

Aquatic-GS: A Hybrid 3D Representation for Underwater Scenes

Shaohua Liu, Junzhe Lu, Zuoya Gu, Jiajun Li *Member, IEEE* and Yue Deng, *Senior Member, IEEE*

Abstract—Representing underwater 3D scenes is a valuable yet complex task, as attenuation and scattering effects during underwater imaging significantly couple the information of the objects and the water. This coupling presents a significant challenge for existing methods in effectively representing both the objects and the water medium simultaneously. To address this challenge, we propose Aquatic-GS, a hybrid 3D representation approach for underwater scenes that effectively represents both the objects and the water medium. Specifically, we construct a Neural Water Field (NWF) to implicitly model the water parameters, while extending the latest 3D Gaussian Splatting (3DGS) to model the objects explicitly. Both components are integrated through a physics-based underwater image formation model to represent complex underwater scenes. Moreover, to construct more precise scene geometry and details, we design a Depth-Guided Optimization (DGO) mechanism that uses a pseudo-depth map as auxiliary guidance. After optimization, Aquatic-GS enables the rendering of novel underwater viewpoints and supports restoring the true appearance of underwater scenes, as if the water medium were absent. Extensive experiments on both simulated and real-world datasets demonstrate that Aquatic-GS surpasses state-of-the-art underwater 3D representation methods, achieving better rendering quality and real-time rendering performance with a $410\times$ increase in speed. Furthermore, regarding underwater image restoration, Aquatic-GS outperforms representative dewatering methods in color correction, detail recovery, and stability. Our models, code, and datasets can be accessed at <https://aquaticgs.github.io>.

Index Terms—Underwater 3D Representation, 3D Gaussian Splatting, Implicit Modeling, Novel View Synthesis, and Underwater Image Restoration.

I. INTRODUCTION

THE 3D representation of underwater scenes plays a vital role in a wide range of applications, including autonomous underwater vehicles (AUVs) [1], marine ecosystem studies [2], and underwater virtual reality systems [3], [4]. An effective underwater 3D representation should encompass both objects (i.e., the water-free scene) and the water medium. The objects provide critical information about the scene's appearance and geometry, while the water medium conveys essential water parameters. However, achieving this goal remains difficult for vision-based 3D representation methods due to the impact of distance-dependent and wavelength-selective attenuation and scattering inherent in the underwater imaging process [5]. As illustrated in Fig. 1(a), attenuation

S. Liu, J. Lu, Z. Gu, J. Li and Y. Deng are with the School of Astronautics, Beihang University, Beijing 100191, China.

S. Liu is also with Shenyuan Honors College, Beihang University, Beijing 100191, China.

Corresponding author: Yue Deng, ydeng@buaa.edu.cn

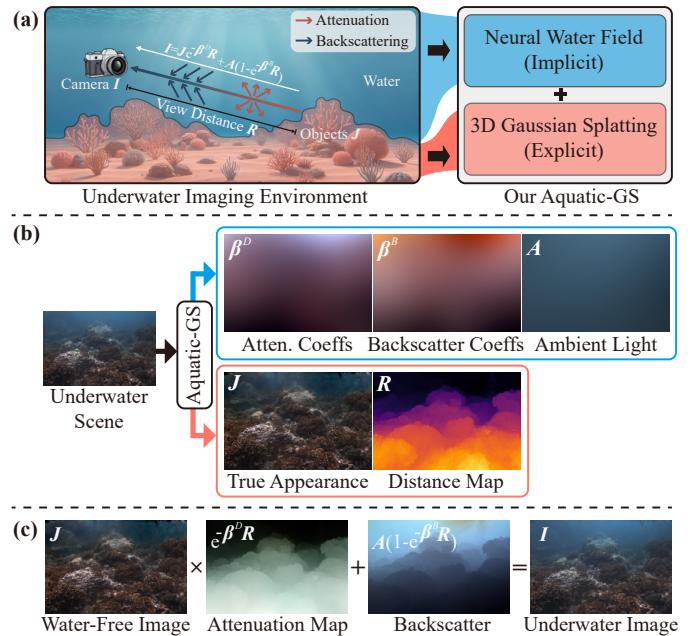


Fig. 1. (a) Underwater imaging environment and the hybrid representation strategy employed by Aquatic-GS. (b) Scene information learned by Aquatic-GS, including water parameters, the true appearance, and the geometry of the underwater scene. 'Atten.' and 'Coeffs' are abbreviations for Attenuation and Coefficients, respectively. (c) Rendering of an underwater image by Aquatic-GS using a physics-based underwater imaging model.

occurs when light reflected from objects is absorbed by the water as it travels toward the camera, with red light being attenuated more than blue and green, leading to a color cast in captured images. Scattering, particularly backscattering, happens when underwater ambient light is scattered toward the camera by particles suspended in the water, leading to a hazy appearance and low contrast in underwater images. Moreover, these two water effects become more pronounced as the distance between the object and the camera increases, leading to a significant coupling of the information from both the objects and the water medium. Such coupling complicates the modeling of both the objects and the water medium, hindering the capture of the true appearance, geometry, and water parameters in underwater scenes [5].

Recently, advanced 3D scene representation methods, such as Neural Radiance Field (NeRF) [6] and 3D Gaussian Splatting (3DGS) [7], have received significant attention. NeRF employs an implicit modeling strategy, typically leveraging neural networks to efficiently encode complex scenes, and

utilizes volume rendering to produce photo-realistic images. In contrast, 3DGS explicitly models the scene with a set of learnable 3D Gaussian primitives and employs a tile-based rasterization pipeline, resulting in superior rendering quality, reduced training cost, and faster rendering speed compared to NeRF. However, in the context of underwater scene representation, both NeRF and 3DGS struggle to effectively represent both the objects and the water medium simultaneously. For instance, some NeRF-based studies [8]–[10] extend their volumetric rendering framework to accommodate the water medium; however, their implicit modeling strategies for objects often lead to blurred details, noisy geometry, and significant rendering costs. Conversely, while the explicit modeling strategy employed by 3DGS adeptly represents opaque objects, it struggles with the abundant translucent water medium [11], leading to inevitable artifacts and inaccurate geometry when applied to underwater scenes. Thus, in underwater scene representation, how to effectively represent both objects and the water medium simultaneously remains a significant challenge.

To address this challenge, we propose a hybrid 3D representation approach called Aquatic-GS, which combines the advantages of explicit representation and implicit modeling to effectively capture the characteristics of both the objects and the water medium in underwater scenes, as shown in Fig. 1. Specifically, considering the spatial non-uniformity of water properties in real environments [12]–[14], we designed a Neural Water Field (NWF) to model the distributions of the water parameters implicitly. Simultaneously, motivated by the efficiency of 3DGS in representing objects, we utilize 3D Gaussians to explicitly capture the true appearance and geometry of the scene. Both components are integrated through a physics-based underwater image formation (UIF) model to represent the underwater scene. Moreover, to tackle the issues caused by water effects in accurately representing the scene, we introduce a Depth-Guided Optimization (DGO) mechanism. The DGO mechanism employs pseudo-depth maps generated by the latest monocular depth estimators like DepthAnything [15] to guide Aquatic-GS in achieving a more precise representation of the scene’s geometry and distant details. Once optimized, Aquatic-GS learns the true appearance, geometry, and distributions of water parameters in underwater scenes. This not only enables real-time rendering of novel underwater viewpoints but also supports restoring the underwater scene as if the water medium were absent.

We evaluated Aquatic-GS’s performance in underwater novel view synthesis (NVS) and underwater image restoration (UIR) on three real-world underwater datasets and our simulated dataset. For the NVS task, Aquatic-GS outperforms state-of-the-art (SOTA) NeRF-based underwater representation methods, achieving superior rendering quality, reduced training time, and a $410\times$ increase in rendering speed for real-time performance. For the UIR task, Aquatic-GS also surpasses representative dewatering methods in color correction, detail recovery, and stability.

Our contributions can be summarized as follows:

- We propose Aquatic-GS, a novel 3D representation method for underwater scenes that implicitly models water parameters and explicitly models the scene’s ap-

pearance and geometry. This hybrid setup effectively represents both the water medium and the objects, ensuring a comprehensive depiction of complex underwater scenes.

- We extend the latest 3DGS to underwater environments and introduce a Depth-Guided Optimization mechanism to tackle the obstacles posed by distance-dependent water effects, achieving better geometry details.
- Extensive experiments on underwater novel view synthesis and image restoration tasks demonstrate the effectiveness of Aquatic-GS for underwater scene representation.

The rest of this paper is structured as follows: Section II covers the related work. In Section III, we provide a detailed description of the proposed Aquatic-GS. Section IV presents extensive experimental results. Finally, Section V concludes the paper.

II. RELATED WORK

A. NeRF-Based Underwater Representation Methods

NeRF is a volumetric scene representation method that utilizes a deep neural network to encode the radiance field. Its implicit representation strategy has the advantage of efficiently compressing complex scenes. NeRF typically queries the scene’s density and color at any given point through a neural network and employs volumetric rendering to produce remarkably photorealistic images. Furthermore, it has been widely applied to various types of scenes, including unbounded [16], [17], in-the-wild [18], large-scale [19], and scattering environments [8]–[10], [20].

