AQUA3R: FEED-FORWARD UNDERWATER RECONSTRUCTION WITH DUAL GAUSSIAN SCENES

Anonymous authors
Paper under double-blind review

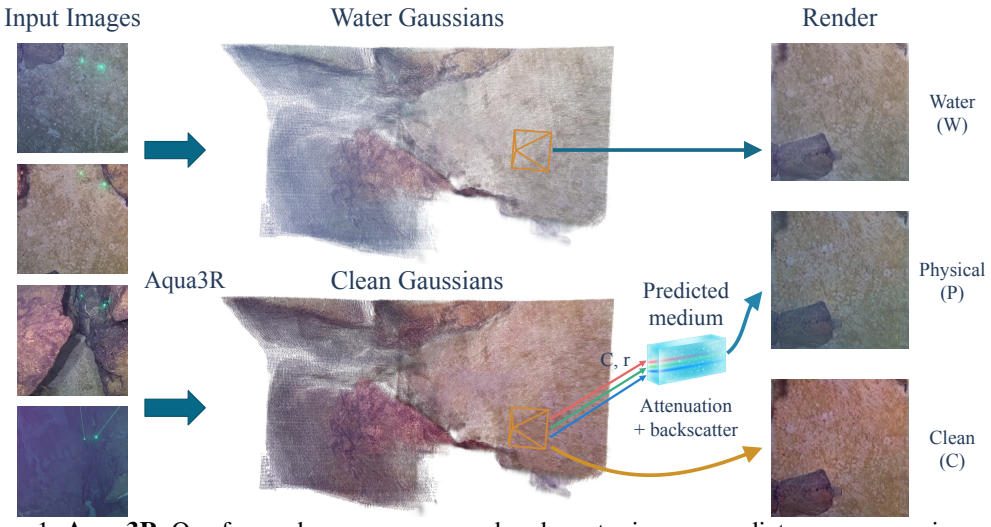

Figure 1: **Aqua3R.** One forward pass over unposed underwater images predicts cameras, an in-water Gaussian scene (render W), a restored Gaussian scene (render C) and a scene-level medium that re-renders C, with its ray range, as the physical re-rendering P.

ABSTRACT

Water attenuation and backscatter couple scene appearance with the surrounding medium, so underwater reconstruction requires both reproducing the observed images and recovering the underlying water-free scene. Per-scene methods can address both goals, but require optimization from calibrated cameras, while existing feed-forward approaches do not explicitly separate in-water and restored scene representations. We present Aqua3R, a pose-free feed-forward model that predicts cameras, in-water and restored Gaussian scenes, and an explicit medium model in a single forward pass. The two scene representations share camera estimates and pre-fusion depth points and diverge only at the attribute heads. The medium model then re-renders the restored scene to reproduce the underwater observation. We further introduce DL3DV-Water, which provides paired clean and underwater views with formation parameters, and DeepSea-ROV, which contains raw deep-sea videos with registered clips. We also construct transfer sets and an evaluation protocol covering all predicted outputs. At 24 views, Aqua3R improves in-water PSNR by 0.98–2.75 dB over the stronger of two underwater-adapted YoNoSplat baselines on real footage and two synthetic transfer sets, while producing substantially more accurate camera estimates on real footage. In a controlled data study, broader medium coverage improves physical re-rendering PSNR on 25 real clips by an average of 1.06 dB. In contrast, gains from increasing the number of source scenes depend on the training design, and additional conditions or optimization updates do not yield consistent improvements. We will release the data and code, along with model checkpoints and per-scene results.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Underwater images entangle scene appearance with wavelength-dependent attenuation and backscatter. Reconstruction must reproduce observed views while recovering the scene's water-free appearance. SeaThru-NeRF, SeaSplat, and WaterSplatting (Levy et al., 2023; Yang et al., 2025; Li et al., 2025a) model water during rendering, but optimise each scene separately from calibrated cameras, which underwater footage rarely provides.

On land, feed-forward methods avoid both costs: DUSt3R and MASt3R (Wang et al., 2024; Leroy et al., 2024) predict point maps from unposed pairs; VGGT (Wang et al., 2025b) predicts cameras and geometry; and AnySplat and YoNoSplat (Jiang et al., 2025a; Ye et al., 2026) predict renderable Gaussians. None models water, so underwater inputs retain its effects. Concurrent underwater methods recover geometry and cameras (WAT3R and Wat3R (Xu et al., 2026; Ren et al., 2026)), add per-scene medium-aware splatting (Swimm3R (Kweon & Sattar, 2026)), or predict Gaussians and medium parameters in one pass (NemoSplat (Guo et al., 2026)). Swimm3R and NemoSplat explain the observation through the formation model without a separate in-water scene.

We propose Aqua3R, which predicts cameras, an in-water Gaussian scene, a restored Gaussian scene, and a medium from unposed underwater images in one pass (Figure 1). A VGGT-Ω aggregator (Wang et al., 2026; 2025b) encodes the images; a single Gaussian head predicts both scenes from shared depth-derived point positions and splits only at the attribute heads. The in-water render W matches the observation, while the restored render C depicts the scene without water. A medium head estimates scene-level attenuation, backscatter, and veiling light θ (Akkaynak & Treibitz, 2018). Re-rendering C through θ gives P, trained against the observation. W, C, and P share cameras and can be compared directly.

Paired clean references are unavailable in real footage. We therefore render DL3DV scenes (Ling et al., 2024) through the formation model to build DL3DV-Water: clean and underwater views paired with medium parameters. We also curate DeepSea-ROV, registered clips from towed deep-sea vehicle footage, split by dive for adaptation and testing. These datasets let us compare the effects of scene count, water-condition coverage, and training budget on W, C, and P.

For in-water rendering, Aqua3R outperforms underwater-adapted AnySplat and YoNoSplat in PSNR and LPIPS on real footage and two synthetic transfer sets, with far more accurate cameras on real footage. In our data study, wider medium coverage consistently raises physical re-rendering PSNR on real footage. More scenes help only under some recipes, and longer training gives no consistent gain.

Our contributions are:

- **Dual-scene feed-forward model.** Aqua3R predicts cameras, in-water and restored Gaussian scenes, and an explicit medium in one pass, so W, C, and P share one camera set.
- **Paired and real underwater data.** DL3DV-Water pairs clean and underwater views with formation parameters; DeepSea-ROV provides registered real deep-sea clips; a single pose-free protocol evaluates both.
- **Controlled study of underwater training data.** We isolate the effects of scene count, medium coverage, and training budget on W, C, and P.

2 RELATED WORK

Underwater imaging and per-scene reconstruction. The revised underwater image formation model (Akkaynak & Treibitz, 2018) separates attenuation from range-dependent backscatter, and Sea-thru (Akkaynak & Treibitz, 2019) inverts it given a range map. Image-based restoration and enhancement methods (Li et al., 2020; Ancuti et al., 2018) correct colour directly, but do not recover geometry for re-rendering. To obtain both, SeaThru-NeRF (Levy et al., 2023) incorporates the formation model into volume rendering, while WaterSplatting and SeaSplat (Li et al., 2025a; Yang et al., 2025) incorporate it into Gaussian splatting, alongside other per-scene variants (Mualem et al., 2026; Liu et al., 2024; 2025; Yuan et al., 2026; Li et al., 2025b; Xing et al., 2025; Zhang et al., 2026; Huang et al., 2025; Hu et al., 2026; Kweon & Park, 2026; Jiang et al., 2025b; Wang et al., 2025a). All of these optimise one scene at a time from estimated cameras. We instead build the formation model into a feed-forward predictor: it synthesises paired training data and re-renders the restored scene through the predicted medium.

Feed-forward reconstruction. Feed-forward Gaussian prediction began with posed image pairs (Charatan et al., 2024) and became pose-free with NoPoSplat (Ye et al., 2025). DUSt3R and its successors (Wang et al., 2024; Leroy et al., 2024; Wang et al., 2025b) regress point maps from unposed images, with VGGT also predicting cameras. AnySplat (Jiang et al., 2025a) adds a rendering-supervised Gaussian head to such a model, but none of these methods models water. Underwater, WAT3R and Wat3R (Xu et al., 2026; Ren et al., 2026) predict geometry and cameras, while Swimm3R (Kweon & Sattar, 2026) couples medium-aware geometry with splatting. NemoSplat (Guo et al., 2026) predicts cameras, Gaussians, and medium parameters from video. Feed-forward priors also seed sparse-view splatting (Caldeira et al., 2026), and medium-aware video diffusion has been explored (Huang et al., 2026). Unlike these methods, Aqua3R predicts in-water and restored Gaussian scenes under one camera set. A predicted medium links the two: re-rendering the restored scene through it must explain the observation.

3 METHOD

We introduce Aqua3R, a feed-forward model that reconstructs an underwater scene from unposed images and restores it without the water in one pass, under one set of predicted cameras (Figure 2).

Problem formulation. Given V unposed underwater images $\{\mathbf{I}_i\}_{i=1}^{V}$ of one scene, we learn a feed-forward network that maps them to per-view cameras and depth, two 3D Gaussian scenes and the water medium:

$$f_\phi : \{\mathbf{I}_i\}_{i=1}^{V} \;\mapsto\; \big(\{\mathbf{E}_i, \mathbf{K}_i, z_i\}_{i=1}^{V},\ \mathcal{G}^w,\ \mathcal{G}^c,\ \theta\big), \qquad \theta = (\boldsymbol{\beta}_a, \boldsymbol{\beta}_b, \mathbf{B}_\infty) \in \mathbb{R}^9, \tag{1}$$

where $\mathbf{E}_i = [\mathbf{R}_i \,|\, \mathbf{t}_i]$, $\mathbf{K}_i$ and z_i are the world-to-camera extrinsics, intrinsics and depth map of view i. The *in-water scene* $\mathcal{G}^w$ and the *restored scene* $\mathcal{G}^c$ are built from pixel-aligned Gaussians $(\mu_j, \alpha_j^b, \mathbf{q}_j^b, \mathbf{s}_j^b, \mathbf{c}_j^b)$, $b \in \{w, c\}$: the centres μ_j are shared before fusion, while opacity, rotation, scale and colour are predicted per scene. The medium θ holds per-channel attenuation $\boldsymbol{\beta}_a$, backscatter $\boldsymbol{\beta}_b$ and veiling light $\mathbf{B}_\infty$, constant over the scene. At a query camera, splatting gives the in-water render W from $\mathcal{G}^w$ and the restored render C from $\mathcal{G}^c$, and passing C through θ gives the physical re-rendering P (Section 3.3).

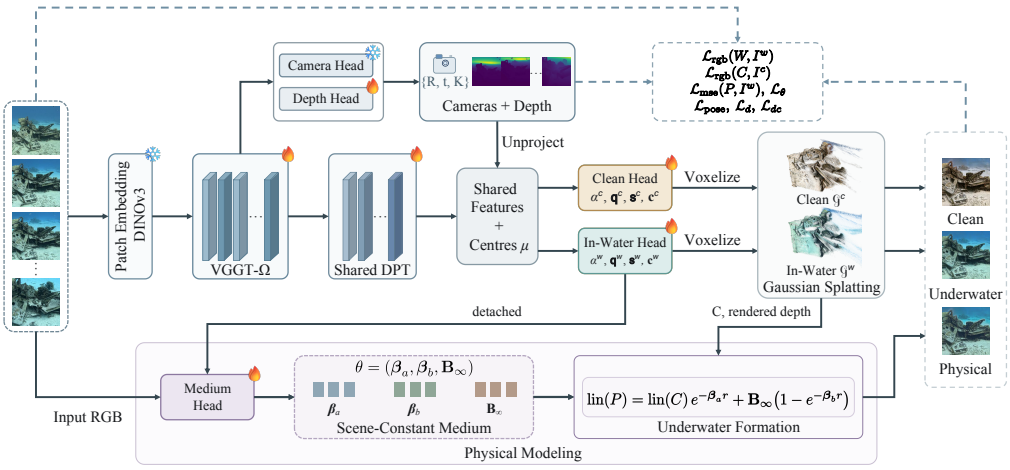

Figure 2: **Overview of Aqua3R.** The VGGT-Ω aggregator turns unposed underwater frames into cameras and depth, whose unprojected points are the pre-fusion support shared by the in-water and restored Gaussian branches. The medium head reads the detached in-water branch output and the input images and supplies θ, with which Equation (3) re-renders C into P; the physical loss on P is stopped at C and its depth. Snowflakes mark frozen modules, flames trained ones.

