8 Complete medium-disabled comparison against the hidden clean target for Beach and Street at easy, medium, and hard water levels on the official Plenodium simulation [16]. The view for each scene is selected from the clean ground truth and frozen before formal method predictions are read. Every panel preserves the full 512×512 image without crop, deformation, color adjustment, or sharpening. Numbers are full-frame PSNR [28]. Nerfstudio 3DGS [5] has no medium-separation branch, so its ordinary underwater RGB is retained as a disclosed no-separation reference; the remaining columns use the raw medium-disabled branch of WaterSplatting [8], Plenodium [16], and Ours.

B Known-Water Identifiability Audit

Table 10 and Fig. 9 report the complete end-to-end audit introduced in Section 4.7. The hidden clean images and optical maps are used only after optimization for evaluation.

Table 10 End-to-end known-water audit on three frozen Blender scenes synthesized with spatially varying direct attenuation and backscatter [4]. The exact final representation is transferred unchanged except that image formation is evaluated in the dataset's declared linear-RGB space. Training uses 105 underwater images per scene; 15 hidden clean images and true optical maps are read only after training. Metrics use common PSNR/SSIM/AlexNet-LPIPS implementations [27–29]. T_D and T_B are direct and backscatter transmittance RMSE. This is an identifiability audit, not a superiority table.

	Underwater reconstruction			Medium-disabled vs. hidden clean			Known optics	
Scene	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	T_D RMSE↓	T_B RMSE↓
Monster	38.946	0.9775	0.0180	14.251	0.7951	0.3735	0.1740	0.2501
Junkshop	36.755	0.9733	0.0224	10.976	0.4997	0.4217	0.3225	0.3744
Classroom	20.557	0.6668	0.5362	12.144	0.3535	0.7861	0.3386	0.2915
Macro	32.086	0.8726	0.1922	12.457	0.5494	0.5271	0.2784	0.3054

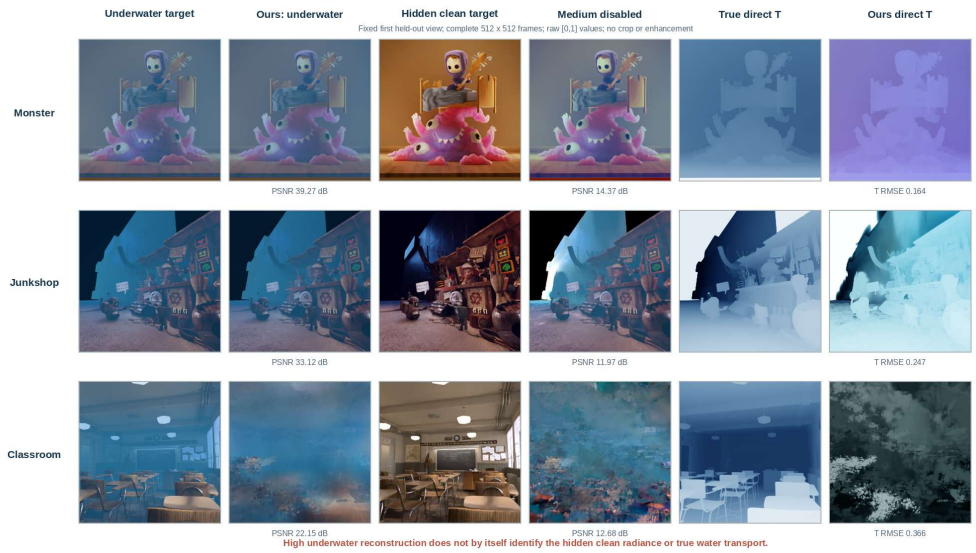


Fig. 9 Fixed first held-out view from every known-water scene in Table 10. All panels are complete 512×512 frames or RGB direct-transmittance maps in their raw $[0, 1]$ range; there is no crop, per-panel normalization, enhancement, or method-dependent processing. The first pair compares underwater reconstruction, the second compares the raw medium-disabled branch with the hidden clean target, and the final pair compares true and predicted direct transmittance. Numbers are full-frame PSNR [28] and transmittance RMSE.

C Numerical Audit Details

Tables 11 and 12 provide the complete unit-test evidence summarized in Section 4.9. They assess analytic and known-coefficient behavior, not learned restoration quality.