For the 3D representation of underwater scenes, NeRF-based methods typically extend their volumetric rendering framework to accommodate the water medium. NeuralSea [8] incorporating different light-transmitting physics for the water medium and the objects to represent underwater 3D scenes. SeaThru-NeRF [9] assigns colors and densities for objects and water at each spatial point, utilizing a scattering-aware volumetric rendering framework to simulate attenuation and scattering, ultimately supporting the rendering of new viewpoints for underwater scenes and water-free scenes. Tang et al. [10] treat water as a semi-transparent object with uniform density and color throughout the space, and jointly optimize its parameters with those of the scene objects. However, the implicit modeling of objects in existing methods often results in blurred details, noisy geometry, and considerable computational overhead, including lengthy training times and slow rendering speeds. In contrast, our Aquatic-GS introduces a hybrid modeling strategy. Unlike these methods that consider the water medium from a volumetric rendering perspective, we view the water medium from the perspective of underwater imaging models. We leverage the advantages of implicit modeling to learn the spatial distribution of water parameters, and we extend the advanced 3DGS to explicitly represent the objects, thereby enabling more accurate geometry, efficient training, and real-time rendering.

B. 3D Gaussian Splatting

3DGS [7] represents a significant breakthrough in 3D scene representation and has attracted considerable attention for its

superior high-fidelity rendering quality and real-time rendering capabilities compared to NeRF. Subsequent research has extended 3DGS to various challenging scenarios, including unconstrained photo collections [21], sparse views [22]–[25], blurry image collections [26], [27], large-scale scenes [28], and dynamic scenes [29]. Several approaches introduce edge [30], frequency [31], and effective rank [32] regularization to address over-reconstruction issues, while others [33]–[35] focus on improving over-reconstructed Gaussian identification for finer-grained representation. Methods like SparseGS [23] and DNGaussian [24] use monocular depth maps for depth supervision, mitigating overfitting in sparse-view scenarios. Other methods incorporate optical flow [25] and normal priors [36] to further enhance 3DGS’s geometric representation. Additionally, Chen et al. [26] and Lee et al. [27] model the formation of motion and defocus blur, enabling the reconstruction of sharp radiance fields from blurry image collections. Despite these advancements, 3DGS’s explicit modeling strategy limits its effectiveness in underwater scene representation and restoration [37], [38]. In this work, we extend 3DGS to underwater environments by integrating a neural water field and a physics-based underwater image formation model. Our Aquatic-GS accurately captures the scene’s true appearance, geometry, and water parameters, enabling novel underwater viewpoint synthesis and supporting reliable scene restoration.

C. Underwater Image Restoration

The goal of underwater image restoration is to correct the color cast and low contrast caused by water, obtaining the true appearance of underwater scenes. Existing underwater image restoration methods can be broadly classified into four categories: visual prior-based, data-driven, physics-based, and NeRF-based approaches.

Visual prior-based methods [39]–[41] typically adjust pixel values to enhance underwater images but often overlook the underwater imaging mechanism. Data-driven approaches use supervised learning with synthetic datasets [42]–[44], GANs [45], or contrastive learning [46] to map underwater images to in-air distributions. Physics-based methodss [5], [14], [47]–[51] consider underwater image formation models to estimate parameters and reverse the degradation process. These methods often incorporate priors, such as underwater dark channel [47], and haze-lines [50], to facilitate parameter estimation, while some [5], [14] leverage multi-view observations to enhance parameter estimation by constructing 3D structures. For instance, SUCRe [5] employs the 3D structure to track color changes of a point from different viewpoints, resulting in more accurate parameter estimation. However, most rely on simplified image formation models that assume water parameters are globally uniform within each channel, which may lead to instability in practical applications [12], [43]. NeRF-based methods [8]–[10], [52], like SeaThru-NeRF [9], typically extend volumetric rendering framework to accommodate the water medium and learn the water-free scenes. However, their reliance on implicit modeling results in blurred details, high training costs, and considerably slow rendering speed. In contrast, our Aquatic-GS extends the latest 3DGS to learn the

water-free scene, while integrating a neural water field to learn the distribution of water parameters. This integration better characterizes complex underwater environments, decouples water effects from the true appearance of underwater scenes, and enables reliable underwater image restoration with lower training costs and real-time rendering performance.

III. METHODOLOGY

A. Underwater Image Formation Model

According to the physics-based underwater image formation (UIF) model proposed by Akkaynak et al. [13], the relationship between the captured underwater image and the true appearance of the scene can be expressed as follows:

$$\mathbf{I} = \mathbf{J} e^{-\beta^D \mathbf{R}} + \mathbf{A} (1 - e^{-\beta^B \mathbf{R}}), \quad (1)$$

where $\mathbf{R} \in \mathbb{R}^{h \times w \times 1}$ represents the distance between the objects and the camera corresponding to each pixel in the captured image $\mathbf{I} \in \mathbb{R}^{h \times w \times 3}$, with h , w and 3 denoting the height, width, and channels of the image, respectively. $\mathbf{J} \in \mathbb{R}^{h \times w \times 3}$, the water-free image, represents the image of objects that the camera would capture in the absence of the water medium, reflecting the true appearance of the underwater scene. $\mathbf{A} \in \mathbb{R}^{h \times w \times 3}$ denotes the underwater ambient light, which can be considered as the backscatter water color at infinity. $\beta^D \in \mathbb{R}^{h \times w \times 3}$ and $\beta^B \in \mathbb{R}^{h \times w \times 3}$ represent the attenuation coefficients and backscatter coefficients of the water medium, respectively, with three channels modeling the wavelength selectivity of these effects.

Typically, the water parameters \mathbf{A} , β^D and β^B are assumed to be uniform within each channel in the scene, while being different across channels, simplifying the UIF model to nine unknowns [5], [49]. However, some studies suggest that this simplified version may not fully account for factors such as solar directionality [12], viewing angles [13], and local water composition [14], among others, which can lead to instabilities and visually unpleasing results [43]. Therefore, in our work, we relax the assumption of uniformity within each channel and attempt to learn the spatially non-uniform distributions of these parameters across each channel.

B. Overview of Aquatic-GS

The pipeline of Aquatic-GS is illustrated in Fig. 2. The distance-dependent water effects cause exponential decay of object information in underwater images as distance increases. Therefore, we set a bounded viewing frustum with a maximum visible distance of r_{max} for each camera to focus on the scene representation within the visible range. For a given viewpoint, we utilize the neural water field to output the water parameters \mathbf{A} , β^D , and β^B . Simultaneously, we employ 3DGS to render the water-free image \mathbf{J} and corresponding geometric information of the scene, such as depth map \mathbf{D} and distance map \mathbf{R} . As shown in Fig. 2(c), these components are unified through the physics-based UIF model described in Eq.(1),

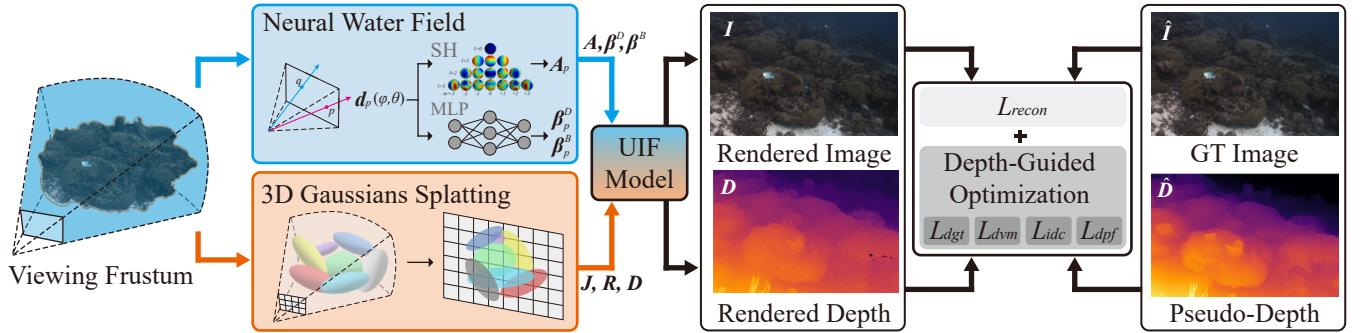


Fig. 2. Pipeline of Aquatic-GS. For a given viewpoint within a bounded viewing frustum, Aquatic-GS uses the Neural Water Field to obtain the underwater ambient light \mathbf{A}_p , attenuation coefficients β_p^D , and backscattering coefficients β_p^B for the current viewpoint, while utilizing 3D Gaussian Splatting to render the water-free image \mathbf{J} along with the corresponding depth \mathbf{D} and distance maps \mathbf{R} . These outputs are then unified through a physics-based underwater image formation (UIF) model to render the corresponding underwater image \mathbf{I} . During the optimization, along with the reconstruction loss between \mathbf{I} and $\hat{\mathbf{I}}$, we design a depth-guided optimization mechanism that uses a pseudo-depth map $\hat{\mathbf{D}}$ to guide Aquatic-GS in producing more accurate underwater scene representation.

which computes the water effects and applies them to the water-free image, enabling the rendering of the corresponding underwater image \mathbf{I} . During the optimization, in addition to computing the reconstruction loss between \mathbf{I} and the ground truth underwater image $\hat{\mathbf{I}}$, we design a depth-guided optimization mechanism that uses a pseudo-depth map $\hat{\mathbf{D}}$ to guide Aquatic-GS in generating more accurate underwater scene representation.

In subsequent sections, we will elaborate on the detailed configurations of the neural water field (Section III-C) and the 3DGs (Section III-D). The depth-guided optimization mechanism will be discussed in Section III-E, and the implementation details will be provided in Section III-F.