3.1 ANALYSIS OF THE OUTPUT SPACE AND TRAINING STRATEGY

Output space: one restored scene, two models, or two scenes. A feed-forward model can provide observation-faithful reconstruction and restoration in three ways. *(i) One restored scene and a medium,*

as in per-scene underwater methods (Levy et al., 2023; Li et al., 2025a), explains observations only through the formation model, so effects not captured by a global medium, such as lights moving with the vehicle, can bias restoration. *(ii) Two single-output models*, trained on underwater and clean targets respectively, can predict inconsistent cameras, as the adapted YoNoSplat models do (see Appendix C.3). We choose *(iii) two scenes on one geometric support*: W fits observations directly, C learns from clean references, and both share one camera set. Re-rendering C through the medium gives a separate physical training signal P; W is never formed from P.

Training strategy: blocking the physical shortcut. Physical re-rendering should tie C to observations, but it does not uniquely determine the restoration: an underwater-looking C with a nearly clear medium can also explain the observation. In a real-clip adaptation pilot scored on Real25 without clean references, letting the physical loss reach C and its depth improved physical consistency but increased drift from the unadapted restoration and turned the deep-sea seafloor pink (see Appendix D.1). We therefore detach C and its rendered depth on the physical path, blocking direct recolouring or reshaping by the physical loss. This loss trains the medium and still reaches the shared aggregator indirectly through the camera intrinsics; C retains its clean-image and geometric supervision.

3.2 Model Architecture

Backbone network. The patch embedding and the alternating frame and global attention aggregator of VGGT-Ω (Wang et al., 2026; 2025b) process all views jointly. A camera head predicts $\mathbf{E}_i$ and $\mathbf{K}_i$, and a DPT depth head predicts z_i, which places one Gaussian centre per pixel u at $\mu_i(u) = \mathbf{R}_i^\top \big(z_i(u)\mathbf{K}_i^{-1}\tilde{u} - \mathbf{t}_i\big)$, with $\tilde{u}$ the homogeneous pixel coordinate. Both scenes share these centres.

Dual Gaussian head. Water changes how a surface looks rather than where it is, so we share every module that places a Gaussian and split only at the layers that predict its attributes. Following AnySplat (Jiang et al., 2025a), a feature adapter maps the aggregator tokens $\mathbf{t}$ to four DPT levels, which a shared refinement trunk F_d fuses. Each branch $b \in \{w, c\}$ then predicts raw Gaussian attributes $\boldsymbol{\rho}^b$ (opacity, rotation, scale and colour) and a fusion confidence κ^b,

$$\big(\boldsymbol{\rho}^b, \kappa^b\big) = F^b_{\text{out}}\big(\uparrow F^b_{\text{mid}}(F_d(\mathbf{t})) + F^b_{\text{rgb}}(\mathbf{I})\big), \tag{2}$$

where $\uparrow$ upsamples to full resolution and F^b_{rgb} merges the input image to restore detail lost to patchification. Both branches read the same underwater images, and the restored branch is initialised as a copy of the in-water branch, so restoration is learned as a departure from the observation. The complete model has 1.195B parameters, against 1.930B for the adapted YoNoSplat W and C pair (about 0.97B each). Each branch then applies AnySplat's differentiable voxel fusion with its own confidence, so the fused centres of the two scenes may differ although they start from the same support. Each scene is rendered with gsplat (Ye et al., 2024) over a background colour: the mean context colour for W, and that colour plus a bounded learned residual for C.

Medium head. A small head estimates the medium from the raw attributes of the in-water branch and the input images, both detached and average-pooled by four. After a 1×1 and a 3×3 convolution, features pooled over pixels and views give $\boldsymbol{\beta}_a$ and $\boldsymbol{\beta}_b$ through an MLP with sigmoid bounds, and features pooled per view give a veiling light that is averaged over views to form $\mathbf{B}_\infty$. Detaching the inputs keeps the medium losses out of the Gaussian features along this path. Appendix A.1 details both heads and their gradient paths.

3.3 Physical Re-rendering and Range Gauge

Physical re-rendering. Following the revised underwater image formation model (Akkaynak & Treibitz, 2018), P is formed from C in linear light,

$$\text{lin}(P) = \text{lin}(C)\, e^{-\boldsymbol{\beta}_a r} + \mathbf{B}_\infty\big(1 - e^{-\boldsymbol{\beta}_b r}\big), \tag{3}$$

where $\text{lin}(\cdot)$ converts sRGB to linear light and r is the camera-ray range of each pixel, computed from the z-depth rendered with C and the rendering intrinsics; P is converted back to sRGB for supervision. Swapping θ at fixed C and r probes medium effects (Appendix D.4).

Range gauge. Feed-forward geometry is defined only up to scale, and Equation (3) sees the coefficients only through the optical depths $\boldsymbol{\beta}_a r$ and $\boldsymbol{\beta}_b r$: multiplying every range by s and dividing $\boldsymbol{\beta}$ by s leaves P unchanged, so a predicted medium is undefined until the range scale is fixed. We

fix it by convention. One detached scale per scene maps the median valid context ray range to $4\,\mathrm{m}$ and is applied to all rendered views; pixels whose accumulated alpha in C is below 0.5 first take the 95th-percentile valid depth of their view. The synthetic labels are generated in the same gauge (Section 4.1), so predicted coefficients are comparable across scenes as effective coefficients of this convention, not as metric water properties. As argued in Section 3.1, C and its depth are detached before Equation (3); the physical loss reaches θ and, through the intrinsics in r, the aggregator.

Optical-depth augmentation. The same invariance widens the training media without new scenes. At the fixed gauge, rendering a source with its ranges scaled by s equals rendering it with coefficients $s\boldsymbol{\beta}$ and the same $\mathbf{B}_\infty$, so we re-render every synthetic variant twice with scales drawn log-uniformly from $[0.15, 0.8]$ and $[1.3, 2.6]$ and label the copies with $s\boldsymbol{\beta}$. Clean references and cameras are unchanged, and the copies form the second training stage (Section 3.4).

3.4 Model Training

Each update samples a context count $n \in \{2, \ldots, 24\}$ with probability proportional to n^2 and two target views per scene, and both scenes are rendered at all of them. Clean references and medium labels come from paired synthetic data (Section 4.1). The objective is

$$\begin{aligned}\mathcal{L} = \lambda_w \mathcal{L}_{\mathrm{rgb}}(W, I^w) + \lambda_c \mathcal{L}_{\mathrm{rgb}}(C, I^c) + \lambda_p \mathcal{L}_{\mathrm{mse}}(P, I^w) + \mathcal{L}_\theta \\ + \lambda_{\mathrm{pose}} \mathcal{L}_{\mathrm{pose}} + \lambda_d \mathcal{L}_d + \lambda_{dc} \mathcal{L}_{dc}.\end{aligned} \tag{4}$$

Image, physical and medium losses. W is compared with the underwater images I^w and C with the paired clean images I^c at the same cameras, where $\mathcal{L}_{\mathrm{rgb}} = \mathcal{L}_{\mathrm{mse}} + \lambda_{lp}\mathcal{L}_{\mathrm{lpips}} + \lambda_g \mathcal{L}_{\mathrm{grad}}$ averages the context-view and target-view means, as does the physical MSE. P, formed from the detached C, is scored against I^w by MSE alone, so it trains the medium without a perceptual term. $\mathcal{L}_\theta$ regresses θ to the synthesis labels with a Smooth-L1 loss on $\log \boldsymbol{\beta}_a$ and $\log \boldsymbol{\beta}_b$ and an MSE on $\mathbf{B}_\infty$.

Distillation losses. As in AnySplat, a frozen copy of the VGGT-Ω initialisation supervises the geometry. $\mathcal{L}_{\mathrm{pose}}$ regresses the camera encodings to the teacher's with an iterative Huber loss, its field of view replaced by that of the cropped intrinsics; $\mathcal{L}_d$ regresses the depth rendered from C to the teacher's depth where the teacher's confidence exceeds its per-view 30th percentile; and $\mathcal{L}_{dc}$ ties this rendered depth to the depth-head prediction. The patch embedding and the camera head stay frozen.

Schedule and real footage. The full model trains for 4000 base updates on the original DL3DV-Water conditions and 1500 updates on the pool extended by the optical-depth copies; a *one-stage model* stops after the base stage. Loss weights are listed in Appendix A. Real clips, when added (Section 4.4), have no clean reference or medium label and keep only the in-water, physical and distillation losses, so the physical loss is the only signal the medium receives from real water.

4 Experiments

4.1 Experimental Setup

Baselines. *Feed-forward:* AnySplat (Jiang et al., 2025a) and YoNoSplat (Ye et al., 2026), each adapted on DL3DV-Water separately for W and C. YoNoSplat-R and YoNoSplat-D start from RE10K and DL3DV checkpoints, respectively; AnySplat (rect.) keeps rectangular training. YoNoSplat-D and Aqua3R start from checkpoints whose upstream data may include Test32 sources. *Per-scene:* WaterSplatting (Li et al., 2025a), SeaSplat (Yang et al., 2025), 3D-UIR (Yuan et al., 2026) without its distance-guided regularisation optimisation (DRO), Gaussian Splashing (Mualem et al., 2026) and 3DGS, from the cameras and points of our frozen VGGT-Ω initialisation. *Single-image:* FUnIE-GAN (Islam et al., 2020), Fusion (Ancuti et al., 2018), U-shape (Peng et al., 2023) and Sea-thru (Akkaynak & Treibitz, 2019) with our predicted range, used only for restoration. *Camera pose:* the released Wat3R of Ren et al. (2026), which does not render. Swimm3R, NemoSplat and WAT3R (Kweon & Sattar, 2026; Guo et al., 2026; Xu et al., 2026) have no released renderable model. Appendix B.2 gives the protocols.

Implementation and evaluation. We report PSNR, SSIM and LPIPS for rendering, and RRA@5°, RTA@5° and AUC@30° for cameras. A difference is *resolved* (*) when its paired 95% bootstrap interval excludes zero; bold and underline mark the best and second-best entries. Training and evaluation details are in Appendices A, B.3 and C.3.

Datasets. Dataset construction and splits are detailed in Appendix B. DL3DV-Water comprises 1,996 DL3DV sources (Ling et al., 2024) of 64 views each, rendered through Equation (3) in one oceanic and one coastal Jerlov water with same-camera clean references and medium labels. Pseudo-ranges are normalised to a 4 m median context range by convention (Section 3.3). Sources split into 1,928 training, 32 validation, 32 held-out test (Test32, in domain) and four development sources. Transfer sets use the same synthesis: Mip360-Water (seven Mip-NeRF 360 scenes (Barron et al., 2022)) and OOD-Water (eleven sources fixed before evaluation). DeepSea-ROV provides 34 registered clips from 22 deep-sea recordings, split by dive into 26 adaptation and eight test clips. Real25 combines these eight with four SeaThru-NeRF scenes (Levy et al., 2023) and 13 Water3D clips (Ren et al., 2026), without clean references. Figure 3 shows synthetic and real examples.

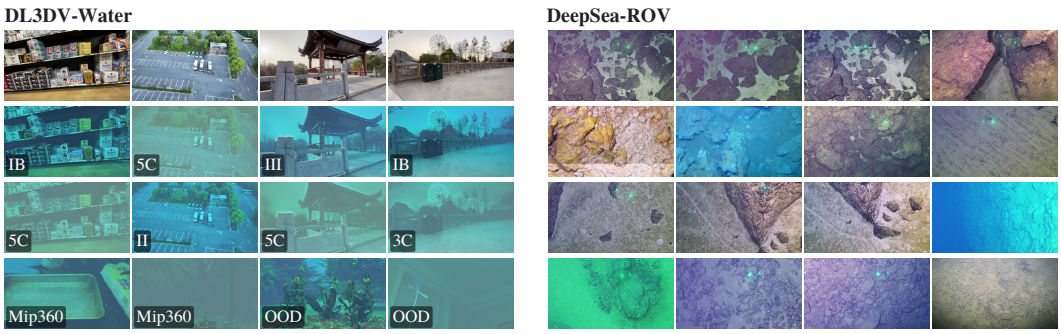

Figure 3: **Paired synthetic data and real footage.** Left: DL3DV-Water clean views (top row), their underwater renders under two Jerlov classes at the same cameras (tags), and the transfer sets (bottom row). Right: DeepSea-ROV frames, which have no clean reference.

4.2 Main Results

In-water novel view synthesis. As shown in Table 1a, Aqua3R renders the observed water better than the adapted baselines in PSNR and LPIPS on Real25 and the transfer sets. It leads at every view count, with one network for all three outputs. Another seed preserves the 24-view PSNR leads (see Appendix C.2). YoNoSplat, with one model per output, remains stronger on the in-domain Test32; in our architecture, dual-output sharing can reduce Test32 W fit, as without the physical and medium losses a single-output version gains 0.30 dB there (Appendix D.2).

Table 1: **Feed-forward novel-view synthesis from 8 to 24 context views. (a)** W and P against the underwater observation; **(b)** C against the clean reference. Transfer averages the 18 Mip360-Water and OOD-Water sources with equal weight. Real25 clips have $L = 11$ to 21 context frames and use $\min(16, L)$ of them in the 16-view column and all L in the 24-view column.