Table 11 Numerical audits. (a) Homogeneous assigned full-space mass and finite-box tails for normalized volumetric Gaussians [26]. (b) Independent endpoint surrogate under the revised underwater model [4], with PSNR [28]. (c) Optical-depth RMSE against Gauss–Legendre ground truth for continuous known fields over 16 unit rays and four geometric distances per field; the boundary subset contains 45 paths and POU evaluates $\boldsymbol{\tau} t^*/q$. Neither (b) nor (c) invokes the CUDA renderer or learned image factorization.

(a) Assigned mass and finite-box boundary

N	σ/Δ	Points	Assigned	Box mass	Tail
8	1.3	2,048	2.4000	1.5736	34.44%
8	0.7	2,048	2.4000	1.9130	20.29%
16	0.7	16,384	2.4000	2.1473	10.53%
32	0.7	131,072	2.4000	2.2713	5.36%
64	0.7	1,048,576	2.4000	2.3351	2.71%

(b) Finite-endpoint composition stress test

Visibility configuration	PSNR↑	MAE↓	Max.↓
Single opaque surface	120.00	$< 10^{-15}$	$< 10^{-15}$
Two opaque layers	40.87	0.00974	0.03155
Distributed low alpha	49.59	0.00334	0.01501
Occlusion edge	37.89	0.01278	0.04152
Proxy depth −20%	49.25	0.00374	0.01589

(c) Known-field raw finite support versus POU correction

Continuous field	Raw, all↓	POU, all↓	Raw, boundary↓	POU, boundary↓
Constant	0.08248	$< 10^{-12}$	0.09814	$< 10^{-12}$
Affine	0.08432	0.01439	0.10035	0.01716
Smooth plume	0.03718	0.01759	0.04267	0.01804
Layered	0.05241	0.01277	0.06244	0.01466

Table 12 Deterministic known-parameter operator check over four non-homogeneous fields, three severities, and three origins [4]. All rows use the same water segment. The direction oracle fits one RGB coefficient per sampled unit ray direction jointly across origins, using path length but no origin or 3D coordinate; it is not a WaterSplatting reproduction. Ours samples the known field at $N = 16$ Gaussian centers. This is a synthetic transport unit test, not a learned-image comparison. The first and second values are highlighted.

Carrier	Direct $\tau \downarrow$	Back. $\tau \downarrow$	Direct $T \downarrow$	Back. $T \downarrow$
Global oracle	0.10960	0.07930	0.06284	0.04712
Direction oracle	0.09507	0.06841	0.05451	0.04055
Known-coefficient Ours	**0.02847**	**0.02267**	**0.01622**	**0.01359**

D Controlled Carrier Protocol

The immutable supplementary record is `R146/freeze.json`, SHA-256 `3a3fbac070b82836a89acce4980c87fc33eaa2a6d69c34477a48374fde3c2726`.

Field generation

R146 uses box $[-1,1]^2\times[1,3]$ and draws 4,096 candidates with NumPy PCG64 seed 14620260810. The component count is discrete-uniform on $\{2,3,4\}$. For each component, center coordinates are independently uniform on $[-0.18,0.18]$, $[-0.16,0.16]$, and $[1.48,2.52]$; axis scales are independently uniform on $[0.18,0.38]$, $[0.16,0.34]$, and $[0.24,0.48]$; yaw, pitch, and roll are independently uniform on $[-\pi,\pi]$; and the superquadric power is discrete-uniform on $\{1.35,2,3.2,5\}$. The compact profile is $[1-r]_+^2$. With probability 0.35 it becomes hollow by subtracting $0.88[1-r/h]_+^2$ and rectifying, where $h\sim U(0,0.58)$; otherwise $h=0$. Each component has weight $U(0.62,1)$ and sinusoidal modulation with three frequencies $U(0.7,2.4)$, phase $U(-\pi,\pi)$, and range $0.84+0.16\sin(\cdot)$; its value is clipped to $[0,0.95]$, and components combine as $1-\prod_c(1-v_c)$. Independent field gains $g_D\sim U(0.90,1.30)$ and $g_B\sim U(0.95,1.42)$ multiply RGB base densities $(0.32,0.22,0.14)$ and $(0.14,0.20,0.28)$ for direct and backscatter optical depth. No observation noise is added.

