

UnderwaterEVER: Physics-Informed Underwater Novel View Synthesis via Exact Volumetric Rendering

Final project Digital Image Processing (2360860) – Winter 25-26

Merav Keidar

1 Introduction

1.1 Novel view synthesis (NVS)

Novel view synthesis (NVS) is the task of generating unseen views of a 3D scene from a collection of ground truth (GT) 2D views of the scene. In recent years, this problem has been revolutionized by two paradigms:

1. **Neural Radiance Fields – NeRF** [1] These use an MLP to represent a scene. The MLP’s input is the coordinates of a point, and the output is the density and radiance of the point in that scene. New views can be rendered by ray projection. This method provides high-quality volumetric rendering, and enjoys the benefits of ray tracing (which is a well known rich rendering method in the field of computer graphics).
2. **3D Gaussian Splatting – 3DGS** [2] This represents the scene as a collection of 3D Gaussians. The initial Gaussians are created using SFM (structure from motion) [3]. The Gaussians are later optimized by “splatting” them to 2D Gaussians, rasterizing them to create a 2D image, and using a gradient descent algorithm with a photometric loss between the rendered image and the GT. The rasterization rendering is much faster than ray racing, allowing 3DGS to achieve real-time rendering speeds.

Overall, both methods achieve impressive results, with 3DGS being faster than NeRF, but more limited and prone to artifacts. This is because rasterization is faster than ray tracing but less flexible – ray tracing doesn’t allow volumetric rendering and complex camera effects, such as rolling shutter, fish eye camera and more.

1.2 Underwater Imaging

The underwater computer vision realm suffers from challenges that do not exist above water. Standard NVS assumes light travels through a vacuum, but underwater this is not the case. While there are many formulations for modeling underwater imaging, [4–6] they all consider the following two effects:

1. **Wavelength-Dependent Attenuation:** Red light is absorbed significantly faster than blue light, causing severe color shifts.
2. **Backscattering:** Light reflects off suspended particles (marine snow), creating a distance-dependent “veil” or haze that obscures geometry.

2 Chosen Papers

I chose to base my project on two different papers from the realm of NVS. The first, SeaThru-Nerf [7] is a variant of NeRF [1], specifically built for the underwater environment. The Second, EVER [8], is a 3DGS [2] based method, allowing new capabilities such as volumetric rendering.

2.1 SeaThru-NeRF

SeaThru-NeRF [7] extends the NeRF framework [1] to handle underwater environments by incorporating a physics-based image formation model into the rendering process. Standard NeRF assumes that light traverses the scene unaffected by the media, which is violated in underwater environments. SeaThru-NeRF addresses this by integrating physics directly into the volumetric rendering equation.

Physical model. The starting point is the revised SeaThru image formation model [6] [9]. For a scene point at depth z and per color channel, the camera observes:

$$I(z) = \underbrace{J \cdot e^{-\beta^D \cdot z}}_{\text{attenuated object}} + \underbrace{B^\infty \cdot (1 - e^{-\beta^B \cdot z})}_{\text{backscatter}}, \quad (1)$$

where J is the clean object radiance, B^∞ is the veiling light (the backscatter color at infinite depth), and β^D , β^B are wavelength-dependent attenuation and backscatter coefficients. The revised model treats β^D and β^B as functions of the medium’s physical properties (ambient illumination, sensor spectral response, scattering and beam attenuation), allowing it to correctly separate the two effects.

Integration into NeRF. To embed Eq. 1 into volumetric rendering, SeaThru-NeRF discretizes each ray into N intervals; interval i starts at depth s_i and has width δ_i . Each interval is assigned *two separate densities*: σ_i^{obj} for scene geometry and σ^{med} for the scattering medium. The object transmittance T_i^{obj} — the probability that light reaches interval i without hitting an object — is computed from object density only, so the semi-transparent water does not occlude scene content:

$$T_i^{\text{obj}} = \exp\left(-\sum_{j=0}^{i-1} \sigma_j^{\text{obj}} \delta_j\right). \quad (2)$$