(a) In water: W and P, scored against the observation

Method	8 views			16 views			24 views		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Real: Real25 (25 clips)									
AnySplat (rect.)	18.15	0.379	0.586	18.44	0.385	0.589	18.51	0.390	0.596
YoNoSplat-R	18.39	**0.416**	0.579	18.63	**0.415**	0.571	18.55	**0.415**	0.569
YoNoSplat-D	18.46	0.412	0.584	18.49	0.411	0.577	18.48	0.412	0.573
Ours (W)	**19.34**	0.397	**0.506**	**19.58**	0.402	**0.497**	**19.53**	0.402	**0.493**
Ours (P)	18.20	0.388	0.548	18.37	0.394	0.535	18.34	0.395	0.534
Synthetic, transfer: Mip360-Water and OOD-Water (18 sources)									
AnySplat (rect.)	16.27	0.549	0.584	17.34	0.572	0.568	18.01	0.582	0.567
YoNoSplat-R	20.13	0.587	0.507	20.37	0.606	0.500	20.33	0.612	0.502
YoNoSplat-D	20.98	0.628	0.507	20.74	0.631	0.506	20.60	0.632	0.509
Ours (W)	**22.00**	0.630	**0.447**	**22.66**	0.649	**0.418**	**22.79**	0.646	**0.405**
Ours (P)	21.56	**0.644**	0.463	22.09	**0.662**	0.443	22.47	**0.662**	0.426
Synthetic, in domain: Test32 (32 sources)									
AnySplat (rect.)	17.83	0.536	0.523	18.09	0.551	0.524	18.12	0.558	0.524
YoNoSplat-R	25.48	0.715	0.225	25.80	0.724	0.212	26.04	0.736	0.203
YoNoSplat-D	**25.88**	**0.725**	**0.209**	**26.21**	**0.737**	**0.200**	**26.66**	**0.752**	**0.186**
Ours (W)	25.04	0.704	0.245	25.47	0.719	0.232	25.76	0.728	0.222
Ours (P)	24.07	0.711	0.280	24.48	0.726	0.267	24.72	0.736	0.260

Restored novel view synthesis. As shown in Table 1b, C gives the best pooled transfer PSNR and LPIPS (per-set results in Appendix C.1) and, like W, trails on Test32. Its physical re-rendering P has the best transfer SSIM in Table 1a, so the restored scene and the predicted medium are consistent with the observation. As shown in Figure 4a, the baselines blur or flatten scenes outside their training domain, whereas W keeps the scene layout. Figure 4b shows that the restored views of the baselines stay blurry, whereas C recovers the clean colours of transfer scenes.

Table 1: **(continued) (b)** C against the clean reference.
(b) Restored: C, against the clean reference

Method	8 views			16 views			24 views		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Synthetic, transfer: Mip360-Water and OOD-Water (18 sources)									
AnySplat (rect.)	10.96	0.234	0.701	11.63	0.247	0.691	11.78	0.249	0.692
YoNoSplat-R	13.20	0.270	0.639	13.26	0.268	0.636	13.65	0.276	0.626
YoNoSplat-D	13.30	**0.278**	0.643	13.26	0.273	0.643	13.32	0.274	0.638
Ours (C)	**13.97**	0.277	**0.616**	**14.53**	**0.308**	**0.585**	**14.91**	**0.308**	**0.563**
Synthetic, in domain: Test32 (32 sources)									
AnySplat (rect.)	12.59	0.254	0.622	12.74	0.262	0.622	12.69	0.267	0.624
YoNoSplat-R	18.28	0.496	0.321	18.56	0.508	0.307	19.11	0.540	0.290
YoNoSplat-D	**18.98**	**0.526**	**0.286**	**19.09**	**0.534**	**0.279**	**19.49**	**0.558**	**0.266**
Ours (C)	17.65	0.478	0.366	18.00	0.503	0.349	18.25	0.520	0.339

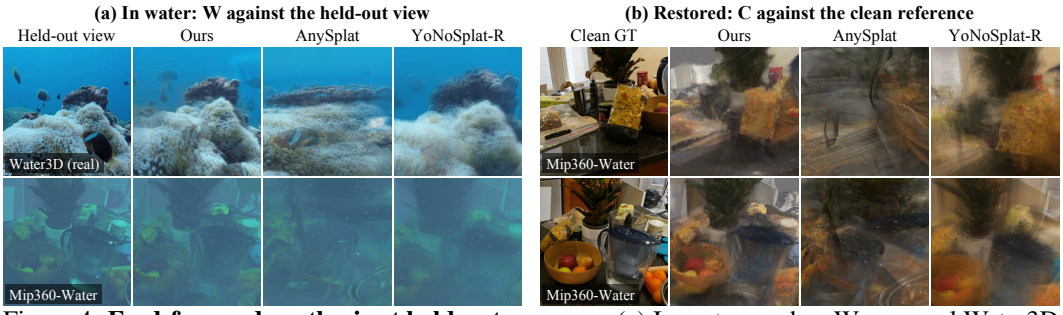

Figure 4: **Feed-forward synthesis at held-out cameras.** (a) In-water renders W on a real Water3D clip and on Mip360-Water; (b) restored renders C on Mip360-Water with the clean reference. YoNoSplat-R is the RE10K-initialised variant.

Camera pose estimation. W, C and P share one camera set, so pose accuracy matters for both scenes. As shown in Table 2, the adapted YoNoSplat models estimate cameras poorly on real footage, whereas ours are the most accurate on Real25. Ours also improve on the VGGT-Ω initialisation, a gain of the full adaptation recipe rather than of the dual outputs alone. At its default resolution, Wat3R leads only on its SeaThru-NeRF benchmark but falls far behind on deep-sea clips (Appendix C.3).

Table 2: **Relative context-pose accuracy on Real25 (%, clip-balanced).** Clip counts in brackets. All inputs are 224×224 except the separately reported 518^2 Wat3R row.

Model	RRA@5°	RTA@5°	AUC@30°	AUC@30° by collection		
				SeaThru (4)	Water3D (13)	DeepSea (8)
AnySplat (rect., W)	66.4	13.7	47.6	69.1	61.7	14.0
YoNoSplat-R (W)	47.3	5.6	36.2	37.5	48.0	16.3
YoNoSplat-D (W)	50.2	8.5	38.0	41.0	50.1	16.8
Wat3R (geometry only)	63.1	9.1	43.8	63.6	56.5	13.3
Wat3R (geometry only, 518^2)	69.5	34.0	54.5	**85.6**	70.1	13.5
VGGT-Ω init. (unadapted)	73.8	34.9	59.8	79.0	71.1	32.0
Ours	**74.9**	**41.3**	**64.6**	78.4	**78.3**	**35.5**

Restoration against dedicated methods. Only synthetic data provide a clean target, so we compare on ten Test32 sources with 56 context views. As shown in Table 3, per-scene methods trained without

clean references restore colour worse than the untouched observation, which scores 12.96 dB against the clean reference on the same frames. Single-image methods cannot render new views, so they must dewater our W. Table 4 shows that they fall far behind C, which predicts the restored scene directly, and a separate 24-view evaluation on Mip360-Water agrees.

Table 3: **Restoration against per-scene methods** on ten Test32 sources with 56 context views.

Method	PSNR↑	SSIM↑	LPIPS↓
WaterSplatting	10.47	0.198	0.702
SeaSplat	10.40	0.189	0.691
3D-UIR (DRO off)	10.07	0.167	0.656
Gaussian Splashing	9.78	0.144	0.748
Ours (C)	**18.73**	**0.563**	**0.321**

Table 4: **Restoration against single-image methods** applied to our W on these frames.

Method	PSNR↑	SSIM↑	LPIPS↓
Fusion	14.50	0.442	0.498
FUnIE-GAN	14.42	0.445	0.502
U-shape	14.16	0.502	0.507
Sea-thru	11.33	0.397	0.506
Ours (C)	**18.73**	**0.563**	**0.321**

Given the scored frame itself, single-image methods still trail C in Test32 PSNR but retain higher SSIM, and U-shape leads on Mip360-Water (see Appendix C.5). Figure 5 shows held-out synthetic restoration and real-frame restoration; without a clean reference, the real colours cannot be ranked.

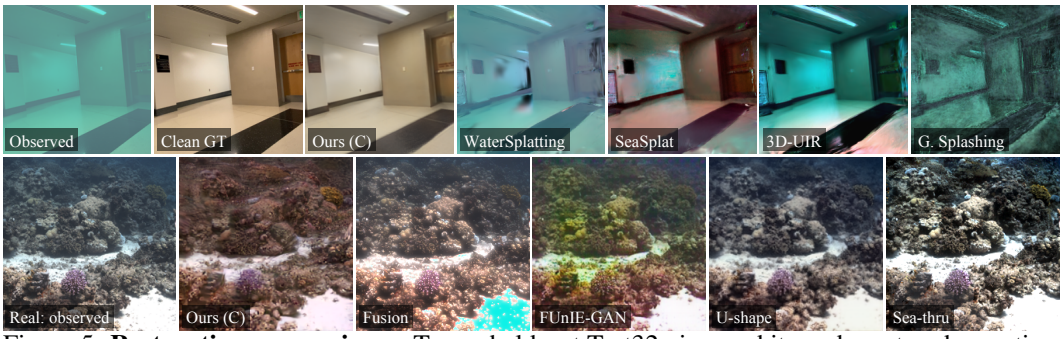

Figure 5: **Restoration comparisons.** Top: a held-out Test32 view and its underwater observation. Bottom: a real frame, where the single-image methods see the observed frame; without a clean reference its colours cannot be ranked.

Comparison with per-scene optimisation. Per-scene optimisers take minutes to hours per scene. Started from the same predicted cameras and points, as in the dense-view protocol of AnySplat, they misplace texture. As shown in Table 5, Aqua3R reconstructs each sequence in 1.0–2.2 s with the best PSNR and SSIM from 32 to 64 views. WaterSplatting keeps sharper texture and the lowest LPIPS. With structure-from-motion cameras, points and reference intrinsics, 3DGS and WaterSplatting exceed our 64-view PSNR. Appendix C.4 gives these runs and examples.

Table 5: **Dense-view reconstruction on five DeepSea-ROV sequences.** Eight held-out views at 224×448. As in AnySplat, per-scene optimisers start from feed-forward cameras and points (here from frozen VGGT-Ω) and run 15000 steps; time is per-scene wall clock on shared A800 GPUs.

Method	32 views			48 views			64 views			Time
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	s↓
3DGS	16.49	0.150	0.597	16.46	0.159	0.605	16.82	0.170	0.606	164–237
SeaSplat	16.77	0.169	0.595	16.71	0.178	0.602	17.38	0.197	0.601	3686–5230
WaterSplatting	18.08	0.182	**0.574**	18.29	0.192	**0.578**	18.61	0.205	**0.575**	346–355
Ours	**19.25**	**0.272**	0.607	**19.22**	**0.274**	0.616	**19.26**	**0.277**	0.617	**1.0–2.2**

4.3 Ablation Studies

Dual versus single output. With the physical and medium-label losses off, single-output controls stay within 0.35 dB of the dual model in PSNR on every set, and only their Test32 W PSNR gain is resolved. Unresolved differences do not establish equivalence, and the dual model renders both scenes from one camera prediction (see Appendix D.2).

Contribution of each loss. Table 6 removes one loss group at a time. In PSNR, removing physical and medium-label losses lowers Real25 P by 1.15 dB with no resolved change in W or C, removing the clean loss lowers Test32 C by 3.95 dB, and removing the underwater loss lowers Test32 W by 1.17 dB. The only resolved PSNR gain, 0.44 dB on Test32 W without the clean loss, is water fit traded for restoration. Appendices D.3 and D.4 vary λ_p and evaluate the medium.

Table 6: **Loss-group ablation.** W, C and P, θ: underwater image, clean image, and physical plus medium-label losses (✓: used, –: removed). The first row is the full model; *: resolved against it.

Losses			Test32 C			Test32 W			Real25 W			Real25 P		
W	C	P, θ	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
✓	✓	✓	18.25	0.520	0.339	25.76	0.728	0.222	19.53	0.402	0.493	**18.34**	0.395	**0.534**
✓	✓	–	18.23	0.519	0.340	25.86	0.727	0.223	19.70	0.404	0.493	17.19*	0.377*	0.570*
✓	–	✓	14.29*	0.419*	0.571*	**26.20***	**0.742***	**0.211***	**19.72**	**0.410***	**0.492**	17.59*	**0.407***	0.600*
–	✓	✓	**18.40**	**0.530**	**0.337**	24.59*	0.716	0.244*	19.03*	0.398	0.508*	18.21	0.395	0.539

4.4 Effects of Scene Count, Medium Coverage and Training Budget

With architecture fixed, we vary source count under the full two-stage recipe and a one-stage recipe with one original water per source. Medium coverage expands from the original water to one or all six water-type and range-scale conditions per source. Appendix E gives the designs and all values, which Figure 6 summarises.