Selection and scores

For each camera, target strength is the ℓ_2 norm of the six-channel direct/backscatter optical-depth change. A ray is affected at strength at least 0.012 and outside at strength at most 0.002. A candidate is accepted only if, separately for every camera in train, origin-only, direction-only, and joint splits, affected pixels are nonempty but not all pixels, outside pixels are nonempty, affected fraction lies in $[0.02,0.85]$, and outside fraction is at least 0.05. The first 24 passing candidates are fixed before any carrier basis or prediction is built. PR-AUC uses the binary affected label and predicted score $\|\widehat{\Delta\boldsymbol{\tau}}\|_2$. Soft IoU uses $a=\mathrm{clip}(\|\Delta\boldsymbol{\tau}\|_2/0.035,0,1)$ and $\hat{a}=\mathrm{clip}(\|\widehat{\Delta\boldsymbol{\tau}}\|_2/0.035,0,1)$, then $\sum\min(a,\hat{a})/\sum\max(a,\hat{a})$. Edited-transmittance RMSE compares $\exp[-\max(\boldsymbol{\tau}_0+\Delta\boldsymbol{\tau}-\widehat{\Delta\boldsymbol{\tau}},0)]$ with $\exp(-\boldsymbol{\tau}_0)$ over rays and six channels.

Cameras and integration

All training cameras have origin $(x,0,0)$ for $x\in\{-0.36,-0.12,0.12,0.36\}$, zero yaw/pitch, and look along world $+z$. Each uses a 21×21 pinhole grid with tangent half-FOV 0.20, giving 1,764 training rays. The origin-only test uses $(x,0,0)$ for $x\in\{-0.30,-0.18,0,0.18,0.30\}$ with the same orientation and pixels. The direction-only split reuses training origins with subpixel shift 0.43. The joint split uses the following $(x,y,z;\mathrm{yaw},\mathrm{pitch})$ values, with angles in degrees:

$$(-0.62,0.18,0.03;2,-1.5),\quad (-0.25,-0.20,0.08;-2,2),$$
$$(0.05,0.22,-0.06;2.5,1),\quad (0.32,-0.17,0.06;-2.5,-1),$$
$$(0.62,0.14,0;1,-2).$$

The subpixel shift is 0.31. Targets and bases use 256 midpoint samples over the identical ray–box intersection, without boundary padding.

Optimization and aggregation

The positive trilinear carrier has 13^3 sites and six RGB values per site; each FCC carrier has 2,048 centers and six values per center. For every field and carrier, deterministic

projected FISTA solves $\min_{X\geq 0} \operatorname{mean}(AX-Y)^2 + 10^{-7}\operatorname{mean}(X^2)$ for at most 20,000 steps. Checks occur every 100 steps and convergence requires relative KKT violation at most 10^{-3} and relative prediction change at most 2×10^{-4} twice consecutively. Fieldwise metrics are arithmetic means; the relative interval resamples 24 paired field reductions 200,000 times with seed 146910202608.

Statements and Declarations

Author Contributions. Jun Zhao: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Visualization, Writing – original draft. Jiafei Cao: Methodology, Software, Validation, Resources, Writing – review and editing. Rongfeng Lu: Data curation, Validation, Resources, Writing – review and editing. Longjian Zeng: Conceptualization, Methodology, Supervision, Project administration, Writing – review and editing. Yiming Cui: Software, Validation, Visualization. Zhangyi Yang: Software, Investigation, Visualization. Xichun Sheng: Data curation, Validation. Yahong Chen: Resources, Validation, Writing – review and editing.

Funding. This work was supported by the National Natural Science Foundation of China (Grant No. 12371349) and the Joint Fund of Zhejiang Provincial Natural Science Foundation of China (Grant No. LLSSZ25F030002).

Competing Interests. The authors declare that they have no known competing financial or non-financial interests that could have appeared to influence the work reported in this paper.

Ethics Approval and Consent to Participate. Not applicable. This study involves no human participants, human data, or animals.

Consent for Publication. Not applicable.