The two components of Eq. 1 map directly onto two per-interval color contributions:

$$\hat{C}_i^{\text{obj}} = T_i^{\text{obj}} \cdot e^{-\sigma^{\text{attn}} s_i} \cdot (1 - e^{-\sigma_i^{\text{obj}} \delta_i}) \cdot \mathbf{c}_i^{\text{obj}}, \quad (3)$$

$$\hat{C}_i^{\text{med}} = T_i^{\text{obj}} \cdot e^{-\sigma^{\text{bs}} s_i} \cdot (1 - e^{-\sigma^{\text{med}} \delta_i}) \cdot \mathbf{c}^{\text{med}}, \quad (4)$$

where $\sigma^{\text{attn}}, \sigma^{\text{bs}} \in \mathbb{R}^3$ are learnable per-channel coefficients (predicted by a view-direction MLP, constant per ray) corresponding to β^D and β^B , and $\mathbf{c}^{\text{med}}$ corresponds to B^∞ . The total rendered color is $\hat{C}(\mathbf{r}) = \sum_i \hat{C}_i^{\text{obj}} + \sum_i \hat{C}_i^{\text{med}}$, which jointly models scene appearance and water properties end-to-end.

Importance and influence. SeaThru-NeRF is the first work to bridge physics-based underwater imaging and neural rendering. Being the first method to do so, they also established a standard benchmark for underwater NVS, and introduced the SeaThru-NeRF dataset.

The paper demonstrates that incorporating domain-specific physics into NeRF significantly improves reconstruction quality in challenging environments. It contributes to the growing line of work on physics-informed neural rendering, demonstrating the benefits of incorporating explicit domain knowledge into the NeRF pipeline.

Critical review. SeaThru-NeRF has a few limitations. Firstly, like standard NeRF, it requires long training times (hundreds of thousands of iterations), making it impractical for real-time or large-scale applications. The integration of physics increases the number of parameters and makes optimization more difficult and even longer.

On top of that, the paper makes some simplifying assumptions - it assumes a homogeneous medium, which does not fully capture all underwater environments.

2.2 EVER: Exact Volumetric Ellipsoid Rendering

EVER [8] introduces a new rendering framework that improves upon 3DGS by enabling exact volumetric rendering. While 3DGS achieves real-time performance through rasterization, it relies on approximations that make it less physically accurate and less flexible than ray-tracing-based rendering approaches.

Recently, 3DGRT [10] introduced a ray-tracing-based rendering pipeline for Gaussian primitives, leveraging hardware acceleration to achieve near real-time performance. This shift from rasterization to ray tracing enables more flexible rendering effects. However, similar to 3DGS, these approaches still rely on approximations, and do not allow exact volumetric rendering, causing artifacts such as inconsistent blending and “popping” when the viewpoint changes.

EVER addresses these artifacts by removing the reliance on the approximations and introducing **exact volumetric rendering**. They achieve that by replacing Gaussian primitives with constant-density ellipsoids and deriving a closed-form solution to the volume rendering integral, obtained by computing the exact entry and exit points of each primitive along a ray. The rendered color of a ray is:

$$\mathbf{C} = \sum_k T_k \alpha_k \mathbf{c}_k, \quad T_k = \exp\left(-\sum_{j < k} \rho_j \Delta t_j\right), \quad \alpha_k = 1 - e^{-\rho_k \Delta t_k}, \quad (5)$$

where segment k spans $[t_a^k, t_b^k]$, $\Delta t_k = t_b^k - t_a^k$, ρ_k is the ellipsoid density, $\mathbf{c}_k$ is its color, α_k is the segment opacity, and T_k is the accumulated transmittance — the fraction of light that survives all segments before k . Because EVER evaluates these quantities analytically rather than via numerical approximation, it accumulates density and color contributions exactly.

This approach allows accurate blending of overlapping primitives and ensures physically consistent rendering.

Importance and influence. By performing exact volumetric integration, EVER ensures view-consistent rendering and effectively eliminates “popping” artifacts caused by incorrect primitive ordering in 3DGS.

On top of that, EVER achieves improved SSIM and LPIPS compared to 3DGS-based and even NeRF methods, and demonstrates state-of-the-art performance on Zip-NeRF [11].

The method narrows the gap between fast rasterization-based approaches and physically accurate volumetric rendering. This makes it particularly relevant for applications where both speed and accuracy are required.

EVER represents a meaningful step in the evolution of 3DGS-based methods, pushing the field toward more principled and physically grounded rendering techniques.

Critical review. Despite its advantages, EVER introduces additional computational and implementation complexity. Although it achieves real-time performance, it is slower than standard 3DGS due to the use of ray tracing and the need to construct and traverse bounding volume hierarchies (BVHs) for ray-primitive intersection.