C. Neural Water Field

In real scenarios, the water effects are influenced by the solar directionality and viewing angles [12], [13]. Thus, we define the distributions of the water parameters within each channel as functions of the viewing direction and design a neural water field to implicitly learn these distributions. As shown in Fig. 2, for a given camera, the viewing direction corresponding to the p -th pixel is denoted as \mathbf{d}_p , and the corresponding underwater ambient light $\mathbf{A}_p \in \mathbb{R}^3$, attenuation coefficient $\beta_p^D \in \mathbb{R}^3$, and backscatter coefficient $\beta_p^B \in \mathbb{R}^3$ for this pixel can be queried from the neural water field. For implementation, we use spherical harmonics (SH) to model the underwater ambient light and a shallow multilayer perceptron (MLP) to model the attenuation and backscatter coefficients. Below, we describe these two modeling strategies in detail.

1) *SH for Underwater Ambient Light*: Spherical harmonics have been widely applied for modeling low-frequency variations in environmental lightings, such as the sky's illumination distribution [53]. In this work, we explore their application in underwater scenes. We use learnable SH coefficients to construct a weighted sum of SH basis functions, approximating the complex distribution of underwater ambient light relative to the viewing direction.

Following the standard practice [54], the underwater ambient light \mathbf{A}_p for a given viewing direction \mathbf{d}_p can be represented as:

$$\mathbf{A}_p = \sum_{l=0}^{l_{max}} \sum_{m=-l}^l \mathbf{k}_l^m Y_l^m(\mathbf{d}_p), \quad (2)$$

where $\{Y_l^m(\cdot)\}_{l:0 \leq l \leq l_{max}}^{m:-l \leq m \leq l}$ denotes the set of SH basis functions, and $\{\mathbf{k}_l^m\}_{l:0 \leq l \leq l_{max}}^{m:-l \leq m \leq l}$ represents the corresponding learnable SH coefficients. Each $\mathbf{k}_l^m \in \mathbb{R}^3$ corresponds to the RGB components of the ambient light. The parameter l indicates the degree of the SH basis function. Notably, each SH basis function defines a distribution on the sphere, with higher l corresponding to more complex distribution and greater variation. The parameter l_{max} represents the highest degree of SH basis functions used in the approximation, with a larger l_{max} incorporating more SH bases, thus improving the capacity to model complex distributions. For computational efficiency, we set $l_{max} = 3$.

2) *MLP for Attenuation and Backscatter Coefficients*: Considering the additional influence of the local composition of the water [14], we employ a shallow MLP to model the complex distributions of β^D and β^B . First, for a given viewing direction \mathbf{d}_p , we convert it into a Cartesian unit vector and apply the positional encoding strategy [6] to obtain a high-dimensional embedding, denoted as $\mathbf{d}'_p \in \mathbb{R}^{27}$. The corresponding β_p^D and β_p^B are then computed as follows:

$$(\beta_p^D, \beta_p^B) = \sigma(F_2(\sigma(F_1(\mathbf{d}')))), \quad (3)$$

where $F_1(\cdot)$ and $F_2(\cdot)$ represent linear layers with output feature dimensions of 128 and 6, respectively, and $\sigma(\cdot)$ is the Softplus activation function.

By inputting all pixels' viewing directions into the neural water field, we can query the corresponding \mathbf{A} , β^D , and β^B for the current viewpoint.

D. 3D Gaussian Splatting

3D Gaussian Splatting employs a collection of 3D Gaussians to represent scenes explicitly. The spatial distribution of

each Gaussian is modeled by the equation:

$$G(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}, \quad (4)$$

where \mathbf{x} is a position in world space, $\boldsymbol{\mu}$ is the mean position of the Gaussian, and $\boldsymbol{\Sigma}$ is the covariance matrix. In practice, $\boldsymbol{\Sigma}$ is computed using a scaling matrix \mathbf{S} and a rotation matrix \mathbf{R} , defined as $\boldsymbol{\Sigma} = \mathbf{R} \mathbf{S} \mathbf{S}^T \mathbf{R}^T$, ensuring positive semi-definiteness. Each Gaussian also includes a set of learnable spherical harmonics coefficients to model view-dependent color and an opacity attribute o that determines its contribution during the blending process.

For efficient rendering, 3DGS utilizes a tile-based differentiable rasterization pipeline. Initially, each 3D Gaussian $G(\mathbf{x})$ is projected onto the image plane as a 2D Gaussian $G'(\mathbf{x})$ [55]. These are then depth-sorted based on their z-axis coordinates in camera space. The pixel color on the image plane is computed using α -blending:

$$c(\mathbf{x}') = \sum_{i=1}^{N_g} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad \alpha_i = o_i G'(\mathbf{x}'), \quad (5)$$

where \mathbf{x}' is the pixel position in image space, c_i and α_i are the color and opacity of the i -th sorted Gaussian at that position, and N_g represents the number of Gaussians involved in the blending. The differentiable rasterization process enables the end-to-end optimization of Gaussian parameters, such as $\boldsymbol{\mu}$, \mathbf{R} , \mathbf{S} , o , and SH coefficients, by calculating and backpropagating the gradients of the reconstruction loss (i.e., L1 loss and DSSIM) between the rendered image and the ground truth.

In our work, 3DGS is employed to model the underwater objects, which correspond to the underlying water-free scene. For a given viewpoint, the RGB image rendered by 3DGS corresponds to the water-free image (i.e., the true appearance of the underwater scene) and is denoted as \mathbf{J} . Alongside \mathbf{J} , we extend the 3DGS to render the accumulated opacity \mathbf{o} , defined as follows:

$$\mathbf{o}(\mathbf{x}') = \sum_{i=1}^{N_g} \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (6)$$

Additionally, the corresponding depth map \mathbf{D} and distance map \mathbf{R} , essential for rendering underwater images, are rendered as follows:

$$\mathbf{D}(\mathbf{x}') = \begin{cases} \frac{\sum_{i=1}^{N_g} d_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j)}{\mathbf{o}(\mathbf{x}')}, & \text{if } \mathbf{o}(\mathbf{x}') > 0 \\ r_{max}, & \text{if } \mathbf{o}(\mathbf{x}') = 0 \end{cases} \quad (7)$$

$$\mathbf{R}(\mathbf{x}') = \begin{cases} \frac{\sum_{i=1}^{N_g} r_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j)}{\mathbf{o}(\mathbf{x}')}, & \text{if } \mathbf{o}(\mathbf{x}') > 0 \\ r_{max}, & \text{if } \mathbf{o}(\mathbf{x}') = 0 \end{cases} \quad (8)$$

Here, d_i is the distance from the i -th Gaussian to the camera along the z-axis in camera space, while r_i is the Euclidean distance between the i -th Gaussian and the camera. We compute \mathbf{D} and \mathbf{R} under two conditions: First, if $\mathbf{o}(\mathbf{x}') = 0$, this indicates that no Gaussians contribute to the blending at pixel \mathbf{x}' within the current viewing frustum. In this case, we set both $\mathbf{D}(\mathbf{x}')$ and $\mathbf{R}(\mathbf{x}')$ to the frustum's radius, r_{max} . Second,

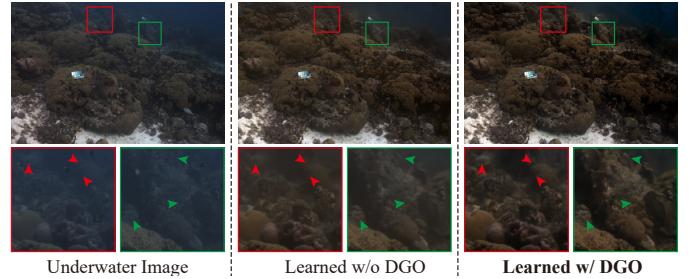


Fig. 3. Visual comparison of the water-free scene learned with reconstruction loss alone versus those learned using our DGO mechanism. The former displays a haze, especially in distant areas, as highlighted by the red and green boxes. The introduction of our DGO mechanism effectively reduces haze and restores more geometry details in these challenging regions.

if $\mathbf{o}(\mathbf{x}') > 0$, we normalize the α -blended depth and distance using $\mathbf{o}(\mathbf{x}')$.

Once we have obtained the water-free image \mathbf{J} , the distance map \mathbf{R} , and the water parameters \mathbf{A} , β^D , and β^B generated by the NWF, we can utilize the UIF model described in Eq.(1) to synthesize the corresponding underwater image \mathbf{I} , enabling the end-to-end optimization of Aquatic-GS.

E. Depth-Guided Optimization Mechanism

Solely using the reconstruction loss does not ensure that Aquatic-GS can effectively decouple the water-free scene from the water effects in underwater images. As shown in Fig. 3, the learned water-free scene using only the reconstruction loss exhibits two main issues: first, the appearance remains hazy, especially in distant areas. Second, in distant, low-contrast areas of the underwater images, the corresponding water-free areas show severe blurring and loss of detail.

These issues primarily stem from the distance-dependent nature of water effects. On one hand, these effects weaken the efficacy of multi-view reconstruction loss in enforcing geometric constraints on Aquatic-GS, especially with increasing distance. Rather than focusing on opaque surfaces, the distribution of 3D Gaussians tends towards a semi-transparent arrangement in space, resulting in a hazy appearance and inaccurate geometry. On the other hand, water effects cause distant areas in underwater images to display significantly weaker texture details and lower contrast than in reality, making it challenging for Aquatic-GS to accurately represent these regions.

To overcome these issues, we have developed a Depth-Guided Optimization (DGO) mechanism, which leverages the pseudo-depth map estimated by DepthAnything [15] to guide Aquatic-GS in accurately representing scene geometry and distant details. This mechanism addresses the issues from four specific aspects: transmittance regularization, minimization of depth variance, coarse-grained depth supervision, and patch-wise frequency domain supervision. Each aspect is implemented through a specifically designed loss function detailed in the following sections.