As shown in Figure 6a, more scenes help real and OOD-Water W with one-stage training, while under the full recipe only Test32 gains are resolved. Figure 6b shows that wider medium coverage reliably raises physical-output PSNR on real footage at both source counts, as it does on a new dive held out under a plan fixed before training (see Appendix E.2). It slightly lowers Test32 C and roughly doubles attenuation and backscatter errors (see Appendix D.4); all-condition training also hurts Real25 W at 1928 sources. In Figure 6c, four times more updates lowers Real25 and OOD-Water W. Figure 6d shows that, relative to a continuation without real clips, a small amount of registered real footage helps Real25 W and P, whereas unregistered pseudo-posed segments hurt P. The real fraction grows with clip count, so the effect of clip count itself is not isolated.

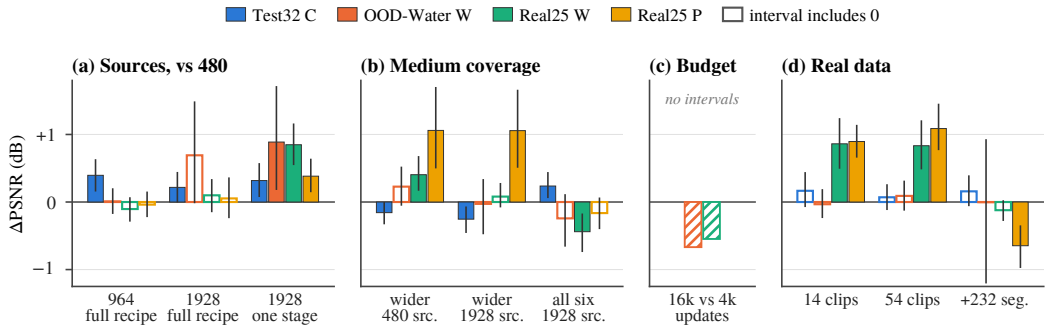

Figure 6: **Effects of training data and budget.** PSNR differences with paired 95% source-bootstrap intervals; hollow bars include zero. (a, b) Two-seed means; in (b), wider minus original coverage and all six minus wider. (c) Seed means without intervals; C and P not evaluated for this contrast. (d) One seed; clips minus a continuation without them, and +232 seg. minus 54 clips.

5 Conclusion

We presented Aqua3R, which reconstructs in-water and restored scenes and the water between them from unposed underwater images in one pass, with one camera set and an explicit medium tying restoration to the observation. It generalises better than underwater-adapted feed-forward baselines to unseen scenes and real footage and recovers more accurate cameras. Wider medium coverage consistently raises physical re-rendering PSNR on real footage, whereas more scenes help only under some recipes.

AI USE STATEMENT

In this work, we used generative AI tools to help formulate the range-gauge invariance of Section 3.3, to supply key steps of its short derivation and to write that derivation, to propose and refine hypotheses, to design and critique experiments, to implement the method and the data-processing and evaluation code, to clean and reformat datasets, to interpret results, and to translate text written by the authors into English. We have not used generative AI tools to develop the conceptual framework, which the authors designed, or to generate synthetic data, which our image formation model renders from real DL3DV frames; qualitative or thematic data analysis is not applicable to this work. Additionally, we used generative AI tools to run and monitor the recorded training and evaluation jobs, to draft and edit parts of the paper and suggest its structure, to create figures and tables from recorded results, to search for, identify and summarise related literature, and to format and check references. We have reviewed all AI-assisted work: the authors checked the reported numbers, and every reference was verified against its arXiv, DOI or publisher record. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

DL3DV-Water is synthesised from DL3DV-10K, whose licence combines CC BY-NC 4.0 with terms restricting onward sharing.[1] We will release generation code, scene lists, medium parameters, range maps and manifests so users can regenerate the paired variants from DL3DV frames obtained under those terms. SeaThru-NeRF, Water3D and the Sea-thru data are used under their respective licences. The 20 coral-reef clips of Appendices D.4 and E.2 come from the public *sweet-corals* survey of Indonesian reef plots (GoPro stereo rigs, CC BY 4.0; https://huggingface.co/datasets/wildflow/sweet-corals) and are used for evaluation only. DeepSea-ROV consists of seafloor footage recorded by a towed deep-sea vehicle during scientific cruises in the South Pacific (Appendix B.1); it shows no people, and the provider withholds the exact positions and dates. We release the raw videos, curated registered scenes, splits and permitted acquisition metadata for non-commercial research under the accompanying dataset licence. The released checkpoints are derivatives of the public VGGT-Ω weights and inherit their licence. Restored colours are model predictions and can differ from the unknown true appearance; scientific interpretation of biological or material properties should retain the original observations alongside them.

REPRODUCIBILITY STATEMENT

We will release code, checkpoints, the DL3DV-Water generation pipeline (scene lists, medium parameters, range maps and manifests), the raw and curated DeepSea-ROV data, and the evaluation scripts. Architecture and training details are given in Appendix A; benchmark construction, real-data splits and the evaluation protocol in Appendix B; the design of the scene-count, coverage and budget study, including seeds, budgets and the source-bootstrap procedure, in Appendix E; and full comparison and training-seed records in Appendix C. Per-scene metrics for every table row will be released with the code, each linked to its checkpoint, configuration and input manifest.

REFERENCES

Derya Akkaynak and Tali Treibitz. A revised underwater image formation model. In *CVPR*, 2018.

Derya Akkaynak and Tali Treibitz. Sea-thru: A method for removing water from underwater images. In *CVPR*, 2019.

Codruta O. Ancuti, Cosmin Ancuti, Christophe De Vleeschouwer, and Philippe Bekaert. Color balance and fusion for underwater image enhancement. *IEEE Transactions on Image Processing*, 27(1):379–393, 2018.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022.

[1] https://github.com/DL3DV-10K/Dataset/blob/main/License.md

Harvey Caldeira, Haoran Wang, Guoxi Huang, Shaoyu Cai, Rachel Fu, and Nantheera Anantrasirichai. Geometry beneath the waves: Dense priors for sparse-view underwater 3D Gaussian splatting. In *SIGGRAPH Asia Posters*, 2026. arXiv:2609.18737.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D Gaussian splats from image pairs for scalable generalizable 3D reconstruction. In *CVPR*, 2024.

Xiaopeng Guo, Wai Chung Tse, Yipeng Zhu, Hanwen Zhang, Huajian Huang, and Sai-Kit Yeung. NemoSplat: Feed-forward 4D Gaussian splatting for media-aware underwater reconstruction. *arXiv preprint arXiv:2608.22888*, 2026.

Jiangbei Hu, Weichao Song, Shibo Yu, Mohan Wang, Zihan Yi, Rui Wu, Mingkang Xiang, Na Lei, Shengfa Wang, Zhongxuan Luo, and Ying He. Underwater360: Reconstructing underwater scenes from panoramic images with omnidirectional Gaussian splatting. *arXiv preprint arXiv:2605.26447*, 2026.

Guoxi Huang, Haoran Wang, Zipeng Qi, Wenjun Lu, David Bull, and Nantheera Anantrasirichai. From restoration to reconstruction: Rethinking 3D Gaussian splatting for underwater scenes. *arXiv preprint arXiv:2509.17789*, 2025.

Yuqiang Huang, Yuxi Wang, Junyu Dong, and Zhaoxiang Zhang. Ocean4D: Generative underwater 4D reconstruction via medium-aware video diffusion. *arXiv preprint arXiv:2606.23298*, 2026.

Md Jahidul Islam, Youya Xia, and Junaed Sattar. Fast underwater image enhancement for improved visual perception. *IEEE Robotics and Automation Letters*, 5(2):3227–3234, 2020.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, Dahua Lin, and Bo Dai. AnySplat: Feed-forward 3D Gaussian splatting from unconstrained views. *ACM Transactions on Graphics*, 44(6):1–16, 2025a.

Zhuodong Jiang, Haoran Wang, Guoxi Huang, Brett Seymour, and Nantheera Anantrasirichai. RUSplatting: Robust 3D Gaussian splatting for sparse-view underwater scene reconstruction. In *BMVC*, 2025b. arXiv:2505.15737.

Minseong Kweon and Jinsun Park. OceanSplat: Object-aware Gaussian splatting with trinocular view consistency for underwater scene reconstruction. In *AAAI*, 2026. arXiv:2601.04984.

Minseong Kweon and Junaed Sattar. Swimm3R: Splatting with medium-aware SfM for underwater 3D reconstruction. *arXiv preprint arXiv:2608.00950*, 2026.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3D with MASt3R. In *ECCV*, 2024.

Deborah Levy, Amit Peleg, Naama Pearl, Dan Rosenbaum, Derya Akkaynak, Simon Korman, and Tali Treibitz. SeaThru-NeRF: Neural radiance fields in scattering media. In *CVPR*, 2023.

Chongyi Li, Chunle Guo, Wenqi Ren, Runmin Cong, Junhui Hou, Sam Kwong, and Dacheng Tao. An underwater image enhancement benchmark dataset and beyond. *IEEE Transactions on Image Processing*, 2020.

Huapeng Li, Wenxuan Song, Tianao Xu, Alexandre Elsig, and Jonas Kulhanek. WaterSplatting: Fast underwater 3D scene reconstruction using Gaussian splatting. In *3DV*, 2025a.

Jiachen Li, Guangzhi Han, Jin Wan, Yuan Gao, and Delong Han. DualPhys-GS: Dual physically-guided 3D Gaussian splatting for underwater scene reconstruction. *Computers & Graphics*, 133: 104405, 2025b.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision. In *CVPR*, 2024.

Shaohua Liu, Junzhe Lu, Zuoya Gu, Jiajun Li, and Yue Deng. Aquatic-GS: A hybrid 3D representation for underwater scenes. *arXiv preprint arXiv:2411.00239*, 2024.

Shaohua Liu, Ning Gao, Zuoya Gu, Hongkun Dou, Yue Deng, and Hongjue Li. Spatiotemporal degradation-aware 3D Gaussian splatting for realistic underwater scene reconstruction. In *ACM MM*, 2025. arXiv:2604.23551.

Nir Mualem, Roy Amoyal, Oren Freifeld, and Derya Akkaynak. Gaussian splashing enables direct volumetric rendering underwater. *Scientific Reports*, 16:29658, 2026.

Lintao Peng, Chunli Zhu, and Liheng Bian. U-shape transformer for underwater image enhancement. *IEEE Transactions on Image Processing*, 32:3066–3079, 2023.

Jiangwei Ren, Xingyu Jiang, Zijie Song, Wei Xu, Hongkai Lin, Dingkang Liang, and Xiang Bai. Wat3R: Underwater 3D geometry learning without underwater annotations. In *ECCV*, 2026. arXiv:2607.08772.

Haoran Wang, Nantheera Anantrasirichai, Fan Zhang, and David Bull. UW-GS: Distractor-aware 3D Gaussian splatting for enhanced underwater scene reconstruction. In *WACV*, 2025a. arXiv:2410.01517.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *CVPR*, 2025b.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *CVPR*, 2026. arXiv:2605.15195.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. DUSt3R: Geometric 3D vision made easy. In *CVPR*, 2024.

Wenpeng Xing, Jie Chen, Zaifeng Yang, Changting Lin, Jianfeng Dong, Chaochao Chen, Xun Zhou, and Meng Han. UW-3DGS: Underwater 3D reconstruction with physics-aware Gaussian splatting. *arXiv preprint arXiv:2508.06169*, 2025.

Jiayi Xu, Jiahao Lu, Ziqiang Zheng, Yihao Tan, Yaolong Zhu, Yuan Liu, and Sai-Kit Yeung. WAT3R: Feedforward underwater 3D reconstruction. *arXiv preprint arXiv:2607.21023*, 2026.

Daniel Yang, John J. Leonard, and Yogesh Girdhar. SeaSplat: Representing underwater scenes with 3D Gaussian splatting and a physically grounded image formation model. In *ICRA*, 2025.

Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3D Gaussian splats from sparse unposed images. In *ICLR*, 2025.

Botao Ye, Boqi Chen, Haofei Xu, Daniel Barath, and Marc Pollefeys. YoNoSplat: You only need one model for feedforward 3D Gaussian splatting. In *ICLR*, 2026.

Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for Gaussian splatting. *arXiv preprint arXiv:2409.06765*, 2024.

Jieyu Yuan, Yujun Li, Yuanlin Zhang, Chunle Guo, Xiongxin Tang, Ruixing Wang, and Chongyi Li. 3D-UIR: 3D Gaussian for underwater 3D scene reconstruction via physics based appearance-medium decoupling. *IEEE Transactions on Image Processing*, 2026. arXiv:2505.21238.