Data and Code Availability. The datasets analyzed in this study are publicly available from the sources cited in Section 4.1. The implementation, configuration files, figure-generation scripts, and experiment records are included in the supplementary archive submitted with this manuscript for reviewer access. A public repository with a persistent identifier will be released upon acceptance, and its URL will be added to the final published version.

AI-Assisted Tools. During the preparation of this work, the authors used OpenAI Codex (desktop application, accessed August 2026) to support code editing, experiment orchestration, language editing, document organization, LaTeX formatting, and the initial layout of the explanatory workflow in Fig. 1, which was subsequently rebuilt and edited as native vector objects by the authors. The tool was not used as an image-quality metric or as a source of experimental observations, and no AI tool altered primary research images or generated quantitative plots. After using this tool, the authors reviewed and edited the resulting material and take full responsibility for the content of the article.

References

[1] Cao, Z., Wang, Z., Zhang, Y., Jin, C., Cai, W., Zeng, Z., Liu, J.: Enhancing 3D gaussian splatting for low-quality images: semantically guided training and unsupervised quality assessment. The Visual Computer **41**, 5815–5833 (2025) https://doi.org/10.1007/s00371-024-03754-z

[2] Teuber, M., Kunert, C., Schwandt, T., Broll, W.: Geometry-based light probe placement for realtime lighting in gaussian splatting environments. The Visual Computer **41**, 11783–11795 (2025) https://doi.org/10.1007/s00371-025-04124-z

[3] Levy, D., Peleg, A., Pearl, N., Rosenbaum, D., Akkaynak, D., Korman, S., Treibitz, T.: SeaThru-NeRF: Neural radiance fields in scattering media. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 56–65 (2023)

[4] Akkaynak, D., Treibitz, T.: A revised underwater image formation model. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6723–6732 (2018)

[5] Kerbl, B., Kopanas, G., Leimkuehler, T., Drettakis, G.: 3D gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023)

[6] Yang, D., Leonard, J.J., Girdhar, Y.: SeaSplat: Representing underwater scenes with 3D gaussian splatting and a physically grounded image formation model. In: 2025 IEEE International Conference on Robotics and Automation, pp. 7632–7638 (2025). https://doi.org/10.1109/ICRA55743.2025.11128502

[7] Mualem, N., Amoyal, R., Freifeld, O., Akkaynak, D.: Gaussian Splashing enables direct volumetric rendering underwater. Scientific Reports (2026) https://doi.org/10.1038/s41598-026-56917-4

[8] Li, H., Song, W., Xu, T., Elsig, A., Kulhanek, J.: WaterSplatting: Fast underwater 3D scene reconstruction using gaussian splatting. In: International Conference on 3D Vision, pp. 969–978 (2025)

[9] Yuan, J., Li, Y., Zhang, Y., Guo, C., Tang, X., Wang, R., Li, C.: 3D-UIR: 3D gaussian for underwater 3D scene reconstruction via physics-based appearance-medium decoupling. IEEE Transactions on Image Processing **35**, 5452–5465 (2026) https://doi.org/10.1109/TIP.2026.3694143

[10] Zhang, X., Wang, Y., Fang, S., Wang, Z., Qi, D., Ding, W.: WaterClear-GS: Optical-aware gaussian splatting for underwater reconstruction and restoration. arXiv preprint arXiv:2601.19753 (2026)

[11] Akkaynak, D., Treibitz, T.: Sea-thru: A method for removing water from underwater images. In: Proceedings of the IEEE/CVF Conference on Computer Vision

and Pattern Recognition, pp. 1682–1691 (2019)

[12] Berman, D., Levy, D., Avidan, S., Treibitz, T.: Underwater single image color restoration using haze-lines and a new quantitative dataset. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(8), 2822–2837 (2021) https://doi.org/10.1109/TPAMI.2020.2977624

[13] Wu, J., Wang, T., Siddique, M.A.B., Islam, M.J., Fermuller, C., Aloimonos, Y., Metzler, C.A.: Single-step latent diffusion for underwater image restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025) https://doi.org/10.1109/TPAMI.2025.3599775

[14] Qaisar, Z.H., Khan, R., Mehmood, A., Khan, A., Ibrahim, M.M.: A deep adaptive framework for low-light image enhancement in adverse lighting conditions. The Visual Computer **41**, 12109–12124 (2025) https://doi.org/10.1007/s00371-025-04147-6