1) *Depth-Guided Transmittance Regularization*: Ideally, high-opacity Gaussians should be concentrated on object

surfaces, while semi-transparent Gaussians should be minimized [35]. And, for each pixel, the transmittance of Gaussians involved in α -blending tends towards 0 or 1, avoiding intermediate values [56]. To achieve this, we apply transmittance regularization to suppress semi-transparent Gaussians.

First, inspired by [57], we introduce a metric $t(\cdot)$ to assess the presence of semi-transparent Gaussians during the α -blending process for each pixel as follows:

$$t(\mathbf{x}') = \frac{1}{N_g} \sum_{i=1}^{N_g} \{-\log(e^{-10|T_i|} + e^{-10|1-T_i|})\}, \quad (9)$$

where \mathbf{x}' represents the coordinates of any given pixel, and $T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)$ is the transmittance of i -th Gaussian in the pixel's α -blending process. A smaller value of $t(\mathbf{x}')$ indicates that the transmittance values of all the Gaussians contributing to the pixel are closer to either 0 or 1, suggesting fewer semi-transparent Gaussians. We further define a depth-guided transmittance regularization term L_{dgt} , calculated as:

$$L_{dgt} = \frac{1}{HW} \cdot \sum_{\mathbf{x}' \in \Omega} \gamma(\mathbf{x}') t(\mathbf{x}'), \quad (10)$$

where the entire image domain Ω is segmented into a near region $\Omega_n = \{(x, y) | \hat{\mathbf{D}}(x, y) > \tau_1\}$ and a far region $\Omega_f = \{(x, y) | \hat{\mathbf{D}}(x, y) \leq \tau_1\}$ based on the pseudo-depth map $\hat{\mathbf{D}}$. The weight $\gamma(\mathbf{x}')$ is set as $\gamma_n = 1$ for Ω_n and $\gamma_f = 10$ for Ω_f , with γ_f being larger to enhance the suppression of semi-transparent Gaussians in the distant regions.

2) *Depth Variance Minimization Regularization*: For any given pixel, the normalized depth, $\mathbf{D}(\mathbf{x}')$, computed using Eq.(7), represents the weighted average depth of all Gaussians participating in the α -blending at that pixel. The weight, w_i , for the i -th Gaussian is defined as: $w_i = \frac{\alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j)}{\sum_{i=1}^{N_g} \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j)}$. Similarly, the variance of depths, $V_D(\mathbf{x}')$, for all Gaussians involved in the α -blending process for that pixel can be computed as:

$$V_D(\mathbf{x}') = \sum_{i=1}^{N_g} w_i (d_i - \mathbf{D}(\mathbf{x}'))^2. \quad (11)$$

Typically, a smaller value of $V_D(\mathbf{x}')$ implies that the Gaussians contributing to the current pixel are more concentrated. Consequently, we propose a depth variance minimization regularization term, L_{dvm} , to encourage a tighter concentration of Gaussians on object surfaces by minimizing the depth variance for each pixel:

$$L_{dvm} = \frac{1}{HW} \cdot \sum_{\mathbf{x}' \in \Omega} \eta(\mathbf{x}') V_D(\mathbf{x}'), \quad (12)$$

where $\Omega = \Omega_n \cup \Omega_f$ and the weights $\eta(\mathbf{x}')$ are defined as η_n within $\Omega_n \setminus \Omega_e$, η_f within $\Omega_f \setminus \Omega_e$, and η_e within Ω_e . The edge region, Ω_e , is determined by the pseudo-depth map $\hat{\mathbf{D}}$. We assign the weights $\eta_n = 1$, $\eta_f = 0.1$, and $\eta_e = 0.001$ to reflect the varying concentration needs across the image.

This depth-guided weighting strategy is motivated by the observation that pixels corresponding to distant regions involve a wider spatial distribution of 3D Gaussians due to perspective projection. To avoid excessive compression of the depth field

across this wider spatial distribution, smaller weights are applied. Moreover, in regions identified as edges in the pseudo-depth map, where Gaussians exhibit varied depths, the minimal weight is assigned to prevent excessive smoothing and retain geometric details.

3) *Inverse Depth Correlation Loss*: Due to the scale ambiguity inherent in the pseudo-depth maps, which encode normalized relative disparity rather than absolute depth values, conventional loss functions like L1 loss do not work well for depth supervision [25]. Recent work by Xiong et al. [23] demonstrated that normalized cross-correlation can effectively measure the similarity between two maps, regardless of discrepancies in their absolute value ranges. Inspired by this, we propose an inverse depth correlation loss function, L_{idc} , to facilitate effective depth supervision. We first convert our depth map to its inverse form $\mathbf{D}' = 1/(\mathbf{D} + 1)$, and then compute the normalized cross-correlation between \mathbf{D}' and $\hat{\mathbf{D}}$. The loss function L_{idc} is defined as follows:

$$L_{idc} = 1 - Cov(\mathbf{D}', \hat{\mathbf{D}}) / \sqrt{Var(\mathbf{D}') Var(\hat{\mathbf{D}})}, \quad (13)$$

where $Cov(\cdot, \cdot)$ and $Var(\cdot)$ denote covariance and variance, respectively. This approach not only addresses the issue of scale ambiguity but also enhances the geometric fidelity by aligning \mathbf{D}' and $\hat{\mathbf{D}}$ more closely, providing coarse-grained geometric supervision.

4) *Depth-Guided Patch Frequency Loss*: Frequency domain analysis, adept at capturing subtle details and texture variations, has been widely applied across tasks like image generation [58], super-resolution [59], and image fusion [60]. In this study, we propose a depth-guided patch frequency loss L_{dpf} , which utilizes frequency domain insights to enhance texture detail perception in distant, low-contrast areas of underwater images, thereby enabling Aquatic-GS to model these challenging areas more effectively.

We begin by partitioning the rendered underwater image \mathbf{I} , the ground truth underwater image $\hat{\mathbf{I}}$, and the pseudo-depth map $\hat{\mathbf{D}}$ into patches of $k \times k$ pixels using the same grid pattern. This results in K patch triplets, denoted as $\{\mathbf{I}_i, \hat{\mathbf{I}}_i, \hat{\mathbf{D}}_i\}_{i=1}^K$, where $K = \lfloor \frac{h}{k} \rfloor \times \lfloor \frac{w}{k} \rfloor$. We then apply the Discrete Fourier Transform (DFT) to the patches from \mathbf{I} and $\hat{\mathbf{I}}$. For the i -th patch from \mathbf{I} , denoted as \mathbf{I}_i , the DFT is computed as follows:

$$\mathbf{F}_i(u, v) = \sum_{x=0}^{k-1} \sum_{y=0}^{k-1} \mathbf{I}_i(x, y) \cdot e^{-2\pi(\frac{ux}{k} + \frac{vy}{k})}, \quad (14)$$

where (x, y) are the spatial coordinates, $\mathbf{I}_i(x, y)$ represents the pixel value, (u, v) are the frequency coordinates, and $\mathbf{F}_i(u, v)$ is the computed complex frequency value. We calculate the magnitude of \mathbf{F}_i , $|\mathbf{F}_i|$. A similar process is performed for each patch of $\hat{\mathbf{I}}$. We also calculate the average depth in each patch based on $\hat{\mathbf{D}}$, denoted \hat{D}_i^{avg} . The L_{dpf} loss is then computed as:

$$L_{dpf} = \frac{1}{K} \sum_{i=1}^K \psi_i \|\mathbf{F}_i - \hat{\mathbf{F}}_i\|_1, \quad (15)$$

where ψ_i is a depth-dependent weight, and $\|\mathbf{F}_i - \hat{\mathbf{F}}_i\|_1$ represents the L1-norm between the magnitudes of \mathbf{F}_i and $\hat{\mathbf{F}}_i$. Since $\hat{\mathbf{D}}$ corresponds to disparity, ψ_i is defined as $\psi_i =$

$1 - \hat{D}_i^{avg}$, ensuring that patches corresponding to distant areas are assigned greater weight and attention. The final L_{dpf} loss encourages Aquatic-GS to focus on enhancing texture detail representation in these challenging distant areas.

F. Implementation Details

Given M observed images of an underwater scene, we first employ COLMAP [61] to calibrate the extrinsic and intrinsic camera matrices. This process also yields a sparse point cloud, which serves as the initialization for 3D Gaussians. The positions of the cameras in world space are denoted as $\{\mathbf{O}_i\}_{i=1}^M$, and the point cloud is denoted by $\{\mathbf{P}_j\}_{j=1}^{N_p}$, where N_p is the number of points. We then determine the maximum visible distance in the scene as $r_{max} = \lambda \cdot \max_{i,j} \|\mathbf{O}_i - \mathbf{P}_j\|$, with λ being a scale factor empirically set to 2.

Because the surfaces in underwater scenes can often be considered Lambertian reflectors [62], we limit the highest order of SHs used in 3DGS to zero to enable view-independent color rendering. The final loss function used in the optimization is expressed as follows:

$$\begin{aligned} L_{final} = & L_{recon} + \lambda_{dgt} L_{dgt} + \lambda_{dvm} L_{dvm} \\ & + \lambda_{idc} L_{idc} + \lambda_{dpf} L_{dpf}, \end{aligned} \quad (16)$$

where $L_{recon} = (1 - \lambda_{ssim})L_1 + \lambda_{ssim}L_{D-SSIM}$ denotes the reconstruction loss between the rendered and ground truth underwater images, and L_{D-SSIM} and λ_{ssim} denote the D-SSIM loss and its weight, respectively. Additionally, λ_{dgt} , λ_{dvm} , λ_{idc} , and λ_{dpf} are the weights assigned to the respective loss functions within our DGO mechanism. After optimization, for a given viewpoint, Aquatic-GS not only renders the underwater image but also supports the rendering of the corresponding water-free image through the 3DGS branch, achieving underwater image restoration.

