

Osmosis: RGBD Diffusion Prior for Underwater Image Restoration

Opher Bar Nathan, Deborah Levy, Tali Treibitz, and Dan Rosenbaum

University of Haifa

Abstract. Underwater image restoration is a challenging task because of strong water effects that increase dramatically with distance. This is worsened by lack of ground truth data of clean scenes without water. Diffusion priors have emerged as strong image restoration priors. However, they are often trained with a dataset of the desired restored output, which is not available in our case. To overcome this critical issue, we show how to leverage in-air images to train diffusion priors for underwater restoration. We also observe that only color data is insufficient, and augment the prior with a depth channel. We train an unconditional diffusion model prior on the joint space of color and depth, using standard RGBD datasets of natural outdoor scenes in air. Using this prior together with a novel guidance method based on the underwater image formation model, we generate posterior samples of clean images, removing the water effects. Even though our prior did not see any underwater images during training, our method outperforms state-of-the-art baselines for image restoration on very challenging scenes. Data, models and code are published in the project page osmosis-diffusion.github.io.

Keywords: Diffusion Models · Underwater Image Restoration

1 Introduction

Underwater images are used in many applications, such as, underwater construction and maintenance, marine sciences, and fisheries. However, their automatic analysis is hindered because of the optical effects of the water that strongly attenuates and scatters light in a wavelength dependent manner. This causes color distortion and loss of contrast that exponentially increase with depth¹. With growing human activity in the oceans, clear underwater vision becomes increasingly important.

Restoring underwater scenes is still a very challenging ill-posed problem. Classic approaches are based on designing priors for clean images or inverting the water formation model, and are limited by the ability to form strong priors. On the other hand, learning based approaches are limited by the lack of supervised training data of clean underwater images. This is a critical issue, as the ocean cannot be emptied for the sake of data collection.

¹ For consistency with computer vision literature the term *depth* is used throughout to refer to the distance from the camera rather than water depth.

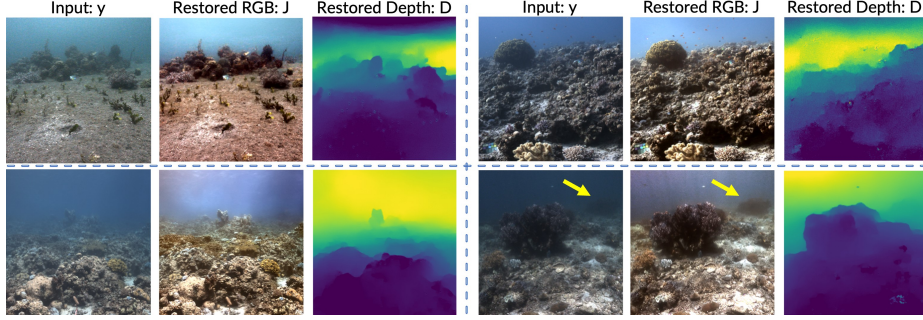


Fig. 1: Our method receives as an input a single underwater image and outputs the restored clean image and an estimated depth map. The output is estimated using a diffusion prior trained on RGBD images and the physical image formation model.