Xinrui Zhang, Yufeng Wang, Shuangkang Fang, Zesheng Wang, Dacheng Qi, and Wenrui Ding. WaterClear-GS: Optical-aware Gaussian splatting for underwater reconstruction and restoration. *arXiv preprint arXiv:2601.19753*, 2026.

A Implementation Details

A.1 Architecture

Gaussian head. Tokens have width 1024 and patch size 16. The feature adapter maps four VGGT-Ω taps onto the four DPT levels that AnySplat reads from its aggregator layers 4, 11, 17 and 23, through one full-rank affine map per level; the taps are VGGT-Ω layers 4, 11, 11 and 11, so layer 11 feeds three levels through separate maps. All four refinement blocks of F_d are shared. Each branch predicts 83 Gaussian channels and a fusion confidence per pixel, and voxel fusion uses a voxel size of 0.002 in normalised scene coordinates.

Medium head. Its inputs are the 83 raw channels of the in-water branch and the three image channels, the two convolutions use SiLU, and a sigmoid bounds $\boldsymbol{\beta}_a$ and $\boldsymbol{\beta}_b$ to $[10^{-4}, 2]$. The physical loss detaches the rendered clean colour and z-depth but not the intrinsics that convert z-depth to ray range, so it trains the medium head and, through the intrinsics, the shared aggregator; an autograd check confirms that it reaches both but not C.

Background. Renders are composited as $\alpha C_{\text{fg}} + (1 - \alpha)b$, with b_w the mean context colour and $b_c = \text{clip}(b_w + 0.5 \tanh(h(b_w)), 0, 1)$ for a learned MLP h.

A.2 Training

Initialisation. The initialisation combines VGGT-Ω and AnySplat modules: the feature adapter is first aligned on 500 training sources against a frozen AnySplat teacher, followed by a 50-step decoder warm-up with learning rate 2×10^{-7} and a frozen backbone. Neither step uses validation or test sources or counts toward the reported training budget.

Optimisation. We use AdamW with $\beta_2 = 0.95$ and weight decay 0.05, learning rates of 10^{-4} for the Gaussian head, 10^{-5} for the adapter and backbone and 2×10^{-4} for the medium head, and gradient clipping at norm 0.5. Each stage restarts the optimiser with a 100-step warm-up and decays to 0.1 of its learning rate. Each of the eight GPUs packs $\lfloor 24/n \rfloor$ scenes at the sampled view count n, and images and intrinsics are cropped to 224×224; field-of-view targets come from the cropped intrinsics, and extrinsics, depth and confidence from the frozen teacher. After step 100, a loss guard scales the gradients by 10^{-10} whenever the total loss exceeds 0.2. The model uses seed 20260902, and Table 7 lists the loss weights.

Table 7: **Loss weights of Equation (4).** The RGB and physical terms average the context-view and target-view means.

Term	Symbol	Weight
In-water RGB	λ_w	1
Clean RGB	λ_c	1
LPIPS term in $\mathcal{L}_{\text{rgb}}$	λ_{lp}	0.05
Image-gradient term in $\mathcal{L}_{\text{rgb}}$	λ_g	0.05
Physical re-rendering (MSE)	λ_p	0.5
Medium labels, sum of the two log-coefficient losses	in $\mathcal{L}_\theta$	0.1
Medium labels, veiling light (MSE)	in $\mathcal{L}_\theta$	1.0
Teacher pose	λ_{pose}	10
Teacher depth	λ_d	1
Depth consistency	λ_{dc}	0.1

Upstream data. VGGT, AnySplat and the DL3DV checkpoint of YoNoSplat were pretrained on DL3DV without complete source lists. The YoNoSplat release excludes its official 140-scene benchmark, but 30 of our 32 Test32 sources lie outside that list, so YoNoSplat-D and our model may have seen Test32 sources upstream. The VGGT-Ω authors have also reported an unresolved contamination concern with the original checkpoint and recommend a retrained one for benchmarking;[2] our model starts from the original checkpoint. OOD-Water-2, used only in Appendix E.2, shares three Tanks and Temples scenes with SpatialBench, one of the benchmarks re-evaluated in that notice.

[2] Official VGGT-Ω reproduction notice, commit a3ab0141.

B Data and Evaluation Details

B.1 Data Construction

DL3DV-Water. The two water conditions of each source are drawn from the Jerlov classes IB, II, III, 1C, 3C and 5C, giving 3,992 variants and 255,488 frames whose clean and underwater images share cameras. Because the pseudo-ranges are fixed by the $4\,\mathrm{m}$ gauge, the medium labels are exact for this construction rather than measured water properties. The 64 views of each source split into 56 context candidates and eight query candidates; the main protocol uses 24 fixed context views and two of the queries for each water condition. The range teacher was fine-tuned on clean DL3DV before our split, so held-out sources may have contributed to it.

Optical-depth augmentation. The second stage adds the two range-scaled copies described in Section 3.3 to every training variant, giving 11,568 training variants; re-rendering from the recovered ranges reproduces the stored underwater images at $57\,\mathrm{dB}$ on average. A matched-budget control of this stage is part of Appendix E.

Transfer sets. Mip360-Water renders each of its seven scenes under two conditions. OOD-Water contains Truck and Train from Tanks and Temples, DrJohnson and Playroom from Deep Blending, Flowers and Treehill from Mip-NeRF 360, and Flower, Fortress, Horns, Room and T-Rex from LLFF; its images are cover-cropped to 796×448 and synthesised in linear light with fixed-seed conditions.

DeepSea-ROV and Real25. DeepSea-ROV contains 263 segments (86,079 frames at 960×576 or 2560×1440) from 22 recordings of a towed deep-sea vehicle in the South Pacific; the vehicle lighting moves with the camera, the green dots are laser markers, and the provider withholds exact dates and positions. Registration succeeded for 34 candidate clips and failed for three. Each clip has 32 frames: every eighth is a held-out query, seven more serve as supervision targets when the clip is used for adaptation and are never scored, and the remaining 21 are contexts. The eight test clips come from four dives. Water3D contributes 28 adaptation and 13 test clips, and the four SeaThru-NeRF scenes are test data only. Each Real25 clip has 11 to 21 context frames and two to four queries, 93 queries in total.

B.2 Baselines

Feed-forward. Each adapted AnySplat model adds $8000+3000$ updates on four GPUs with its native objectives and rectangular training. Its scores depend on the protocol: on Test32 it reaches $12.69\,\mathrm{dB}$ (C) and $18.12\,\mathrm{dB}$ (W) under the square protocol of Table 1, against $17.68\,\mathrm{dB}$ (C) and $25.42\,\mathrm{dB}$ (W) under its native 224×448 protocol. Each adapted YoNoSplat model adds $4000 + 1500$ updates on eight GPUs with the native loss weights (MSE 1, LPIPS 0.05, opacity 0.01, focal 0.5, pose 0.1) and the recipe of the release: all weights train with learning rate 10^{-4} (backbone $\times 0.1$), cosine decay to 10^{-5} and ground-truth poses in training with probability 0.9; inference uses no pose or focal labels. The DL3DV initialisation was selected on validation data, and the RE10K initialisation follows the same schedule.

Per-scene. The per-scene methods, with 3DGS as splatfacto in nerfstudio, start from the cameras and 100,000 points predicted by the frozen VGGT-Ω initialisation, run 15,000 steps with their released code and settings, and are scored at the last step; held-out images appear only as black placeholders in the evaluation split, so the optimisers never see their pixels and train on the context views alone. Schedule lengths that a released recipe ties to its total budget are rescaled to 15,000 steps, the published budget of Gaussian Splashing. WaterSplatting runs its PyTorch fallback, and 3D-UIR runs without its distance-guided regularisation optimisation (DRO), which needs structure-from-motion keypoint tracks that a predicted initialisation does not have.

Single-image. The single-image methods run from their released code and weights without retraining. Fusion uses its multi-scale mode, FUnIE-GAN and U-shape process their required 256×256 input and are resized back, U-shape keeps its official per-image normalisation, and Sea-thru receives our predicted range.

B.3 Evaluation Protocol and Statistics

Alignment. Every feed-forward model receives the same unposed 224×224 context views. Held-out query cameras are mapped into each model's frame by a similarity fitted to the context cameras only, without pose or photometric refinement; the restoration and dense-view studies use the protocols of Appendices C.4 and C.5.

Aggregation. Metrics are averaged over the queries of a variant, then over the water conditions of a source, then equally over sources, or over clips for Real25. A paired difference resamples the per-source differences 10,000 times and reports the 95% percentile interval, and it is resolved when this interval excludes zero. The intervals condition on the fixed checkpoints, exclude training-seed variation and are not corrected for multiple comparisons.

Cost. Our model trains in 6.24 hours on eight GPUs including evaluation, about 50 GPU-hours without initialisation and inherited pretraining, against 47 for the adapted YoNoSplat pair and an estimated 121 for the AnySplat pair. Latency is the median encode-and-render time of a 24-view Test32 scene, excluding I/O time and the first scene: 0.77 s for our W, C and P, and, summing the two models of a pair, 0.61 s for YoNoSplat and 1.02 s for AnySplat.

C Additional Results

C.1 Per-Set Results and Uncertainty

Transfer sets. Table 8 separates the two transfer sets that Table 1 pools. Against the adapted baselines, W has the best PSNR and LPIPS on both sets at every view count, and C the best LPIPS everywhere and the best PSNR on OOD-Water at every view count and on Mip360-Water from 16 views.

Table 8: **The two transfer sets of Table 1, reported separately.** W and P are scored against the underwater observation and C against the clean reference.

Method	8 views			16 views			24 views		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Mip360-Water (7 sources): W and P against the observation									
AnySplat (rect.)	17.25	0.572	0.571	18.63	0.585	0.563	19.00	0.581	0.564
YoNoSplat-R	22.86	0.676	0.499	23.02	0.675	0.493	22.20	0.663	0.502
YoNoSplat-D	22.64	0.672	0.503	22.44	0.672	0.498	22.23	0.673	0.500
Ours (W)	**22.92**	0.661	**0.468**	**23.43**	0.666	**0.440**	23.54	0.670	**0.428**
Ours (P)	22.77	**0.680**	0.478	23.41	**0.692**	0.453	**23.60**	**0.697**	0.444
OOD-Water (11 sources): W and P against the observation									
AnySplat (rect.)	15.65	0.534	0.592	16.52	0.564	0.571	17.37	0.582	0.570
YoNoSplat-R	18.39	0.531	0.512	18.68	0.561	0.505	19.15	0.580	0.501
YoNoSplat-D	19.93	0.600	0.510	19.66	0.604	0.512	19.56	0.605	0.514
Ours (W)	**21.41**	0.611	**0.434**	**22.17**	0.639	**0.405**	**22.31**	0.630	**0.391**
Ours (P)	20.79	**0.620**	0.453	21.25	**0.643**	0.437	21.76	**0.640**	0.414
Mip360-Water: C against the clean reference									
AnySplat (rect.)	10.84	0.202	0.709	11.42	0.203	0.695	11.67	0.206	0.699
YoNoSplat-R	**14.54**	**0.265**	0.655	14.40	**0.266**	0.649	14.46	**0.268**	0.644
YoNoSplat-D	14.33	0.261	0.653	14.42	0.261	0.646	14.50	0.263	0.641
Ours (C)	13.99	0.243	**0.635**	**14.61**	0.257	**0.603**	**14.87**	0.264	**0.588**
OOD-Water: C against the clean reference									
AnySplat (rect.)	11.03	0.253	0.695	11.76	0.275	0.688	11.85	0.277	0.688
YoNoSplat-R	12.35	0.273	0.629	12.54	0.269	0.628	13.13	0.282	0.614
YoNoSplat-D	12.64	0.289	0.636	12.52	0.280	0.641	12.57	0.280	0.637
Ours (C)	**13.96**	**0.299**	**0.603**	**14.47**	**0.341**	**0.574**	**14.93**	**0.336**	**0.546**

Uncertainty. Table 9 gives the paired intervals of our model against each adapted YoNoSplat model at 24 views. The PSNR and LPIPS gains of W are resolved on Real25 and on both transfer sets, as are all three gains of C on OOD-Water; on Test32 our model trails, as in Table 1.