IV. EXPERIMENTS

In this section, we demonstrate the effectiveness of our proposed Aquatic-GS through extensive experiments. First, we describe the experimental configurations, including the datasets, implementation details, and evaluation metrics, in Section IV-A. Then, in Section IV-B and Section IV-C, we compare the performance of Aquatic-GS with state-of-the-art methods for underwater novel view synthesis and underwater image restoration, respectively. Finally, in Section IV-D we conduct ablation studies to further analyze the effectiveness of the proposed Aquatic-GS.

A. Experimental Configurations

1) *Datasets*: We employed four distinct datasets to assess the performance of our Aquatic-GS, comprising three datasets from actual underwater environments and one simulated dataset.

SeathruNeRF [9]: This dataset comprises four real underwater scenes: Curacao, Panama, Japanese Gardens, and IUI3, each reflecting diverse aquatic and imaging conditions. The scene image counts are 21, 18, 20, and 29, respectively.

Following [9], these images were downsampled to an average resolution of 900×1400 pixels.

Seathru [49]: We employed the horizontally-viewed D3 and D5 scenes from this dataset, where each scene consists of 68 and 43 images, respectively. Color charts with known patterns are distributed throughout these scenes, serving as ground truth to evaluate Aquatic-GS' effectiveness in underwater image color correction.

In-the-Wild Underwater (IWU) Dataset: We selected two in-the-wild scenes, Coral and Car, from raw, unstructured footage available online to assess Aquatic-GS's performance in general real-world scenarios. For each scene, we sampled frames from the videos, yielding 38 and 54 images, respectively.

Our Simulated Dataset: We constructed three simulated underwater scenes based on Blender, each showcasing distinct challenges: S1 exhibits detailed textures and varied colors; S2, a large-scale environment, displays significant low contrast at distant regions; and S3 shows pronounced greenish color distortions. Each scene contains 50 images with a resolution of 720×1280 . More construction details can be found in the supplementary materials.

For real-world scenes, following [49], we applied white balancing and used COLMAP to obtain camera intrinsics and poses. For simulated scenes, these parameters were exported from Blender and converted to COLMAP format. As our focus is on static underwater scenes, we manually annotated masks to exclude dynamic elements, such as divers in the D5 scene and fish in the Coral scene, to avoid interference.

2) *Implementation Details*: Our implementation is based on the official PyTorch version of 3DGS [7]. We extended its CUDA-based differentiable rasterization pipeline to support the rendering of depth maps, distance maps, and accumulated opacity maps, as well as computations and gradient backpropagation for L_{dgt} and L_{dvm} . During training, the weights in Eq.(16) were $\lambda_{ssim} = 0.2$, $\lambda_{dgt} = 0.0001$, $\lambda_{dvm} = 0.001$, $\lambda_{idc} = 0.1$, and $\lambda_{dpf} = 0.02$. The patch size k used to compute L_{dpf} is set to 256. For the NWF, we set the learning rates for zero-order SH coefficients at 2.5×10^{-3} and higher-order SH coefficients at 1.25×10^{-4} . The MLP within NWF uses an exponential decay learning rate schedule, decreasing from 2×10^{-3} to 2×10^{-5} . All other optimization parameters followed the original 3DGS settings. Inspired by [33], [34], we accumulated the per-pixel absolute gradient norms of μ for finer densification. For the NVS task, one in every eight images was selected for testing, with the rest for training. For the UIR task, all images were utilized for training. Each scene underwent 30,000 iterations on a single NVIDIA RTX 3090 GPU.

3) *Evaluation Metrics*: For the underwater NVS task, we use PSNR, SSIM, and LPIPS metrics to evaluate performance by comparing rendered underwater images of test viewpoints with static ground truth images. We also report the average training time (Avg. Time) and rendering speed (Frames Per Second, FPS) on an RTX 3090 GPU at a resolution of 720×1280 .

For the UIR task, where the water-free reference image is hard to obtain, color charts provide reliable benchmarks for

TABLE I

QUANTITATIVE EVALUATIONS OF THE UNDERWATER NVS TASK ON REAL-WORLD AND SIMULATED DATASETS. FOR EACH DATASET, THE BEST METRIC IS HIGHLIGHTED IN **BOLD** AND THE SECOND-BEST IN UNDERLINE. EFFICIENCY METRICS ARE ALSO INCLUDED.

Methods	Seathru			SeathruNeRF			IWU			Simulated			Speed (FPS)	Avg. Time
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓		
Mip-NeRF 360 [16]	21.707	0.681	0.369	27.030	0.868	0.230	25.765	0.821	0.240	26.721	0.788	0.322	0.13	12.8h
InstantNGP [17]	20.847	0.695	0.376	20.817	0.729	0.381	20.582	0.779	0.240	25.490	0.855	0.213	0.44	4.7h
NeuralSea [8]	-	-	-	25.412	0.709	0.461	-	-	-	28.340	0.751	0.371	0.05	65.8h
SeaThru-NeRF [9]	21.267	0.657	0.409	<u>27.493</u>	<u>0.874</u>	0.226	25.119	0.804	0.268	30.475	0.853	0.222	0.13	13.5h
3DGS [7]	<u>21.764</u>	<u>0.757</u>	<u>0.280</u>	24.295	0.857	<u>0.222</u>	24.527	<u>0.872</u>	<u>0.146</u>	<u>38.243</u>	<u>0.974</u>	<u>0.050</u>	134.03	0.4h
Aquatic-GS	23.193	0.782	0.235	28.125	0.894	0.176	26.664	0.885	0.136	40.008	0.981	0.031	53.64	1.0h

evaluating color correction. Following recent research [12], [45], [63], we use CIEDE2000 (ΔE_{00}) [64] and average angular error $\bar{\psi}$ (in degrees) [5] to measure color fidelity. For simulated scenes with clean images, PSNR, SSIM, and LPIPS assess overall restoration quality. Moreover, visual inspection remains essential alongside quantitative metrics.

B. Performance of Underwater Novel View Synthesis

We compared our method with five state-of-the-art approaches, including general scene 3D representation methods Mip-NeRF 360 [16], InstantNGP [17], and the original 3DGS [7], along with two NeRF-based underwater methods: NeuralSea [8] and SeaThru-NeRF [9]. Due to NeuralSea’s specific scenario requirements, we evaluated it exclusively on the SeaThru-NeRF dataset and our simulated dataset.

The quantitative results are summarized in Table I. Our method consistently demonstrated superior performance across all metrics and datasets, achieving the highest PSNR and SSIM while exhibiting the lowest LPIPS. Notably, the original 3DGS outperformed the NeRF-based approaches overall. On the SeaThru dataset, our method surpassed the second-best 3DGS by 1.429 dB in PSNR. In the SeaThru-NeRF dataset, we improved PSNR by 0.632 dB and SSIM by 0.02 compared to SeaThru-NeRF, while reducing LPIPS by 0.046. In the IWU dataset, Mip-NeRF 360 outperformed 3DGS in PSNR but was still 0.899 dB lower than our method. On the simulated dataset, our method surpassed 3DGS by 1.765 dB in PSNR. The advantages of our approach over the original 3DGS stem from the distinct modeling of water and objects, as well as the DGO mechanism, demonstrating a more effective scene representation.

Visual comparison results are shown in Fig. 4. Mip-NeRF 360 struggles to reconstruct distant details, resulting in blurriness. Conversely, SeaThru-NeRF recovers more details in such areas; however, the substantial noise present in the depth maps indicates numerous floaters, suggesting difficulties in learning the true scene geometry. The original 3DGS also struggles to reconstruct details in distant regions significantly affected by water. Its explicit modeling strategy fails to effectively model the water medium, causing numerous floaters to match the training views and resulting in artifacts and incorrect geometry when rendering from test viewpoints. These floaters

are closer to the camera than the actual scene objects, shown as bright yellow regions in the depth maps. In contrast, our method achieves superior visual quality with reduced blurriness and artifacts, effectively reconstructing more texture details, particularly in challenging, distant regions. Moreover, it generates a more reasonable depth map, characterized by overall smoothness, sharp edges, and fewer floaters, aligning better with underwater object distribution, demonstrating a more effective scene geometry representation.

In terms of efficiency, we achieve $410\times$ faster rendering and reduce training time to 7% of SeaThru-NeRF. Despite the neural water field and DGO mechanism extending training time and slightly reducing speed compared to 3DGS, we maintain real-time rendering at 720×1280 resolution at 53.64 FPS, supporting broad real-time applications.

C. Performance of Underwater Image Restoration

We compared our method with 12 representative underwater image restoration methods, categorized as visual prior-based (Fusion [39], MLLE [40]), data-driven (Water-net [43], TACL [46], TUDA [45], UVEB [44]), physics-based (Seathru [49], Hazeline [50], USUIR [51], SUCRe [5]), and NeRF-based (NeuralSea [8], SeaThru-NeRF [9]) approaches. The latter three leverage multi-view information of the underwater scene for 3D reconstruction, while the rest are primarily implemented based on single images.