The method also requires increased memory compared to 3DGS, as reported in Table 1 (page 6) of [8], partly due to differences in scene representation and densification behavior.

Overall, EVER significantly improves the physical correctness and consistency of 3DGS-based rendering, but at the cost of increased complexity and somewhat reduced efficiency.

3 Proposed Method: UnderwaterEVER

UnderwaterEVER combines EVER’s exact volumetric renderer with a differentiable underwater physics model embedded inside the per-segment rendering integral. The goal is to bridge the

gap between the fast but physics-agnostic EVER approach [8] and physically accurate but slow SeaThru-NeRF [7].

Our method is implemented on top of the EVER pipeline. Namely, we represent the scene as a collection of constant density ellipsoids.

3.1 Physics-Aware Segment Rendering

Recall from Eq. 5 that EVER accumulates per-segment contributions. For a single segment spanning $[t_a, t_b]$ with density ρ , color $\mathbf{c}$, opacity $\alpha = 1 - e^{-\rho\Delta t}$, and entry transmittance T_a , the contribution is:

$$\Delta\mathbf{C} = T_a \cdot \alpha \cdot \mathbf{c}. \quad (6)$$

We replace this with two physics-aware terms, one for attenuated object color and one for backscatter, mirroring the physical decomposition of Eq. 1.

Attenuated object color. Water attenuates the object radiance by $e^{-\sigma_D t}$ along the line of sight, where $\sigma_D \in \mathbb{R}_+^3$ is a per-channel attenuation coefficient. Integrating this decay analytically over $[t_a, t_b]$ gives:

$$\Delta\mathbf{C}_{\text{obj}} = T_a \cdot \mathbf{c} \odot \frac{\rho}{\rho + \sigma_D} \odot e^{-\sigma_D t_a} \odot (1 - e^{-(\rho + \sigma_D)\Delta t}), \quad (7)$$

where $\odot$ is element-wise multiplication. When $\sigma_D \rightarrow \mathbf{0}$, Eq. 7 reduces exactly to Eq. 6.

Backscatter. Suspended particles scatter ambient light toward the camera, adding a depth-dependent veil. The backscatter contribution from $[t_a, t_b]$ is:

$$\Delta\mathbf{C}_{\text{bs}} = T_a \cdot \mathbf{c}_{\text{med}} \odot \left[\alpha - \frac{\rho}{\rho + \sigma_B} \odot e^{-\sigma_B t_a} \odot (1 - e^{-(\rho + \sigma_B)\Delta t}) \right], \quad (8)$$

where $\sigma_B \in \mathbb{R}_+^3$ is the backscatter coefficient and $\mathbf{c}_{\text{med}} \in [0, 1]^3$ is the veiling-light color. When $\sigma_B \rightarrow \mathbf{0}$ the bracket vanishes and backscatter disappears.

The total per-segment contribution is $\Delta\mathbf{C} = \Delta\mathbf{C}_{\text{obj}} + \Delta\mathbf{C}_{\text{bs}}$, replacing Eq. 6 inside EVER’s kernel. All quantities t_a , Δt , ρ , $\mathbf{c}$, T_a , α are already available in the kernel. The only new inputs are σ_D , σ_B , and $\mathbf{c}_{\text{med}}$, described in the next subsection.

3.2 WaterGlobal

We introduce a 9-scalar module, *WaterGlobal*, to store the water parameters $(\sigma_D, \sigma_B, \mathbf{c}_{\text{med}})$ across all rays and views. They are computed as follows:

$$\sigma_D = \max(\text{softplus}(\mathbf{w}_D), 0.1), \quad \sigma_B = \text{softplus}(\mathbf{w}_B), \quad \mathbf{c}_{\text{med}} = \text{sigmoid}(\mathbf{w}_c), \quad (9)$$

where $\mathbf{w}_D, \mathbf{w}_B, \mathbf{w}_c \in \mathbb{R}^3$ are raw learned parameters. The hard floor of 0.1 on σ_D reflects the physical fact that real seawater always has non-negligible absorption, preventing collapse to zero before the scene geometry is established. Scene-global parameters are the right choice for a homogeneous medium, where water properties do not depend on viewpoint.