Table 9: **Paired differences behind Table 1, 24 views.** Ours minus each adapted YoNoSplat model. PSNR in dB over its 95% interval; *: the interval excludes zero.

		vs. YoNoSplat-R			vs. YoNoSplat-D		
Set	Output	ΔPSNR↑	ΔSSIM↑	ΔLPIPS↓	ΔPSNR↑	ΔSSIM↑	ΔLPIPS↓
Real25	W	$+0.98^{*}$ $[+0.38, +1.60]$	-0.012	-0.076^{*}	$+1.05^{*}$ $[+0.38, +1.72]$	-0.010	-0.080^{*}
Mip360-Water	W	$+1.34^{*}$ $[+0.15, +2.83]$	$+0.007$	-0.074^{*}	$+1.31^{*}$ $[+0.39, +2.33]$	-0.004	-0.072^{*}
	C	$+0.40$ $[-0.50, +1.21]$	-0.004	-0.056^{*}	$+0.36$ $[-0.80, +1.32]$	$+0.001$	-0.052^{*}
OOD-Water	W	$+3.16^{*}$ $[+1.36, +5.24]$	$+0.050$	-0.110^{*}	$+2.75^{*}$ $[+1.17, +4.49]$	$+0.025$	-0.123^{*}
	C	$+1.80^{*}$ $[+0.63, +3.07]$	$+0.055^{*}$	-0.068^{*}	$+2.36^{*}$ $[+1.20, +3.53]$	$+0.056^{*}$	-0.090^{*}
Test32	W	-0.28 $[-0.87, +0.36]$	-0.007	$+0.019^{*}$	-0.90^{*} $[-1.46, -0.28]$	-0.024	$+0.036^{*}$
	C	-0.86^{*} $[-1.39, -0.30]$	-0.020	$+0.050^{*}$	-1.25^{*} $[-1.77, -0.73]$	-0.039^{*}	$+0.073^{*}$

C.2 Second Training Seed

We repeat the complete training with seed 20260911. As shown in Table 10, the PSNR of every output changes by at most 0.33 dB between the two seeds, and both seeds keep the 24-view PSNR leads of W on Real25 and both transfer sets and of C on both transfer sets.

Table 10: **Two training seeds of the full model, 24 views.** Mean ± SD is over the two seeds; seed 1 is the model used throughout.

		PSNR by seed		Mean ± SD over seeds		
Set	Output	Seed 1	Seed 2	PSNR↑	SSIM↑	LPIPS↓
Real25	W	19.53	19.68	19.60 ± 0.10	0.407 ± 0.007	0.494 ± 0.001
	P	18.34	18.44	18.39 ± 0.07	0.398 ± 0.004	0.539 ± 0.007
Mip360-Water	C	14.87	14.63	14.75 ± 0.17	0.260 ± 0.005	0.591 ± 0.004
	W	23.54	23.23	23.39 ± 0.22	0.668 ± 0.003	0.432 ± 0.006
	P	23.60	23.27	23.44 ± 0.24	0.693 ± 0.006	0.448 ± 0.005
OOD-Water	C	14.93	14.94	14.94 ± 0.01	0.343 ± 0.009	0.542 ± 0.006
	W	22.31	22.35	22.33 ± 0.03	0.635 ± 0.006	0.389 ± 0.003
	P	21.76	22.00	21.88 ± 0.17	0.646 ± 0.008	0.411 ± 0.004
Test32	C	18.25	18.16	18.20 ± 0.06	0.518 ± 0.002	0.339 ± 0.001
	W	25.76	25.73	25.75 ± 0.02	0.728 ± 0.000	0.223 ± 0.000
	P	24.72	24.75	24.73 ± 0.02	0.736 ± 0.001	0.258 ± 0.003

C.3 Camera Evaluation

Protocol. We export the context cameras of every model on the Real25 inputs and compare all camera pairs within a clip with the registered reference poses using the public Wat3R evaluator (Ren et al., 2026): RRA and RTA at 5° measure the rotation and the sign-folded translation direction, and AUC@30° integrates the larger of the two errors in one-degree bins. Scores average within a clip and then over clips; registered poses are reference estimates rather than independent ground truth.

Wat3R. The released Wat3R runs through its own preprocessing and pose head on the shared 224×224 frames and, in a second row of Table 2, on the same pixels upsampled to its default 518^2 input, which preserves the field of view. At 518^2 its margin over the VGGT-Ω backbone on SeaThru-NeRF (6.7 points) is close to the margin over VGGT that its authors report (6.3 points), which supports our use of the release. Per clip, our AUC@30° is higher on 17 of the 25 clips, including all eight DeepSea-ROV clips, whose dives neither model was trained on.

One camera set. The adapted YoNoSplat baselines are two separate models and therefore predict two camera sets for the same input. Over all 4405 camera pairs within the Real25 clips, the relative transforms of the YoNoSplat-R C and W models differ by 5.74° in rotation and 14.37° in translation

direction, and those of the YoNoSplat-D models by 5.12° and 14.84°; our W, C and P share one camera set by construction.

C.4 Comparison with Per-Scene Optimisation

Protocol. Five long DeepSea-ROV test sequences provide nested sets of 32, 48 and 64 context views and eight fixed held-out views each. The per-scene methods follow Appendix B.2, and held-out cameras are mapped into each reconstruction by the similarity fitted to its context cameras. Redrawing the 32- and 48-view subsets at random with three seeds keeps our lead over WaterSplatting, and so does weighting the four underlying acquisitions equally instead of the five sequences.

Cameras. Every score in Table 5 includes the error of mapping a held-out camera into a predicted reconstruction. To test how much of the gap comes from camera placement, we freeze each 64-view reconstruction and refit only the extrinsics of every held-out camera on its held-out image. As shown in Table 11, the refit lifts the per-scene optimisers more than our model, which still leads, so part of their deficit is misplaced texture. The refit keeps the test intrinsics and does not correct the training focal lengths of the predicted initialisation, 0.69 to 0.84 times the reference; initialised instead from structure from motion on the context views with the reference intrinsics, WaterSplatting and 3DGS exceed every run from predicted cameras.

Table 11: **Camera diagnostics at 64 context views (PSNR).** Refitted: held-out camera extrinsics fitted on the held-out image with the reconstruction frozen; SfM: initialised from structure from motion on the context views.

Method	Protocol	Refitted	Δ	SfM initialisation
3DGS	16.82	19.76	+2.94	<u>25.35</u>
WaterSplatting	<u>18.61</u>	<u>20.15</u>	+1.54	**25.65**
Ours (W)	**19.26**	**20.52**	+1.25	–

Examples. Figure 7 shows held-out in-water views. The real view was picked by inspecting the 40 held-out views of the five sequences, and each synthetic view among the ten Test32 views, one per scene, where our LPIPS margin over WaterSplatting is largest. The per-scene optimisers keep sharper texture but misplace it under the transported cameras, whereas our renders are smoother and in place.

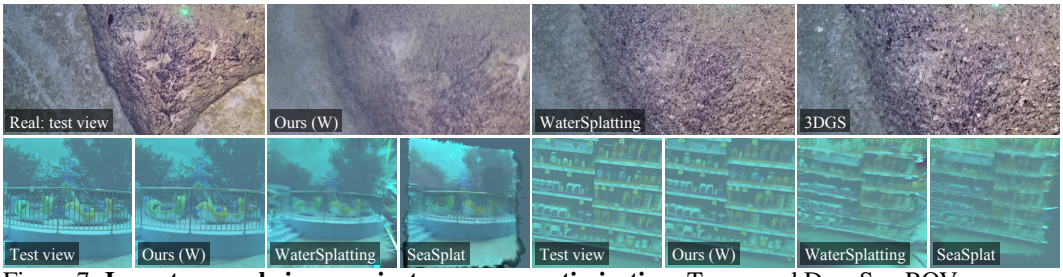

Figure 7: **In-water novel views against per-scene optimisation.** Top: a real DeepSea-ROV sequence with 64 views; bottom: two Test32 scenes with 56 views. The views are selected examples.

C.5 Restoration

Protocols. C is scored on the synthetic sets, which have a clean reference. The restoration panel of Tables 3 and 4 uses ten Test32 sources with 56 context views and eight held-out queries each, the largest split that 64 views allow, and averages the 72 of its 80 queries on which every method returns an output (Sea-thru's optimiser fails on the other eight). The 24-view protocol of Table 1 supports only feed-forward and single-image methods and is also run on Mip360-Water, outside our training sources.

Single-image dewatering at 24 views. As shown in Table 12, rendering W and then dewatering it is worse than predicting C directly, by 3.69 dB on Test32 and 2.18 dB on Mip360-Water against the best such result, with every interval resolved. A single-image method that processes the scored frame itself synthesises no view; on Test32, C still leads Fusion, FUnIE-GAN and U-shape in PSNR



Figure 8: **Restoration at held-out Test32 views, 56 context views.** Of the four scenes whose observed view departs furthest from its clean reference, chosen without looking at any output, the three not shown in Figure 5.

by 2.16 to 2.75 dB while trailing in SSIM, because these methods edit the original pixels, and on Mip360-Water U-shape leads C.

Table 12: **Restoration under the 24-view protocol.** Scored against the clean reference on the frames common to all rows (120 of 128 on Test32, where Sea-thru's optimiser fails on eight, and all 28 on Mip360-Water). ↪: our W dewatered at that camera; *: resolved against ours (C).

	Test32 (in domain)			Mip360-Water (transfer)		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Renders the held-out view						
Ours (C)	**18.28**	**0.519**	**0.339**	**14.87**	**0.264**	**0.588**
Ours (W), no dewatering	12.19*	0.418*	0.570*	11.40*	0.249*	0.740*
↪ Fusion	14.58*	0.423*	0.505*	12.64*	0.239*	0.705*
↪ FUnIE-GAN	14.48*	0.424*	0.511*	11.88*	0.226*	0.713*
↪ U-shape	13.93*	0.472*	0.535*	12.69*	0.243*	0.707*
↪ Sea-thru	10.71*	0.362*	0.519*	11.22*	0.182*	0.687*
Processes the scored frame itself (upper reference)						
Observed frame, no restoration	12.80*	0.654*	0.419*	12.36*	0.546*	0.466*
Fusion	15.75*	**0.697***	0.384*	14.20	0.580*	0.469*
FUnIE-GAN	**16.12***	0.686*	0.387*	13.74	0.550*	0.478*
U-shape	15.53*	0.669*	0.428*	**16.35***	0.594*	0.467*
Sea-thru	11.42*	0.638*	**0.367**	13.50	**0.645***	**0.391***

Per-scene optimisers. As shown in Table 13, the per-scene optimisers do render the observed water: on the same panel their in-water outputs reach 19.4 to 20.0 dB, except Gaussian Splashing, whose published densification grows it to 14.7 million Gaussians on average. Their restored outputs nevertheless fall below the untouched observation, which scores 12.96 dB against the clean reference on the 72 common queries. Rescaling every image by its best exposure gain over all 80 queries keeps WaterSplatting below the observation: WaterSplatting reaches 10.99 dB, or 11.61 dB with its tone-mapping branch, the observation 13.20 dB and C 18.95 dB. Figure 8 shows further scenes.

Table 13: **In-water renders on the restoration panel of Table 3.** Ten Test32 sources, 56 context views, all 80 held-out queries, scored against the observation.

Method	PSNR↑	SSIM↑	LPIPS↓
WaterSplatting	19.97	0.535	0.472
SeaSplat	19.36	0.533	0.494
3D-UIR (DRO off)	19.69	0.526	0.473
Gaussian Splashing	12.86	0.291	0.629
Ours (W)	**26.75**	**0.753**	**0.198**

D Additional Ablations and Medium Analysis

D.1 Gradient Routing

The stop-gradient of Section 3.1 was chosen after a pilot on an earlier checkpoint of the same architecture. The checkpoint was adapted for 1500 updates on the 54 real adaptation clips, once with the physical loss stopped at C and once with it also reaching C and its depth (joint routing), and scored on the Real25 clips under the protocol of that time, so its absolute values are not comparable with Table 1. As shown in Table 14, both routings give similar W drift, whereas joint routing gains 0.5 to 0.9 dB of physical PSNR by moving C 3.2 to 6.6 dB further from its unadapted output. Figure 9

Table 14: **Gradient-routing pilot on the Real25 clips (historical checkpoint).** Drift is the PSNR between the adapted and the unadapted output at the same views; a lower value means the output moved further.

	SeaThru-NeRF (4)	Water3D (13)	DeepSea-ROV (8)
P, joint routing minus stop-gradient (dB)	+0.94	+0.93	+0.53
C drift, stop-gradient	25.17	25.19	19.55
C drift, joint routing	21.59	18.64	16.37
W drift, stop-gradient	29.21	30.87	27.74
W drift, joint routing	29.63	30.71	27.80

shows where the restoration goes: with joint routing the deep-sea seafloor turns pink, whereas the stop-gradient keeps it close to the unadapted restoration.