In terms of color correction accuracy, we analyzed the color differences between restored and expected color patches in three real-world scenes (D3, D5, and Curacao). Table II shows that our method generally outperforms others in ΔE_{00} and $\bar{\psi}$ across all scenes. Notably, in the Curacao scene, our approach achieved the lowest ΔE_{00} and $\bar{\psi}$ values, reducing ΔE_{00} by 3.3 compared to UVEB and lowering $\bar{\psi}$ by 1.95 degrees relative to SeaThru-NeRF. Moreover, we tracked the hue shifts of the restored red and yellow color patches in the Curacao scene at different observation distances (Fig. 5). Most methods, except UVEB, NeuralSea, and SeaThru-NeRF, oversaturated close-range colors and failed to correct greenish tints at longer distances. While UVEB and SeaThru-NeRF effectively corrected nearby color casts, they struggled with stronger color casts as the patches were farther away. Conversely, our approach consistently delivered effective restoration across



Fig. 4. Qualitative comparison in the underwater NVS task. Alongside the rendered underwater images, corresponding rendered depth maps are also displayed to assess the geometric representation capabilities of different approaches. Challenging regions are highlighted within red boxes. The pseudo-depth maps serve as benchmarks for evaluating geometric fidelity. Aquatic-GS demonstrates superior rendering quality, effectively reducing blur and artifacts, while reconstructing more details, particularly in distant regions. Regarding geometric representation, our Aquatic-GS generates more reasonable depth maps that are smooth with clear edges and minimal influence from floaters.

various distances, maintaining stable hues and outperforming competitors in color fidelity.

For overall performance in underwater image restoration,

Table II shows that our approach outperforms others on the simulated dataset, achieving superior PSNR, SSIM, and LPIPS values—8.406 dB, 0.046, and 0.033 better than the next best

TABLE II

QUANTITATIVE EVALUATIONS OF THE UIR TASK WERE CONDUCTED ON THREE REAL-WORLD SCENES (D3, D5, AND CURACAO) WITH COLOR CHARTS AND OUR SIMULATED DATASET. METRICS ON REAL-WORLD SCENES ASSESS ALGORITHMS' COLOR CORRECTION CAPABILITIES, WHILE THOSE ON THE SIMULATED DATASET EVALUATE OVERALL UNDERWATER IMAGE RESTORATION PERFORMANCE. FOR EACH SCENE OR DATASET, THE BEST METRIC IS HIGHLIGHTED IN **BOLD** AND THE SECOND-BEST IN UNDERLINE.

Method		D3 Scene		D5 Scene		Curacao Scene		Simulated Dataset		
		$\Delta E_{00} \downarrow$	$\bar{\psi} \downarrow$	$\Delta E_{00} \downarrow$	$\bar{\psi} \downarrow$	$\Delta E_{00} \downarrow$	$\bar{\psi} \downarrow$	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
Visual	Fusion [39]	22.94	22.72	24.06	25.60	30.22	26.61	18.152	0.873	0.097
Prior-based	MLLE [40]	23.34	<u>21.50</u>	29.97	24.98	23.82	25.20	16.895	0.729	0.174
Data-driven	Waternet [43]	21.63	23.39	25.37	24.86	24.86	25.67	<u>20.795</u>	<u>0.924</u>	0.096
	TACL [46]	21.59	23.70	25.54	26.62	25.34	26.13	20.219	0.909	0.123
	TUDA [45]	24.71	24.52	22.11	25.30	29.00	27.18	19.351	0.904	0.089
	UVEB [44]	<u>19.98</u>	22.18	34.69	26.13	21.03	25.29	19.158	0.855	0.186
Physics-based	Seathru [49]	23.84	23.16	21.75	23.98	28.43	25.78	19.199	0.914	<u>0.075</u>
	Hazeline [50]	22.49	20.01	25.11	24.04	30.90	24.63	14.259	0.715	0.236
	USUIR [51]	25.08	23.47	23.32	26.29	31.89	27.09	18.902	0.883	0.124
	SUCRe [5]	24.99	23.24	16.50	<u>21.90</u>	30.58	25.90	20.140	0.918	0.087
NeRF-based	NeuralSea [8]	-	-	-	-	28.17	29.28	11.236	0.422	0.486
	SeaThru-NeRF [9]	20.24	23.57	34.71	26.98	21.36	<u>23.50</u>	15.597	0.654	0.399
3DGS-based	Aquatic-GS (ours)	19.95	21.95	<u>20.16</u>	18.95	17.73	21.55	29.201	0.970	0.042

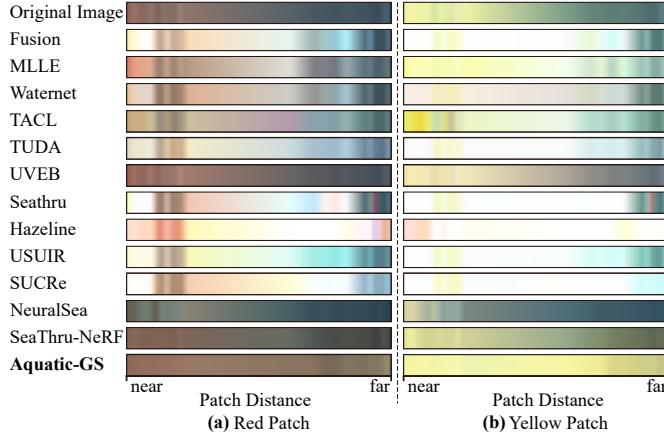


Fig. 5. Hue values vs. distance. We tracked the hue values of the restored (a) red and (b) yellow color patches at different observation distances within the Curacao scene. The first row illustrates the situation in the original underwater images. The results demonstrate that our Aquatic-GS outperforms other approaches in color correction and maintaining stable hue values.

methods, respectively—highlighting its effectiveness. Fig. 6 provides a visual comparison with state-of-the-art methods. In real-world scenes, MLLE struggles with enhancing distant regions due to a lack of underwater physics consideration. UVEB addresses color deviations partially but leaves a reddish tint and severe backscatter in distant areas, likely due to a domain gap between training and testing images. SUCRe corrects colors and recovers details in distant regions but fails in areas with poorly reconstructed 3D models. SeaThru-NeRF recovers distant appearances more accurately, yet its performance is unstable, with the D5 scene unrecovered, distant regions in the Curacao scene dark, and D3 scene textures blurry. In contrast, our method stably restores scenes

TABLE III
ABLATION STUDY ON MODELING NON-UNIFORM DISTRIBUTIONS OF WATER PARAMETERS.

Distribution Setting	$\Delta E_{00} \downarrow$	ΔE_{00} std \downarrow	$\bar{\psi} \downarrow$	$\bar{\psi}$ std \downarrow
Uniform Distribution	19.92	4.03	21.55	2.25
Non-uniform Distribution	17.73	0.64	21.55	0.97

with realistic colors and textures, especially in distant regions. In the simulated S3 scene, MLEE corrects color but over-enhances contrast, UVEB and SeaThru-NeRF show limited restoration, and SUCRe removes backscatter but introduces color artifacts. Unlike these, our method effectively mitigates severe water effects, producing results closer to the ground truth. Overall, our method outperforms others in terms of visual fidelity, color correction, detail recovery, and stability.

D. Ablation Studies

To demonstrate the effectiveness of our Aquatic-GS, we conducted a series of ablation studies. These studies systematically evaluated the effectiveness of modeling water parameters with non-uniform distributions, as well as the contribution of each loss function component in the DGO mechanism. All ablation experiments were performed on the SeathruNeRF dataset, with experimental settings kept consistent across all tests.

1) *Ablation Study on Modeling Non-uniform Distributions of Water Parameters:* To validate the need for modeling water parameters as non-uniform distributions, we create a configuration where water parameters are uniformly distributed within each channel. Table III quantitatively evaluates color