Initialization. Under water, red light is absorbed more strongly than green, which is absorbed more strongly than blue, therefore we want:

$$\sigma_{D,R} > \sigma_{D,G} > \sigma_{D,B}$$

To this end, we initialize σ_D from the mean per-channel intensity of the training images, $\bar{I}_c$, which reflects this absorption, as follows:

$$\sigma_{D,c}^{(0)} = \sigma_{\text{base}} \cdot \frac{\bar{I}_B}{\bar{I}_c}, \quad c \in \{R, G, B\}, \quad (10)$$

with $\sigma_{\text{base}} = 0.05$.

3.3 Training

WaterGlobal is optimized jointly with the scene using a separate Adam optimizer ($\text{lr} = 10^{-3}$). We add two physics-grounding loss terms to the standard EVER reconstruction loss:

$$\mathcal{L}_{\text{EVER}} + \underbrace{\lambda_{\text{trans}} \mathcal{L}_{\text{trans}} + \lambda_{\text{spec}} \mathcal{L}_{\text{spec}}}_{\text{our additions}}. \quad (11)$$

Transparency regularizer $\mathcal{L}_{\text{trans}}$. Prevents σ_D from collapsing to zero. Let $T = e^{-\bar{\sigma}_D D}$ be the per-pixel transmittance, where $\bar{\sigma}_D$ is the channel-mean attenuation and D the rendered depth. The loss pushes T away from both extremes:

$$\mathcal{L}_{\text{trans}} = -\mathbb{E} \left[\log \left(\lambda_f e^{-|1-T|/\tau} + e^{-|T|/\tau} \right) \right], \quad (12)$$

with $\lambda_{\text{trans}} = 5 \times 10^{-4}$, $\lambda_f = 0.1$, $\tau = 0.1$.

Spectral ordering loss $\mathcal{L}_{\text{spec}}$. Penalizes any violation of the seawater ordering $\sigma_{D,R} \geq \sigma_{D,G} \geq \sigma_{D,B}$:

$$\mathcal{L}_{\text{spec}} = \text{ReLU}(\sigma_{D,G} - \sigma_{D,R}) + \text{ReLU}(\sigma_{D,B} - \sigma_{D,G}), \quad (13)$$

with $\lambda_{\text{spec}} = 0.01$.

3.4 Scene Restoration

As a byproduct, rendering with $\sigma_D = \sigma_B = \mathbf{0}$ at inference produces dewatered images with corrected color balance and reduced backscatter haze, without any explicit clean-water supervision. Figure 1 shows an example on the Curaçao scene.

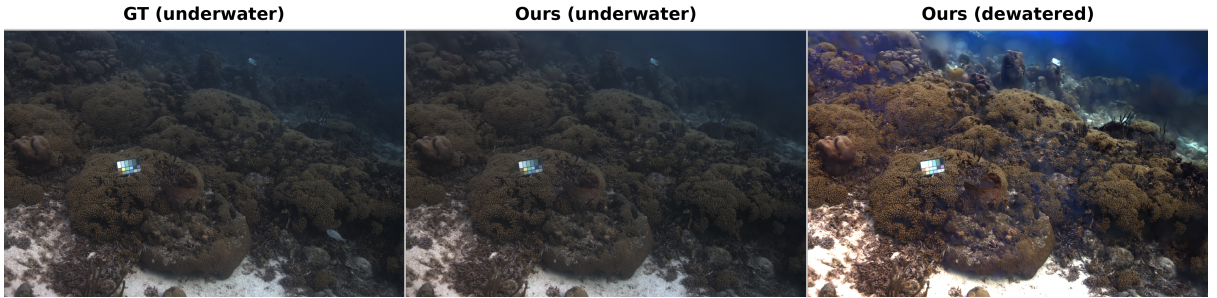


Figure 1: Scene restoration on Curaçao. Left: ground-truth underwater image. Center: UnderwaterEVER render. Right: dewatered render.

4 Experiments

We compare our method against two baselines, SeaThru-NeRF, and vanilla EVER.

All experiments use the SeaThru-NeRF dataset [7], which provides four real underwater scenes with COLMAP-reconstructed camera poses: IUI3-RedSea, Curaçao, JapaneseGradens-RedSea, and Panama. We evaluate all four scenes. For each scene, all methods are trained on that scene and evaluated on the official train/test split from the dataset (holding out every 8th image as test).