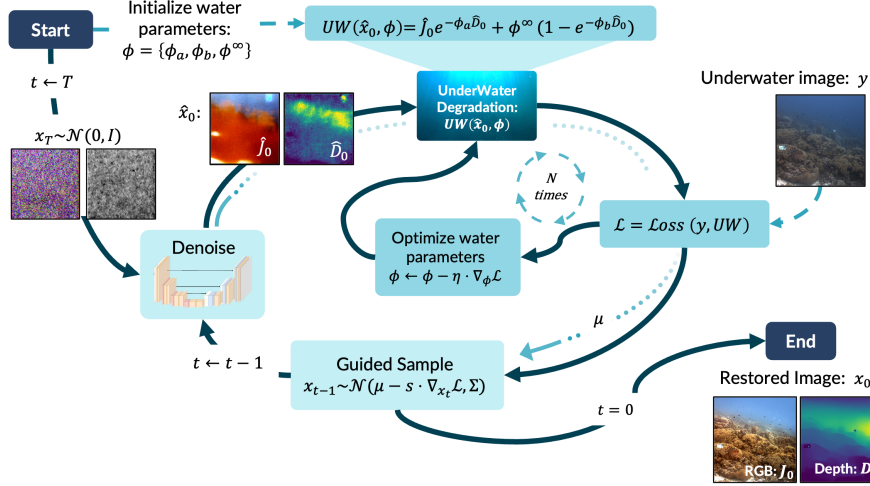


Fig. 2: The iterative sampling process starts in $t = T$ with random noise in 4 channels. The denoising step outputs μ and denoised samples $\hat{x}_0 = (\hat{J}_0, \hat{D}_0)$. We use the physical image formation model together with \hat{x}_0 to optimize the water parameters $\hat{\phi}$. These are then input to the guided sampling. This process repeats itself, gradually updating both the estimated image and depth, until $t = 0$, in which x_0 holds the method’s estimate for both the reconstructed scene J_0 and its depth D_0 .

We suggest an *unsupervised* restoration method (Figs. 1,2) based on an *inverse problem* approach using a diffusion prior for both color and depth, coupled with the underwater image formation model. Restoring an image is formulated as posterior inference, computed using a natural image prior, and a likelihood term that is based on the underwater image formation model. The challenge with applying this approach for underwater image restoration directly is that (i) the degradation for each pixel depends on its depth, and other unknown parameters; and (ii) there is no ground truth clean underwater data to train the prior.

To solve this, we replace the image prior with a prior on the joint space of color and depth of natural images. Adding depth to the prior allows us to

formulate the forward model that forms the likelihood term over the observed corrupted underwater image, and apply posterior sampling. Moreover, this leverages the high capacity of diffusion models, in capturing the strong correlations between color and depth in natural scenes.

We propose to train a prior model of RGBD images using available datasets of natural outdoor scenes that were collected in air. Using in-air scenes for underwater images may be counter-intuitive, but actually has strong benefits: 1) overcomes the lack of clean underwater image data; 2) leads to a strong prior that captures the joint statistics governing color and depth in natural scenes, where, as opposed to underwater images, color does not fade with distance; 3) prevents overfitting to specific types of underwater images.

We then use this prior together with the underwater physical image formation model to simultaneously estimate the clean image, its depth, and the model parameters, all from a single underwater image (summarized in Fig. 2). We show that our method outperforms models that were trained on underwater data.

The main contributions of this paper are:

1. We train an RGBD prior, and demonstrate that modeling color and depth and jointly sampling them, provides a stronger diffusion prior for underwater image restoration.
2. We propose a new method that combines the RGBD prior of in-air data with the underwater image formation model, leading to a diffusion guidance method that generates the restored underwater images as posterior samples.
3. We demonstrate that our method outperforms state-of-the-art underwater image restoration methods both qualitatively and quantitatively on real and simulated data.

We publish all our code, the new trained RGBD prior, images and results.

2 Related Works

2.1 Underwater Image Restoration and Depth Estimation

Here we review recent works that are most relevant to our method. See [62] for a recent more comprehensive review.

Classic image restoration. To cope with the ill-posed nature of the problem, earlier works on underwater image restoration introduced tailored image priors. Some priors aim to estimate a depth or transmission map of the scene to reduce the number of the unknowns and then use the estimated depth to restore the scenes [8, 41, 42].

Supervised learning. Clean underwater image data is very scarce. Several datasets aim to mitigate this issue. In UIEB [35], the images denoted as ground-truth are generated from enhancement results of 9 different baseline methods, and having human observers vote for the best one. The LSUI dataset [40] is larger, produced using the same methodology, based on choosing results from 18 different methods. These datasets enable supervised learning methods, but are still limited by their scale, the quality of the baseline methods, and the human bias to choose visually pleasing rather than physically consistent images.

CWR [25] introduced the HICRD dataset, where the images are restored using optical parameters measured using ocean optics instruments. This is an impressive effort. Unfortunately, the images are acquired in a downward-looking position, and thus their depth range is very limited. In FUnIE-GAN [28] a training dataset is generated by having humans select *good* images from a large set of unprocessed underwater images. The humans are instructed to choose images where the *foreground* objects are identifiable. These are then distorted by a GAN to produce the paired *poor* images. A synthetic dataset was generated in [34]. The dataset is synthesized from the NYU-v2 RGBD dataset [46], using the image formation model equation and several sets of values for the water parameters.