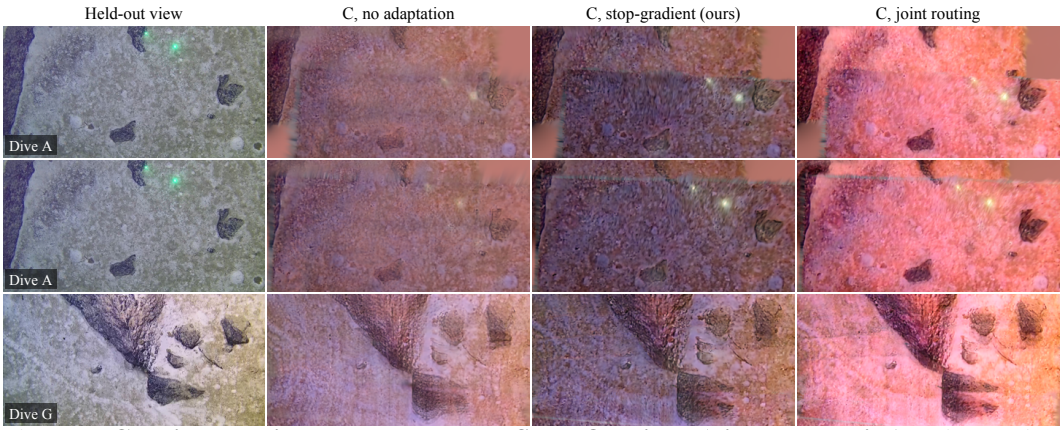

Figure 9: **Gradient routing on held-out DeepSea-ROV views (pilot checkpoint).** Joint routing raises physical consistency by recolouring the restored scene pink.

D.2 Output Count

The two single-output controls of Section 4.3 share the initialisation, data, seed and budget of the dual-output model, and all three train without the physical and medium-label losses. Each single-output control freezes the branch it does not train and masks that branch's RGB losses; both forward paths remain, so the frozen restored branch still supplies the depth-consistency anchor of the W-only control, and only the trained output is scored. Table 15 reports all three metrics of these controls on every set, for C only where a clean reference exists.

D.3 Loss Groups and Physical-Loss Weight

Loss groups. Each control of Table 6 starts from the initialisation of the full model and completes its $4000 + 1500$ schedule with one loss group removed, keeping both forward branches and all other losses. Parameters trained only by the removed group are frozen, so these are supervision controls rather than head deletions. Removing a loss also changes how often the loss guard of Appendix A.2 fires, so each control changes the optimisation as well as the supervision.

Table 15: **Single- versus dual-output models.** Both train without the physical and medium-label losses. *: resolved against the dual model.

	Real25			Mip360-Water			OOD-Water			Test32		
Model	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
In water: W against the observation												
Single	19.60	**0.410***	**0.487**	**23.64**	**0.670**	**0.421**	**22.51**	**0.643***	**0.388**	**26.15***	**0.742***	**0.212***
Dual	**19.70**	0.404	0.493	23.51	0.669	0.428	22.16	0.623	0.398	25.86	0.727	0.223
Restored: C against the clean reference												
Single	–	–	–	14.83	0.261	0.592	14.78	0.328	0.556	**18.39**	**0.531**	**0.336**
Dual	–	–	–	**14.85**	**0.263**	**0.588**	**14.86**	**0.330**	**0.548**	18.23	0.519	0.340

Physical-loss weight. We train the full recipe with λ_p halved and doubled, one seed each. As shown in Table 16, neither weight changes Test32 PSNR resolvedly, both raise Real25 W and P, and at every weight Real25 P stays above the 17.19 dB that Table 6 reports without the physical and medium-label losses, by 1.15 to 1.51 dB with every paired interval above zero. We keep $\lambda_p = 0.5$, which was fixed before these runs, because choosing the weight on Real25 would turn a test set into a validation set.

Table 16: **Physical-loss weight.** The middle row is the full model; *: resolved against it.

	Test32 C			Test32 W			Real25 W			Real25 P		
λ_p	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
0.25	18.16	0.509	0.341	25.68	0.719	0.225	**20.22***	**0.423***	**0.485**	**18.70***	**0.411***	**0.526**
0.5	18.25	0.520	0.339	25.76	0.728	0.222	19.53	0.402	0.493	18.34	0.395	0.534
1.0	**18.38**	**0.529**	**0.332***	**25.93**	**0.734**	**0.216***	19.71*	0.409*	0.490	18.50*	0.399	0.527*

D.4 Medium Analysis

Coefficient accuracy. Coefficients are compared in the 4 m gauge that the synthesis and the prediction share; real water has no measured reference. As shown in Table 17a, the medium head is more accurate than a constant predictor made from the training labels on all three medium parameters and on both synthetic sets, and it more than halves the backscatter error. Table 17b compares the one-stage coverage models of Section 4.4: wider coverage roughly doubles the attenuation and backscatter errors and leaves the veiling-light error unchanged.

Table 17: **Medium recovery in the 4 m gauge.** Mean absolute log-coefficient error for attenuation and backscatter and mean absolute linear-RGB error for veiling light; lower is better. (b) uses the one-stage coverage models at 1928 sources, two seeds averaged.

	Set	Predictor	Attenuation	Backscatter	Veiling light
(a)	Test32	Training-label median	0.327	0.648	0.029
		Ours (full model)	**0.226**	**0.273**	**0.011**
	Mip360-Water	Training-label median	0.334	0.660	0.024
		Ours (full model)	**0.218**	**0.302**	**0.012**
(b)	Test32	Original coverage	0.111	0.129	0.012
		Wider coverage	0.211	0.258	0.012

Scene and medium swaps. To locate the Real25 P gain of wider coverage, we exchange the restored scene (C and range) and the medium between the original-coverage model A and the wider-coverage model B before physical formation. These are the models of the coverage confirmation: recomposing a model from its own scene and medium reproduces its output exactly, so exchanging both, the sum of the rows 'On scene B' and 'Under medium A' of Table 18, gives the Real25, coral and new-dive P rows of Table 22 up to rounding. As shown in Table 18, the medium of B raises the PSNR of P on its own scene in every real set but on the scene of A only on Real25, and on the scene of A, swapping $\boldsymbol{\beta}$ alone closely approximates the change of the whole medium; exchanging the whole medium improves LPIPS and lowers SSIM throughout. The gain thus depends on the pairing of scene and medium, and it does not identify a physically correct decomposition.

Medium replacement. Because the scene and the medium are separate, a restored scene can be re-rendered through another medium. Figure 10 keeps C, range and camera fixed and applies the

Table 18: **Scene and medium swaps between coverage models A and B.** The confirmation models of Table 22 at 1928 sources, two seeds averaged. Each row exchanges the restored scene (C and range) or the medium and reports the change in P; *: the paired interval excludes zero.

	Real25			Coral clips			New dive		
Comparison	ΔPSNR	ΔSSIM	ΔLPIPS	ΔPSNR	ΔSSIM	ΔLPIPS	ΔPSNR	ΔSSIM	ΔLPIPS
Medium of B minus medium of A									
On scene A	$+0.68^{*}$	-0.035^{*}	-0.064^{*}	-0.07	-0.020^{*}	-0.067^{*}	-0.27	-0.096^{*}	-0.249^{*}
On scene A, β only	$+0.69^{*}$	-0.034^{*}	-0.063^{*}	-0.07	-0.020^{*}	-0.067^{*}	-0.25	-0.094^{*}	-0.251^{*}
On scene A, $\mathbf{B}_{\infty}$ only	-0.08	$+0.000$	-0.006	-0.73^{*}	-0.003^{*}	-0.014^{*}	-0.18^{*}	-0.002^{*}	$+0.003$
On scene B	$+1.42^{*}$	-0.022^{*}	-0.090^{*}	$+0.56^{*}$	-0.014^{*}	-0.130^{*}	$+1.24^{*}$	-0.077^{*}	-0.274^{*}
Scene of B minus scene of A									
Under medium A	-0.35^{*}	$+0.005$	$+0.018^{*}$	-0.17^{*}	$+0.000$	$+0.035^{*}$	-0.30	$+0.001$	$+0.027^{*}$
Under medium B	$+0.40^{*}$	$+0.018^{*}$	-0.008	$+0.45^{*}$	$+0.006^{*}$	-0.028^{*}	$+1.20$	$+0.019$	$+0.002$

scene's own medium or that of another scene; the supplementary material also contains two deep-sea videos of W, C and P along interpolated camera paths.

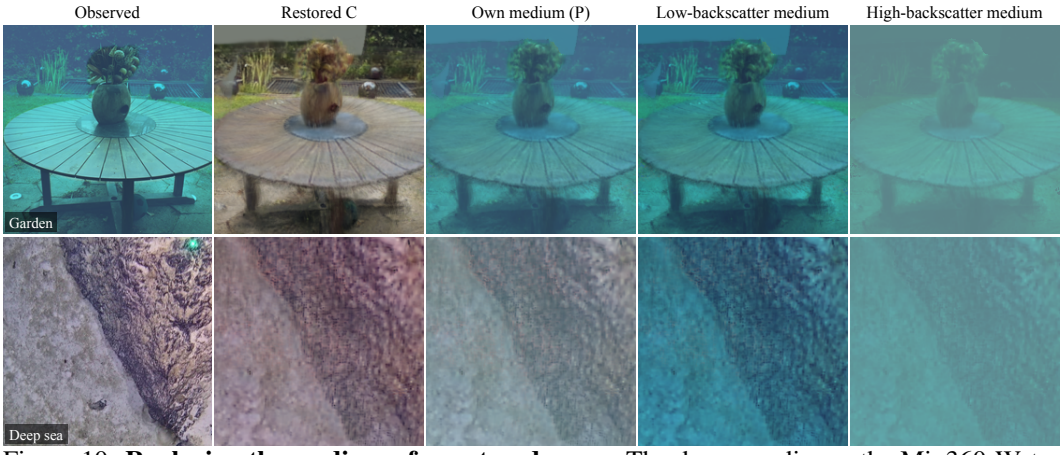

Figure 10: **Replacing the medium of a restored scene.** The donor media are the Mip360-Water variants with the lowest and the highest predicted green backscatter; the figure shows controllable formation, not measured water.

E Data-Effects Study

Section 4.4 varies the training data and schedule at a fixed architecture. Table 19 lists the designs, and Tables 20 and 21 list the paired contrasts, including all those drawn in Figure 6.

Common protocol. Each run changes one named factor, and its seeds, data lists, initialisation and endpoint are fixed before its evaluation. The source-count and coverage runs use seeds 20260902 and 20260911, the real-data continuations seed 20260902, and the confirmation seeds 20260915 and 20260916. Paired intervals resample sources after averaging the seeds of each source.

Coverage designs. Design A assigns each source one of its two original water conditions, design B one of six slots in round-robin order (the two water conditions, each at the original range and at the two range scales of Section 3.3), and design C all six; the one-stage source-count contrast of Figure 6a uses design A. At a fixed budget, C draws each of its six entries per source one sixth as often as A or B draws its single entry.

Real data. The real-data runs restart the optimiser from the full model and mix a fixed set of 162 synthetic sources with 14 registered clips (7 from DeepSea-ROV, 7 from Water3D), 54 clips (26 and 28), or 54 clips and 232 unregistered DeepSea-ROV segments with teacher pseudo-poses, for 1000 updates. Because the shortest registered clip has 13 frames, these continuations sample 2 to 10 context views instead of 2 to 24. Real clips supervise W, the physical loss and the distillation losses only. Their share of the mixture grows with the clip count (8, 25 and 64%), so the number of clips is not isolated. The 14- and 54-clip rows are measured against a continuation without real clips, which

Table 19: **Designs of the data-effects study.**

Factor	Levels	Updates
Source count	nested 480, 964 and 1928 sources, two complete seeds; one-stage variant at 480 and 1928	4000 + 1500; one stage 4000
Medium coverage	original (A), wider (B) and all-conditions (C) designs at 480 and 1928 sources, two seeds per cell	4000
Range stage	original versus range-augmented data from the same base endpoint, two seeds	1500
Training budget	120, 240, 480, 960 and 1928 sources at about 80 draws per source, two seeds	1000 to 16000
Real data	0, 14 or 54 registered clips, or 54 clips and 232 unregistered segments, from the full model	1000; one control 3000
Confirmation	designs A and B at 1928 sources, two new seeds, two new test sets and a secondary set	4000

Table 20: **Paired contrasts of source count and medium coverage.** PSNR difference in dB over its 95% paired interval, two seeds averaged per source; *: the interval excludes zero.