4.1 Experimental Protocol

- **Training iterations:** EVER and UnderwaterEVER use 30,000 iterations (default EVER setup). SeaThru-NeRF uses its released default, 250,000 steps. This is a standard-settings comparison (not an equal-compute comparison).
- **Metrics:** PSNR (dB), SSIM. Higher is better for both.
- **UnderwaterEVER hyperparameters:** $\text{water_lr} = 10^{-3}$, $\sigma_{\text{base}} = 0.05$ (color-stats init), $\lambda_{\text{trans}} = 5 \times 10^{-4}$, $\lambda_f = 0.1$, $\lambda_{\text{spec}} = 0.01$ (spectral ordering), hard floor $\sigma_D \geq 0.1$.
- **EVER baseline:** Trained with the same EVER schedule as UnderwaterEVER, but without the underwater model (`use_underwater_model=False`).
- **SeaThru-NeRF baseline:** Trained with default settings from the released codebase.

4.2 Quantitative Results

Table 1: Quantitative comparison on test views

Scene	Method	PSNR $\uparrow$	SSIM $\uparrow$
IUI3-RedSea	SeaThru-NeRF	25.619	0.750
	EVER (vanilla)	25.787	0.859
	UnderwaterEVER (ours)	26.805	0.861
Curaçao	SeaThru-NeRF	30.010	0.841
	EVER (vanilla)	29.394	0.890
	UnderwaterEVER (ours)	30.557	0.894
JapaneseGradens-RedSea	SeaThru-NeRF	21.544	0.734
	EVER (vanilla)	22.300	0.875
	UnderwaterEVER (ours)	22.408	0.877
Panama	SeaThru-NeRF	26.919	0.788
	EVER (vanilla)	28.637	0.898
	UnderwaterEVER (ours)	29.091	0.897
Average	SeaThru-NeRF	26.023	0.778
	EVER (vanilla)	26.530	0.881
	UnderwaterEVER (ours)	27.215	0.882

4.3 Qualitative Results

Qualitative results can be seen in Figure 2.

For example, the top row shows a reconstructed scene from the IUI3-RedSea dataset. Observe the image in the pink area: the picture is slightly blurred in the SeaThru-NeRF render, overly dark in the EVER reconstruction, and looks best in our synthesis. This is in line with the quantitative result – our PSNR is indeed the highest.

5 Limitations and Future work

WaterGlobal assumes a single set of water parameters for all views, which is physically correct for a homogeneous medium but problematic when training cameras are spatially disjoint. For example, in the IUI3-RedSea scene, 23 cameras cluster near $z \approx -1$ and 6 cameras are located