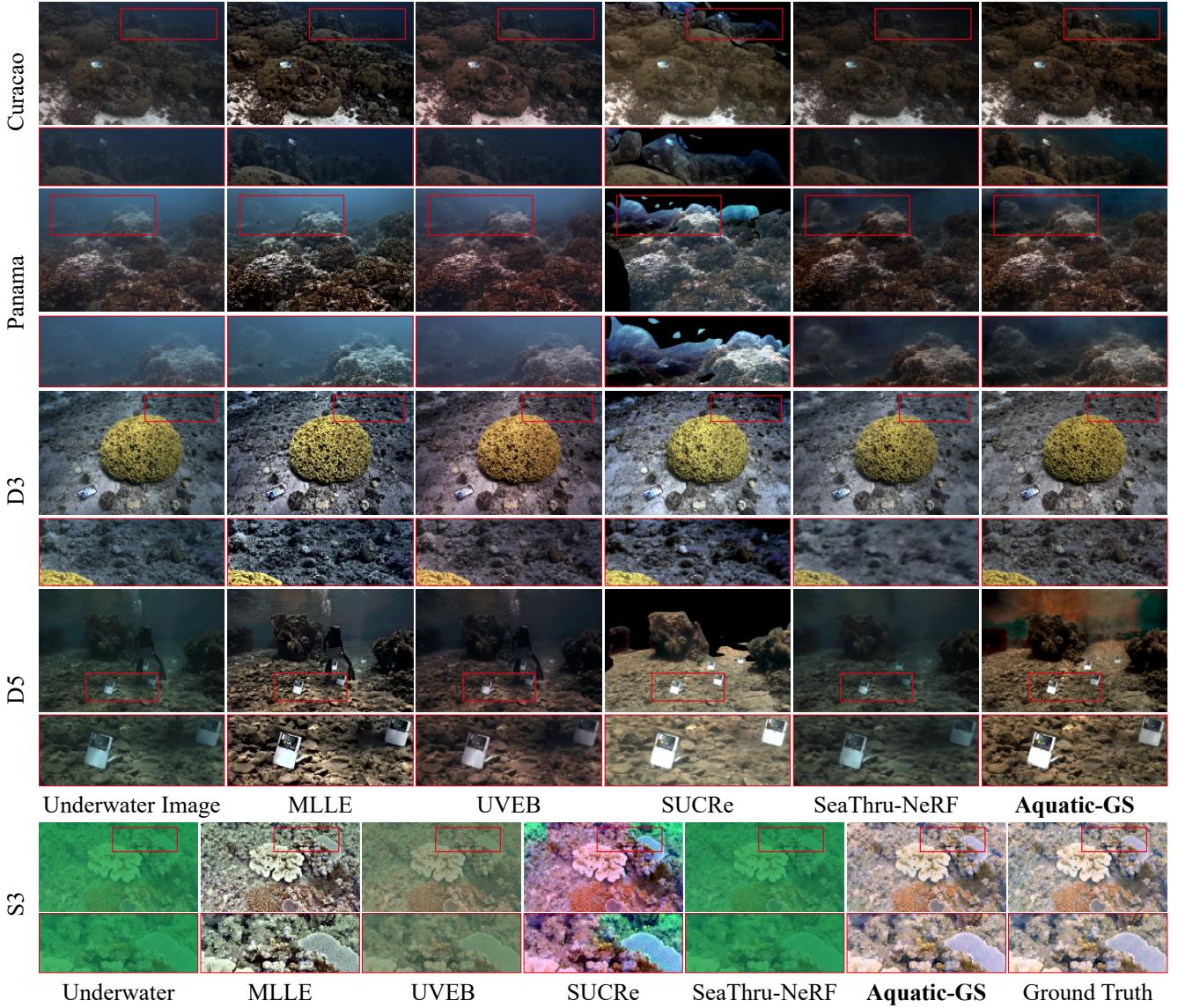


Fig. 6. Visual comparison in the UIR task. We present visual comparison results among our method and the latest methods from visual prior-based, data-driven, physics-based, and NeRF-based approaches. For the simulated dataset, the corresponding water-free ground truth is displayed. Challenging regions are highlighted with red boxes. Our method effectively restores the true appearance of distant areas, where water effects are pronounced and underwater image contrast is low, achieving more realistic colors and finer texture details. Even in the S3 scene, where the color cast is particularly pronounced, our method yields results that closely match the ground truth.

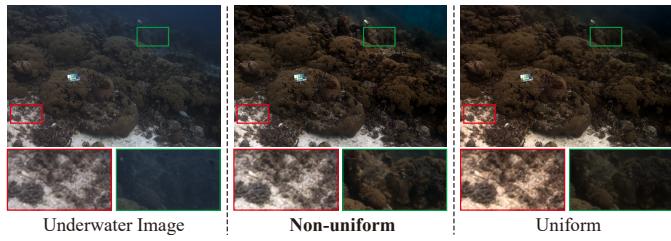


Fig. 7. Visual comparison of the learned water-free images with different settings of water parameter distribution. The red box highlights a region in the lower left of the view, while the green box emphasizes a region in the upper right. Our non-uniform distribution setting faithfully restores the scene colors, avoiding both over-enhancement and under-restoration.

correction performance in the Curacao scene under different

water parameter distribution settings. We report the standard deviations of ΔE_{00} and $\bar{\psi}$ across all restored color charts to assess restoration stability. With uniform water parameter distributions, ΔE_{00} increased by 2.19, indicating reduced color correction accuracy, while the standard deviations of ΔE_{00} and $\bar{\psi}$ rose by 3.39 and 1.28, reflecting greater instability.

Fig. 7 shows the learned water-free images under different settings. With the uniform distribution setting, the restored underwater image shows a reddish tint in the lower left view due to over-restoration, while the upper right view appears too dark, indicating insufficient recovery of attenuated light. This suggests that uniformly distributed water parameters are excessive for the lower left and insufficient for the upper right, failing to capture complex spatial variations of water parameters. In contrast, our non-uniform distribution setting

TABLE IV
ABLATION STUDY ON THE INDIVIDUAL COMPONENTS OF THE DGO MECHANISM.

Comparative Model	L_{dgt}	L_{dvm}	L_{idc}	L_{dpf}	PSNR ↑	SSIM ↑	LPIPS ↓
Base					26.695	0.879	0.205
-	✓				27.092	0.880	0.205
-	✓	✓			27.823	0.890	0.191
-	✓	✓	✓		27.906	0.893	0.190
Aquatic-GS	✓	✓	✓	✓	28.125	0.894	0.176

provides more reliable restoration, avoiding both over- and under-restoration, demonstrating its effectiveness in complex real-world scenarios.

2) *Ablation Study on The Individual Components of The DGO Mechanism:* To validate the effectiveness of each loss function component in the proposed depth-guided optimization mechanism, we designate the Aquatic-GS without the DGO mechanism as the Base model. We then incrementally incorporate L_{dgt} , L_{dvm} , L_{idc} , and L_{dpf} to establish various comparison models, as shown in Table IV. This table presents quantitative metrics for different combinations of the DGO mechanism's components in the underwater NVS task, evaluating their performance in representing underwater scenes. The results indicate that each loss function component contributes to performance improvements to varying degrees. Notably, L_{dgt} and L_{idc} significantly enhance the PSNR, while L_{dvm} effectively improves PSNR, SSIM, and LPIPS. The inclusion of L_{dpf} substantially boosts PSNR and reduces LPIPS, resulting in the best performance.

Compared with the Base model, our Aquatic-GS with the DGO mechanism achieves a PSNR improvement of 1.43 dB, an SSIM increase of 0.015, and a further LPIPS reduction of 0.029. As illustrated in Fig. 3, the DGO mechanism effectively facilitates the decoupling of water effects from the water-free scene, leading to a reduction in haze and blur while restoring more geometry details, particularly in challenging distant regions.

V. CONCLUSION

In this paper, we present Aquatic-GS, a novel 3D representation method for underwater scenes. This method employs a hybrid representation strategy that implicitly models the water medium while explicitly representing the objects. Specifically, the neural water field learns the non-uniform distributions of water parameters, while the extended 3DGS accurately captures the true appearance and geometry of underwater scenes. Both components are unified using a physics-based underwater imaging model to generate high-quality underwater images. Our depth-guided optimization mechanism overcomes the negative effects of water, ensuring accurate scene geometry, especially for distant details. Our Aquatic-GS effectively renders novel underwater viewpoints and faithfully restores underwater scenes. Extensive experiments on real-world and simulated datasets demonstrate that Aquatic-GS surpasses

SOTA NeRF-based underwater representation methods with higher rendering quality and 410× faster real-time rendering performance. Furthermore, it outperforms representative de-watering approaches in color correction, detail recovery, and stability.

REFERENCES

- [1] W. Wang, B. Joshi, N. Burgdorfer, K. Batsosc, A. Q. Lid, P. Mordohaia, and I. Rekleitis, "Real-time dense 3d mapping of underwater environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5184–5191.
- [2] M. Yuval and T. Treibitz, "Releasing a dataset of 3d models of artificial reefs from the northern red-sea for 3d printing and virtual reality applications," *Remote Sensing Applications: Society and Environment*, vol. 36, p. 101305, 2024.
- [3] M. Johnson-Roberson, M. Bryson, A. Friedman, O. Pizarro, G. Troni, P. Ozog, and J. C. Henderson, "High-resolution underwater robotic vision-based mapping and three-dimensional reconstruction for archaeology," *Journal of Field Robotics*, vol. 34, no. 4, pp. 625–643, 2017.
- [4] G. Llorach-Tó, E. Martínez, J. D. R. Fernández, and E. García-Ladona, "Experience obsea: a web-based 3d virtual environment of a seafloor observatory," in *OCEANS 2023-Limerick*. IEEE, 2023, pp. 1–6.
- [5] C. Boittiaux, R. Marquer, C. Dune, A. Arnaubec, M. Ferrera, and V. Hugel, "Sucré: Leveraging scene structure for underwater color restoration," in *2024 International Conference on 3D Vision (3DV)*. IEEE, 2024, pp. 1488–1497.
- [6] 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," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [7] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.
- [8] T. Zhang and M. Johnson-Roberson, "Beyond nerf underwater: Learning neural reflectance fields for true color correction of marine imagery," *IEEE Robotics and Automation Letters*, 2023.
- [9] D. Levy, A. Peleg, N. Pearl, D. Rosenbaum, D. Akkaynak, S. Korman, and T. Treibitz, "Seathru-nerf: Neural radiance fields in scattering media," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 56–65.
- [10] Y. Tang, C. Zhu, R. Wan, C. Xu, and B. Shi, "Neural underwater scene representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 11780–11789.
- [11] B. Fei, J. Xu, R. Zhang, Q. Zhou, W. Yang, and Y. He, "3d gaussian splatting as new era: A survey," *IEEE Transactions on Visualization and Computer Graphics*, 2024.
- [12] Y. Bekerman, S. Avidan, and T. Treibitz, "Unveiling optical properties in underwater images," in *2020 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2020, pp. 1–12.
- [13] D. Akkaynak and T. Treibitz, "A revised underwater image formation model," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6723–6732.
- [14] D. Nakath, M. She, Y. Song, and K. Köser, "In-situ joint light and medium estimation for underwater color restoration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3731–3740.
- [15] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10371–10381.
- [16] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5470–5479.
- [17] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.
- [18] Y. Yang, S. Zhang, Z. Huang, Y. Zhang, and M. Tan, "Cross-ray neural radiance fields for novel-view synthesis from unconstrained image collections," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 15901–15911.
- [19] L. Xu, Y. Xiangli, S. Peng, X. Pan, N. Zhao, C. Theobalt, B. Dai, and D. Lin, "Grid-guided neural radiance fields for large urban scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8296–8306.