	Test32		OOD-Water	Real25	
Contrast	C	W	W	W	P
(a) Source count, minus 480 sources					
964, full recipe	$+0.40^*$ $[+0.17, +0.62]$	$+0.43^*$ $[+0.13, +0.73]$	$+0.01$ $[-0.16, +0.19]$	-0.10 $[-0.28, +0.06]$	-0.04 $[-0.21, +0.14]$
1928, full recipe	$+0.22^*$ $[+0.01, +0.43]$	$+0.18$ $[-0.08, +0.46]$	$+0.69$ $[-0.01, +1.48]$	$+0.10$ $[-0.14, +0.33]$	$+0.05$ $[-0.23, +0.35]$
1928, one stage	$+0.32^*$ $[+0.09, +0.56]$	$+0.22$ $[-0.06, +0.51]$	$+0.89^*$ $[+0.19, +1.70]$	$+0.85^*$ $[+0.56, +1.15]$	$+0.38^*$ $[+0.16, +0.63]$
(b) Medium coverage					
Wider minus original, 480	-0.16^* $[-0.32, -0.01]$	-0.03 $[-0.18, +0.12]$	$+0.23$ $[-0.04, +0.51]$	$+0.40^*$ $[+0.18, +0.67]$	$+1.06^*$ $[+0.51, +1.69]$
Wider minus original, 1928	-0.25^* $[-0.45, -0.08]$	-0.10 $[-0.28, +0.06]$	-0.03 $[-0.47, +0.33]$	$+0.08$ $[-0.07, +0.27]$	$+1.06^*$ $[+0.52, +1.65]$
All six minus wider, 480	$+0.08$ $[-0.05, +0.20]$	-0.07 $[-0.20, +0.06]$	$+0.36$ $[-0.11, +0.94]$	-0.00 $[-0.18, +0.17]$	$+0.13$ $[-0.27, +0.51]$
All six minus wider, 1928	$+0.24^*$ $[+0.07, +0.43]$	$+0.23^*$ $[+0.06, +0.43]$	-0.24 $[-0.65, +0.10]$	-0.44^* $[-0.73, -0.18]$	-0.16 $[-0.39, +0.05]$
Range stage, augmented minus original	-0.25^* $[-0.38, -0.13]$	-0.15^* $[-0.28, -0.02]$	$+0.18$ $[-0.26, +0.71]$	-0.02 $[-0.14, +0.11]$	$+1.00^*$ $[+0.43, +1.64]$

adds two synthetic sources for eight-GPU sharding and is itself below the full model, as the last row of Table 21 shows.

Table 21: **Paired contrasts of training budget and real data.** As in Table 20; the budget row at 1928 sources has seed means only (–: not evaluated), and the real-data rows are one-seed continuations of the full model.

	Test32		OOD-Water	Real25	
Contrast	C	W	W	W	P
(c) Training budget					
16000 minus 4000 updates, 1928	–	$+0.04$	-0.67	-0.55	–
3000 minus 1000 updates, 14 clips	-0.27^* $[-0.50, -0.06]$	-0.24^* $[-0.48, -0.00]$	-0.33^* $[-0.68, -0.07]$	-0.25^* $[-0.41, -0.09]$	$+0.06$ $[-0.14, +0.26]$
(d) Real data					
14 clips minus none	$+0.17$ $[-0.06, +0.43]$	$+0.18$ $[-0.09, +0.48]$	-0.03 $[-0.23, +0.18]$	$+0.86^*$ $[+0.50, +1.23]$	$+0.90^*$ $[+0.67, +1.13]$
54 clips minus none	$+0.07$ $[-0.11, +0.25]$	$+0.11$ $[-0.14, +0.37]$	$+0.09$ $[-0.12, +0.30]$	$+0.83^*$ $[+0.50, +1.20]$	$+1.09^*$ $[+0.78, +1.44]$
Adding 232 segments to 54 clips	$+0.16$ $[-0.05, +0.38]$	$+0.10$ $[-0.18, +0.37]$	-0.00 $[-1.19, +0.92]$	-0.12 $[-0.27, +0.01]$	-0.65^* $[-0.96, -0.36]$
Full model minus zero-clip continuation	$+0.35^*$ $[+0.10, +0.60]$	$+0.48^*$ $[+0.14, +0.82]$	$+0.53^*$ $[+0.18, +0.99]$	$+0.36^*$ $[+0.20, +0.54]$	$+0.86^*$ $[+0.39, +1.40]$

E.1 Equal-Exposure Control

At a fixed budget of 4000 base updates, averaging about ten scenes per update, the two-condition pools of 480, 964 and 1928 sources are drawn about 40, 20 and 10 times per entry, so more sources also means fewer passes over each. To separate the two, the one-condition design trains 120, 240, 480, 960 and 1928 sources for 1000 to 16,000 updates, about 80 draws per source, with two seeds each. As shown in Figure 11, the returns are non-monotone and differ between sets, so we report paired differences and fit no scaling law.

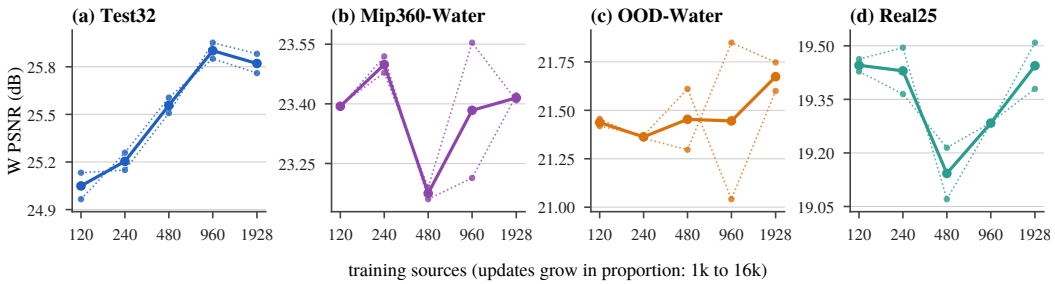

Figure 11: **Source count at equal exposure.** In-water PSNR when 120 to 1928 sources are trained for 1000 to 16,000 updates; solid lines average the two seeds and dotted lines show each seed.

E.2 Coverage Confirmation

After the coverage results, we fixed a plan before training that compares the wider and the original design at 1928 sources with two new seeds on two sets not evaluated before: nine Tanks and Temples sources (OOD-Water-2) and four registered segments of a DeepSea-ROV dive outside the adaptation data. As shown in Table 22, all three primary rules held: the new-dive P interval lies above zero with a mean of at least 0.3 dB, OOD-Water-2 W stays within ± 0.3 dB or its interval includes zero, and OOD-Water-2 P gains at most 0.3 dB. The dive clears its rule narrowly: the second seed alone gives $+0.84$ dB with an interval of $[-0.21, +1.88]$, which includes zero. Two secondary predictions were not met: W falls on the new dive, and in a one-seed YoNoSplat-W control more sources help Real25 W but wider coverage does not. A secondary rule on 20 coral clips from a public reef survey, fixed after the dive result but before these checkpoints were evaluated on a set that earlier models had already been scored on, predicted a P gain of 0.3 to 1.5 dB with an interval above zero and a W change within ± 0.5 dB, and it also held.

Table 22: **Coverage confirmation: wider minus original coverage at 1928 sources.** Two new seeds averaged per source. PSNR in dB over its 95% paired interval; *: the interval excludes zero.

Set	Output	ΔPSNR↑	ΔSSIM↑	ΔLPIPS↓
New dive (4 segments)	W	-0.41^{*} $[-0.93, -0.09]$	-0.020^{*}	$+0.002$
	P	$+0.94^{*}$ $[+0.01, +1.87]$	-0.076^{*}	-0.247^{*}
OOD-Water-2 (9 sources)	W	-0.09 $[-0.46, +0.16]$	-0.002	$+0.004$
	P	-0.19 $[-0.61, +0.15]$	-0.012	$+0.006$
Real25, new seeds	W	$+0.04$ $[-0.16, +0.27]$	$+0.006$	-0.003
	P	$+1.08^{*}$ $[+0.56, +1.65]$	-0.017^{*}	-0.072^{*}
Coral clips (20)	W	-0.06 $[-0.20, +0.07]$	-0.002^{*}	-0.006^{*}
	P	$+0.39^{*}$ $[+0.21, +0.57]$	-0.014^{*}	-0.095^{*}

F Additional Visual Results

The examples come from a bank of twelve scenes, two per synthetic set and real collection, fixed before any score or image was viewed; the rows were chosen from it by visual inspection, and all twelve scenes will be released with the code. Figure 4 uses the same bank and shows YoNoSplat-R, which has the higher PSNR of the two YoNoSplat models on each of its rows. All views are held out and shown without colour adjustment.

Restored views. Figure 12 adds one restored view with a clean reference from each synthetic set. C recovers the layout and colours of the Test32 and Mip360-Water scenes, where AnySplat distorts the geometry and YoNoSplat-R blurs the plants; on the hard OOD-Water view every model loses detail, and C keeps the layout of the room.

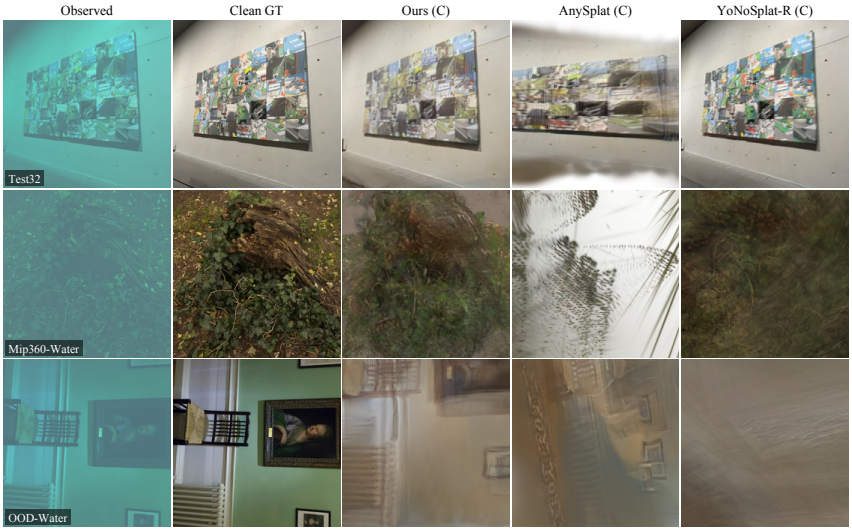

Figure 12: **Additional restored views with a clean reference.** All columns render the same held-out camera; the observed underwater view is shown for orientation and is not an input.

Real footage. Figure 13 shows all three outputs at one held-out view of each real collection. W and P keep structure that YoNoSplat-R blurs, and P re-renders C through the predicted medium; without a clean reference the restored colours cannot be verified.

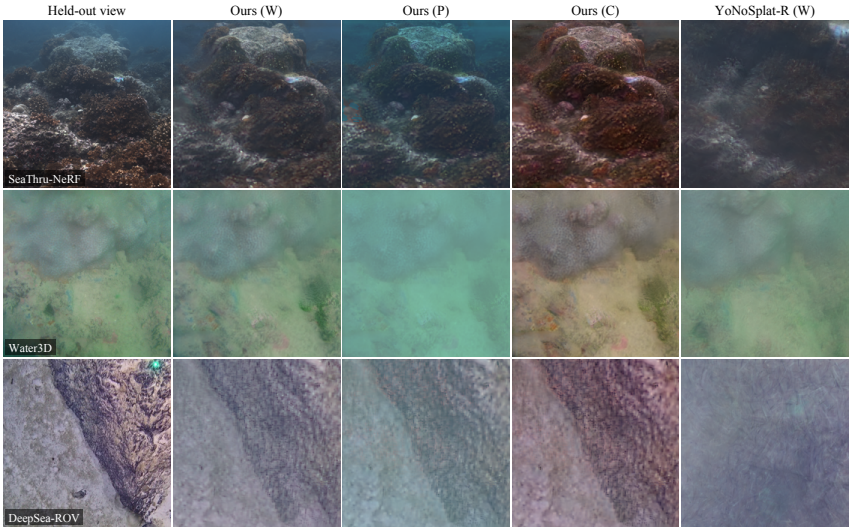

Figure 13: **In-water, physical and restored outputs on real held-out views.** W is rendered directly, P applies the predicted medium to C, and real footage has no clean reference for C.

G LIMITATIONS

Evidence. Most controls use one seed, and the paired intervals condition on fixed checkpoints and few acquisitions. No experiment isolates the depth of the split, shared versus independent point support, or model size, and the data-effects study does not establish a general scaling law. No matched feed-forward comparison tests option (i) of Section 3.1. Upstream exposure of Test32 cannot be verified (see Appendix A.2), and the stop-gradient (chosen in a pilot scored on Real25), the optical-depth augmentation and some study designs were informed by earlier test results.

Model. The medium is one constant vector per scene, so spatially varying water such as moving vehicle light is not modelled, and its coefficients are effective values in our range gauge rather than measured water properties. Scenes are assumed static, so moving objects such as fish are not modelled. Real footage has no clean reference, so real restoration is assessed only qualitatively. Physical re-rendering does not repair reconstruction errors: in hard views, fine structure can be lost and the layout distorted.

Comparisons. The per-scene optimisers start from a predicted initialisation on five dense sequences. With cameras and points from structure from motion they are far more accurate, and our model cannot yet take such cameras as input.