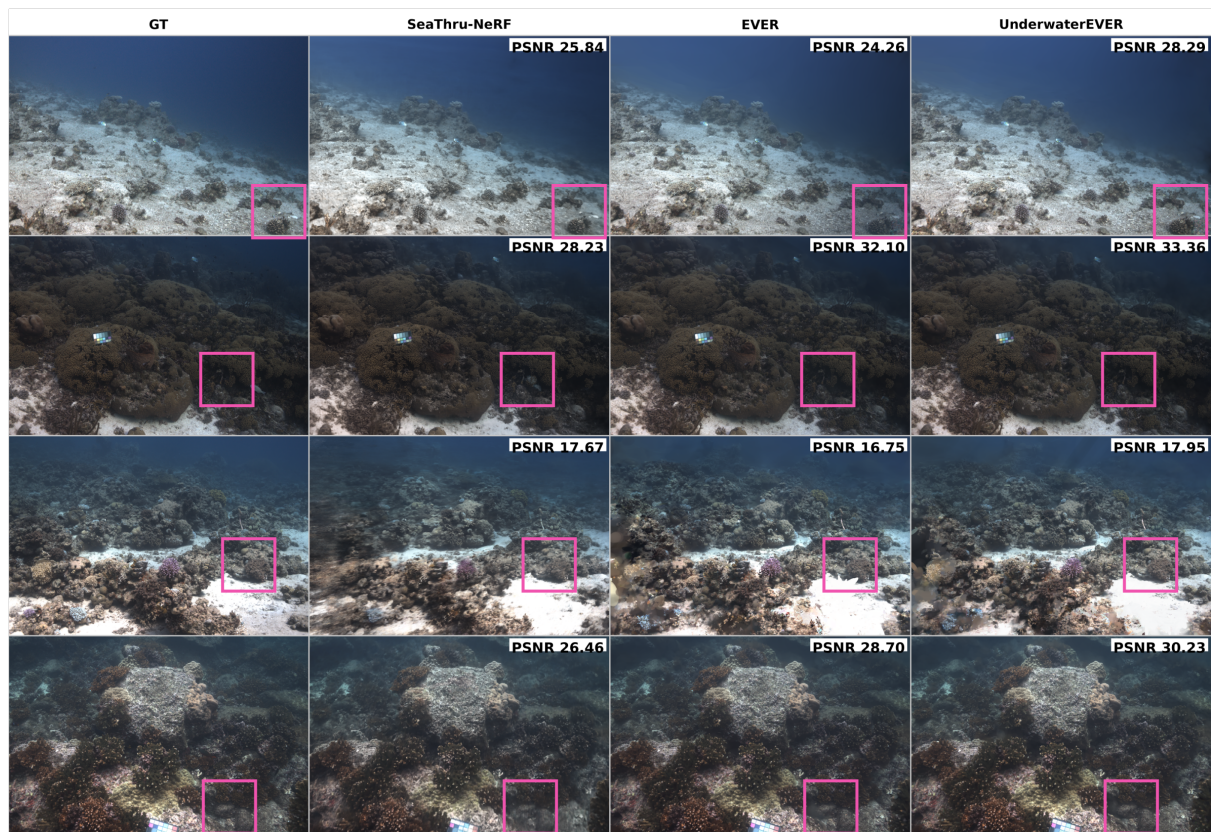


Figure 2: Qualitative Results, each row corresponds to a different dataset. From top to bottom: IUI3-RedSea, Curacao, JapaneseGradens-RedSea, Panama. Interest areas are highlighted with a pink square.

at $z \approx +4.5$. The global σ_D is pulled towards the majority cluster, systematically misfitting the minority. This limitation is shared with SeaThru-Nerf [7].

6 Conclusion

We presented UnderwaterEVER, a new NVS method that combines EVER’s exact volumetric renderer with a differentiable underwater physics model. UnderwaterEVER bridges the gap between the fast but physics-agnostic EVER approach [8] and physically accurate but slow SeaThru-NerF [7], achieving the best of both. Quantitatively, UnderwaterEVER outperforms both baselines on average PSNR and SSIM across all four SeaThru-NerF scenes, with a +0.685 dB PSNR gain over vanilla EVER and a +1.192 dB gain over SeaThru-NerF. As a byproduct, the learned water model enables scene restoration at no extra cost: zeroing the attenuation and backscatter densities at inference yields dewatered, color-corrected novel views without any clean-image supervision.

References

- [1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” 2020.
- [2] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering,” 2023.
- [3] J. Lee and S. Yoo, “Dense-sfm: Structure from motion with dense consistent matching,” 2026.
- [4] J. Jaffe, “Computer modeling and the design of optimal underwater imaging systems,” *IEEE Journal of Oceanic Engineering*, vol. 15, no. 2, pp. 101–111, 1990.
- [5] B. L. McGlamery, “A Computer Model For Underwater Camera Systems,” in *Ocean Optics VI* (S. Q. Duntley, ed.), vol. 0208, pp. 221 – 231, International Society for Optics and Photonics, SPIE, 1980.
- [6] D. Akkaynak and T. Treibitz, “Sea-thru: A method for removing water from underwater images,” pp. 1682–1691, 06 2019.
- [7] D. Levy, A. Peleg, N. Pearl, D. Rosenbaum, D. Akkaynak, S. Korman, and T. Treibitz, “Seathru-nerf: Neural radiance fields in scattering media,” 2023.
- [8] A. Mai, P. Hedman, G. Kopanas, D. Verbin, D. Futschik, Q. Xu, F. Kuester, J. T. Barron, and Y. Zhang, “Ever: Exact volumetric ellipsoid rendering for real-time view synthesis,” 2025.
- [9] D. Akkaynak and T. Treibitz, “A revised underwater image formation model,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6723–6732, 2018.
- [10] N. Moenne-Loccoz, A. Mirzaei, O. Perel, R. de Lutio, J. M. Esturo, G. State, S. Fidler, N. Sharp, and Z. Gojcic, “3d gaussian ray tracing: Fast tracing of particle scenes,” 2024.
- [11] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, “Zip-nerf: Anti-aliased grid-based neural radiance fields,” 2023.