Unsupervised learning. USUIR [20] aims to restore images without supervision, by separately estimating the image components (clean image, transmission, backscatter) and using them to construct an underwater image that is used for supervision against the original one. Subsequent frames were used in [4] for self-supervising monocular depth estimation, and in [54] for self-supervising both depth and restoration. In UW-NET [24] a cycle-GAN is used for learning mapping from RGBD in-air datasets to underwater images. **As opposed to all these methods, we present the first prior based on diffusion models that does not rely on ground truth underwater supervision, and uses the physical model for inference.**

2.2 Diffusion Model Prior

Diffusion models. Diffusion models have emerged as a powerful type of a generative model, especially for images. In the last few years several formulations and variations have been developed [26, 47, 50, 52], most of which use a U-Net architecture [43] as a noise predictor. Because the training relies on very large datasets and is extremely time-consuming, significant work has been devoted to the setup where models are first pre-trained on large datasets and only later fine-tuned to more specific data, closer to the tasks at hand [44, 48]. In DepthGen [45] this approach is taken forward, by using a pre-trained model to kickstart a new model trained on different modalities using a different architecture. This is done by replacing the input and output layers of the pre-trained U-Net. We take a similar approach for training our RGBD prior, by using a pretrained diffusion model that was trained on RGB only.

Conditional diffusion models have been used for various image restoration tasks, by training models using different levels of supervision. Examples of similar tasks as ours include dehazing and deraining [9, 39, 57], and shadow removal [23]. Conditional diffusion models for underwater image enhancement have also been proposed [37, 53]. Our approach differs from these by focusing on image restoration that inverts the physical model, rather than relying on supervised data that is optimized for image appearance.

Diffusion model as a prior for clean images. In addition to conditional generation of images, there is a growing body of research where diffusion models are being used as clean image priors, and image restoration is formulated as

posterior sampling [6, 10, 12–15, 19, 22, 29, 30, 49, 51]. These include tasks like denoising, inpainting, deblurring, and more general tasks. The image degradation model is used to form the likelihood term and sampling from the posterior is implemented by adding the gradient of the degradation model in each step of the inverse sampling iterations, a method also known as *guidance*. This can be implemented in different ways, differing by the way the likelihood term is approximated, and whether its computation and gradient are based on an intermediary noisy image or on the estimated clean image in each sampling step.

The limited access to clean data in many cases, has led to research on training a prior of clean images, using noisy training data only. In [1, 16, 31] a diffusion model prior is trained using noisy data, by assuming a known degradation model. Since this setup is not directly applicable in our case, both because of the model structure and the unknown parameters, we chose instead to train our prior on clean data that was not taken underwater.

Diffusion model prior for blind image restoration. A more challenging task, is to use diffusion models as a clean image prior when the degradation model is unknown or depends on unknown parameters. Several papers have proposed to tackle this problem in different setups. In [18, 38, 60] the unknown parameters are optimized during the sampling process with a reconstruction objective function. In [11] a separate prior is trained for the unknown parameters, and sampled in parallel with image sampling. **Our method differs from the above in that we learn a prior of the main unknown aspect of the degradation model, namely the depth, together with the prior of the variables we want to infer - the image color.** This is done by training a single diffusion model on the joint space of color and depth.

3 Preliminaries

3.1 Underwater Image Formation

In water, we observe two wavelength- and distance-dependent effects. First, the *direct* signal reflected from the object is attenuated. Second, light is scattered onto the object’s line-of-sight (LOS), creating an additive signal termed *backscatter* that increases with distance. The occluding backscatter layer is independent of the scene content. Thus, the visibility and contrast of further objects is significantly reduced and their colors are distorted.

Following the revised underwater image formation model [2, 32], under ambient illumination image intensity (per pixel, per color channel) is given as:

$$I = J \cdot e^{-\phi_a \cdot D} + \phi^\infty \cdot (1 - e^{-\phi_b \cdot D}) \quad , \quad (1)$$

where I is the linear image captured by the camera of a scene with range D , J is the clear scene that would have been captured had there been no water along the LOS, and ϕ^∞ is the water color at infinity, i.e., the backscatter at areas that contain no objects. The two parameters ϕ_a and ϕ_b are the attenuation and backscatter coefficients, respectively.