[15] Hu, J., Wang, H., Xu, B., Ding, N., Lu, Z., Lei, N., He, Y.: AquaSplatting: A hybrid 3D representation for robust underwater scene reconstruction via dual-branch rendering. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 40, pp. 4843–4850 (2026). https://doi.org/10.1609/aaai.v40i6.42487

[16] Wu, C., Dong, J., Li, C., Tang, J.: Plenodium: Underwater 3D scene reconstruction with plenoptic medium representation. Advances in Neural Information Processing Systems **38** (2025)

[17] Xing, W., Chen, J., Yang, Z., Lin, C., Dong, J., Chen, C., Zhou, X., Han, M.: UW-3DGS: Underwater 3D reconstruction with physics-aware gaussian splatting. arXiv preprint arXiv:2508.06169 (2025)

[18] Wang, H., Anantrasirichai, N., Zhang, F., Bull, D.: UW-GS: Distractor-aware 3D gaussian splatting for enhanced underwater scene reconstruction. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3280–3289 (2025)

[19] Kweon, M., Park, J.: OceanSplat: Object-aware gaussian splatting with trinocular view consistency for underwater scene reconstruction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 40, pp. 5773–5781 (2026). https://doi.org/10.1609/aaai.v40i7.37498

[20] Mel, M., Siddiqui, M., Zanuttigh, P.: End-to-end learning for joint depth and image reconstruction from diffracted rotation. The Visual Computer **40**, 5961–5977 (2024) https://doi.org/10.1007/s00371-023-03147-8

[21] Li, J., Han, G., Wan, J., Gao, Y., Han, D.: DualPhys-GS: Dual physically-guided 3D gaussian splatting for underwater scene reconstruction. Computers & Graphics **133**, 104405 (2025) https://doi.org/10.1016/j.cag.2025.104405

[22] Li, Q., Fu, R.: Novel view synthesis for underwater scenes with gaussian splat fields and physically-based water modeling. Neurocomputing **677**, 133060 (2026) https://doi.org/10.1016/j.neucom.2026.133060

[23] Shi, X., Wu, C., Chu, Z., Huang, Y.: HydroGaussian: Physically-guided hierarchical gaussian splatting for underwater scene reconstruction. Computer Graphics Forum, 70501 (2026) https://doi.org/10.1111/cgf.70501

[24] Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A.: Mip-Splatting: Alias-free 3D gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19447–19456 (2024)

[25] Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2D gaussian splatting for geometrically accurate radiance fields. ACM Transactions on Graphics **43**(4) (2024)

[26] Talegaonkar, C., Belhe, Y., Ramamoorthi, R., Antipa, N.: Volumetrically consistent 3D gaussian rasterization. In: CVPR, pp. 10953–10963 (2025)

[27] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004) https://doi.org/10.1109/TIP.2003.819861

[28] Horé, A., Ziou, D.: Image quality metrics: PSNR vs. SSIM. In: 2010 20th International Conference on Pattern Recognition, pp. 2366–2369 (2010). https://doi.org/10.1109/ICPR.2010.579

[29] Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–595 (2018)

[30] Hu, J., Song, W., Yu, S., Wang, M., Yi, Z., Wu, R., Xiang, M., Lei, N., Wang, S., Luo, Z., He, Y.: Underwater360: Reconstructing underwater scenes from panoramic images with omnidirectional gaussian splatting. arXiv preprint arXiv:2605.26447 (2026)

[31] Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth Anything V2. Advances in Neural Information Processing Systems **37** (2024) https://doi.org/10.52202/079017-0688

[32] Efron, B.: Bootstrap methods: Another look at the jackknife. The Annals of Statistics **7**(1), 1–26 (1979) https://doi.org/10.1214/aos/1176344552

[33] Davis, J., Goadrich, M.: The relationship between precision-recall and ROC curves. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 233–240 (2006). https://doi.org/10.1145/1143844.1143874

[34] Rahman, M.A., Wang, Y.: Optimizing intersection-over-union in deep neural networks for image segmentation. In: Advances in Visual Computing. Lecture Notes in Computer Science, vol. 10072, pp. 234–244 (2016). https://doi.org/10.1007/978-3-319-50835-1_22