- [20] A. Ramazzina, M. Bijelic, S. Walz, A. Sanvito, D. Scheuble, and F. Heide, "Scatternerf: Seeing through fog with physically-based inverse neural rendering," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 957–17 968.
- [21] H. Dahmani, M. Bennehar, N. Piasco, L. Roldao, and D. Tsishkou, "Swag: Splatting in the wild images with appearance-conditioned gaussians," *arXiv preprint arXiv:2403.10427*, 2024.
- [22] Z. Zhu, Z. Fan, Y. Jiang, and Z. Wang, "Fsgs: Real-time few-shot view synthesis using gaussian splatting," in *European Conference on Computer Vision*. Springer, 2025, pp. 145–163.
- [23] H. Xiong, S. Muttukuru, R. Upadhyay, P. Chari, and A. Kadambi, "Sparsesgs: Real-time 360 $\{\deg\}$ sparse view synthesis using gaussian splatting," *arXiv preprint arXiv:2312.00206*, 2023.
- [24] J. Li, J. Zhang, X. Bai, J. Zheng, X. Ning, J. Zhou, and L. Gu, "Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 775–20 785.
- [25] A. Palwal, W. Ye, J. Xiong, D. Kotovenko, R. Ranjan, V. Chandra, and N. K. Kalantari, "Coherentgs: Sparse novel view synthesis with coherent 3d gaussians," *arXiv preprint arXiv:2403.19495*, 2024.
- [26] W. Chen and L. Liu, "Deblurgs: 3d gaussian splatting from camera motion blurred images," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 7, no. 1, pp. 1–15, 2024.
- [27] B. Lee, H. Lee, X. Sun, U. Ali, and E. Park, "Deblurring 3d gaussian splatting," *arXiv preprint arXiv:2401.00834*, 2024.
- [28] J. Lin, Z. Li, X. Tang, J. Liu, S. Liu, J. Liu, Y. Lu, X. Wu, S. Xu, Y. Yan *et al.*, "Vastgaussian: Vast 3d gaussians for large scene reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5166–5175.
- [29] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, "4d gaussian splatting for real-time dynamic scene rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 310–20 320.
- [30] Y. Gong, "Eggs: Edge guided gaussian splatting for radiance fields," in *Proceedings of the 29th International ACM Conference on 3D Web Technology*, 2024, pp. 1–5.
- [31] J. Zhang, F. Zhan, M. Xu, S. Lu, and E. Xing, "Fregs: 3d gaussian splatting with progressive frequency regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 424–21 433.
- [32] J. Hyung, S. Hong, S. Hwang, J. Lee, J. Choo, and J.-H. Kim, "Effective rank analysis and regularization for enhanced 3d gaussian splatting," *arXiv preprint arXiv:2406.11672*, 2024.
- [33] Z. Zhang, W. Hu, Y. Lao, T. He, and H. Zhao, "Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting," *arXiv preprint arXiv:2403.15530*, 2024.
- [34] Z. Ye, W. Li, S. Liu, P. Qiao, and Y. Dou, "Absgs: Recovering fine details in 3d gaussian splatting," in *ACM Multimedia 2024*, 2024.
- [35] G. Fang and B. Wang, "Mini-splatting: Representing scenes with a constrained number of gaussians," *arXiv preprint arXiv:2403.14166*, 2024.
- [36] M. Turkulainen, X. Ren, I. Melekhov, O. Seiskari, E. Rahtu, and J. Kannala, "Dn-splatter: Depth and normal priors for gaussian splatting and meshing," *arXiv preprint arXiv:2403.17822*, 2024.
- [37] H. Li, W. Song, T. Xu, A. Elsig, and J. Kulhanek, "Watersplatting: Fast underwater 3d scene reconstruction using gaussian splatting," *arXiv preprint arXiv:2408.08206*, 2024.
- [38] T. Zhang, W. Zhi, K. Huang, J. Mangelson, C. Barbalata, and M. Johnson-Roberson, "Recgs: Removing water caustic with recurrent gaussian splatting," *arXiv preprint arXiv:2407.10318*, 2024.
- [39] C. Ancuti, C. O. Ancuti, T. Haber, and P. Bekaert, "Enhancing underwater images and videos by fusion," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 81–88.
- [40] W. Zhang, P. Zhuang, H.-H. Sun, G. Li, S. Kwong, and C. Li, "Underwater image enhancement via minimal color loss and locally adaptive contrast enhancement," *IEEE Transactions on Image Processing*, vol. 31, pp. 3997–4010, 2022.
- [41] J. Zhou, S. Wang, Z. Lin, Q. Jiang, and F. Sohel, "A pixel distribution remapping and multi-prior retinex variational model for underwater image enhancement," *IEEE Transactions on Multimedia*, 2024.
- [42] M. Badran and M. Torki, "Daut: Underwater image enhancement using depth aware u-shape transformer," in *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2023, pp. 1830–1834.
- [43] C. Li, C. Guo, W. Ren, R. Cong, J. Hou, S. Kwong, and D. Tao, "An underwater image enhancement benchmark dataset and beyond," *IEEE transactions on image processing*, vol. 29, pp. 4376–4389, 2019.
- [44] Y. Xie, L. Kong, K. Chen, Z. Zheng, X. Yu, Z. Yu, and B. Zheng, "Uverb: A large-scale benchmark and baseline towards real-world underwater video enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 22 358–22 367.
- [45] Z. Wang, L. Shen, M. Xu, M. Yu, K. Wang, and Y. Lin, "Domain adaptation for underwater image enhancement," *IEEE Transactions on Image Processing*, vol. 32, pp. 1442–1457, 2023.
- [46] R. Liu, Z. Jiang, S. Yang, and X. Fan, "Twin adversarial contrastive learning for underwater image enhancement and beyond," *IEEE Transactions on Image Processing*, vol. 31, pp. 4922–4936, 2022.
- [47] P. L. Drews, E. R. Nascimento, S. S. Botelho, and M. F. M. Campos, "Underwater depth estimation and image restoration based on single images," *IEEE computer graphics and applications*, vol. 36, no. 2, pp. 24–35, 2016.
- [48] Y.-T. Peng and P. C. Cosman, "Underwater image restoration based on image blurriness and light absorption," *IEEE transactions on image processing*, vol. 26, no. 4, pp. 1579–1594, 2017.
- [49] D. Akkaynak and T. Treibitz, "Sea-thru: A method for removing water from underwater images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1682–1691.
- [50] D. Berman, D. Levy, S. Avidan, and T. Treibitz, "Underwater single image color restoration using haze-lines and a new quantitative dataset," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 8, pp. 2822–2837, 2020.
- [51] Z. Fu, H. Lin, Y. Yang, S. Chai, L. Sun, Y. Huang, and X. Ding, "Unsupervised underwater image restoration: From a homology perspective," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, 2022, pp. 643–651.
- [52] A. V. Sethuraman, M. S. Ramanagopal, and K. A. Skinner, "Waternerf: Neural radiance fields for underwater scenes," in *OCEANS 2023-MTS/IEEE US Gulf Coast*. IEEE, 2023, pp. 1–7.
- [53] R. Habel, B. Mustata, and M. Wimmer, "Efficient spherical harmonics lighting with the preetham skylight model," in *Eurographics (short papers)*, 2008, pp. 119–122.
- [54] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5501–5510.
- [55] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross, "Ewa volume splatting," in *Proceedings Visualization, 2001. VIS'01*. IEEE, 2001, pp. 29–53.
- [56] C. Reiser, S. Garbin, P. Srinivasan, D. Verbin, R. Szeliski, B. Mildenhall, J. Barron, P. Hedman, and A. Geiger, "Binary opacity grids: Capturing fine geometric detail for mesh-based view synthesis," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 4, pp. 1–14, 2024.
- [57] D. Rebain, M. Matthews, K. M. Yi, D. Lagun, and A. Tagliasacchi, "Lolnerf: Learn from one look," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1558–1567.
- [58] L. Jiang, B. Dai, W. Wu, and C. C. Loy, "Focal frequency loss for image reconstruction and synthesis," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 13 919–13 929.
- [59] C. Korkmaz, A. M. Tekalp, and Z. Dogan, "Training generative image super-resolution models by wavelet-domain losses enables better control of artifacts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5926–5936.
- [60] Y. Liu, C. Yu, J. Cheng, Z. J. Wang, and X. Chen, "Mm-net: A mixformer-based multi-scale network for anatomical and functional image fusion," *IEEE Transactions on Image Processing*, vol. 33, pp. 2197–2212, 2024.
- [61] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.
- [62] Z. Murez, T. Treibitz, R. Ramamoorthi, and D. Kriegman, "Photometric stereo in a scattering medium," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3415–3423.
- [63] L. Peng, C. Zhu, and L. Bian, "U-shape transformer for underwater image enhancement," *IEEE Transactions on Image Processing*, vol. 32, pp. 3066–3079, 2023.
- [64] G. Sharma, W. Wu, and E. N. Dalal, "The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations," *COLOR research & application*, vol. 30, no. 1, pp. 21–30, 2005.