3.2 Diffusion Models

Diffusion models have proven to be very effective in capturing the distribution of natural images, given in training data. We describe here the formulation that we use in our work, adopted from [17].

A diffusion model is defined by a Markov chain designed to transform the distribution of real images x_0 to a Gaussian distribution x_T , by gradually scaling down the image values and adding Gaussian noise, using a schedule determined by the scalar parameters α_t ,

$$x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1 - \alpha_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I) . \quad (2)$$

Based on this *forward* process, an *inverse* process is trained to gradually denoise images starting from a Gaussian distribution, back into the distribution of the original image dataset. This can be formulated as a factorization of the joint distribution over the images in reverse order, $p(x_{0:T}) = \prod_{t=1}^T p_t(x_{t-1} | x_t)$, and $p_t(x_{t-1} | x_t)$ is approximated by a Gaussian

$$p_t(x_{t-1} | x_t) \sim \mathcal{N}(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) , \quad (3)$$

with parameters μ_θ and Σ_θ predicted by a trained neural network conditioned on x_t and t . Simulating this process results in samples from the approximated data distribution $p(x_0)$. A popular implementation first predicts the noise at each time step, and then computes the mean by

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) , \quad (4)$$

where $\epsilon_\theta(x_t, t)$ is the neural network trained to approximate ϵ , and $\bar{\alpha}_t = \prod_{s=0}^t \alpha_s$. The same neural network can also be used to predict a diagonal covariance $\Sigma_\theta(x_t, t)$. At any iteration, an estimate of the clean image x_0 can be derived by computing the mean of $p(x_0 | x_t)$:

$$\hat{x}_0(x_t, t) = \frac{1}{\sqrt{\bar{\alpha}_t}} (x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(x_t, t)) . \quad (5)$$

Score-based diffusion models represent another view of the above formulation, where the inverse sampling process is shown to be equivalent to an iterative stochastic differential equation (SDE) solver, and the forward process is assumed to be a continuous time differential equation. In that case it can be shown that $\epsilon_\theta(x_t, t)$ is an approximation to the score function $\nabla_{x_t} \log p_t(x_t)$.

3.3 Posterior Sampling

Under the score-based view, given an observation y and a likelihood function $p(y|x)$, we can use the same mechanism to sample from the posterior $p(x|y)$, using the posterior score function.

$$\nabla_{x_t} \log p_t(x_t|y) = \nabla_{x_t} \log p_t(x_t) + \nabla_{x_t} \log p_t(y|x_t) \quad (6)$$

This idea of adding a conditional signal to the score is also called *guidance*. The challenge with the second term in Eq. 6, is that usually we are given a likelihood model based on the clean image x_0 , and not an intermediary noisy image x_t . In our case the underwater image formation model transforms a clean image to an underwater observation. Connecting the model to the noisy sample x_t leads to the intractable integral $p_t(y|x_t) = \int p(y|x_0)p(x_0|x_t)dx_0$.

The various works on posterior sampling propose different approximations of this integral. Some propose to collapse the uncertainty in y [10, 13–15, 22, 51]. In [19] a variational inference approximation is proposed, and in [6, 12, 29, 30, 49] the likelihood model is computed on either x_t directly, or on the mean $\hat{x}_0 = \mathbb{E}[x_0|x_t]$, as given in Eq. 5. After some experimentation, we decide to use the latter approximation as formulated in DPS [12]:

$$\nabla_{x_t} \log p_t(x_t | y) \approx \nabla_{x_t} \log p_t(x_t) + s \nabla_{x_t} \log p(y | \hat{x}_0(x_t, t)) \quad , \quad (7)$$

where s is the *guidance scale* used to control the weight of the approximated likelihood term.

4 Method

4.1 Our Approach

The approach in this work is to use the underwater image formation model (Eq. 1) as a forward model that maps the space of natural images x to the space of underwater observed images y . We aim to use this forward model to construct a likelihood term $p(y | x)$, and use it to sample from the posterior distribution of images (Eq. 7).

In our case this cannot be implemented directly since the image formation model contains unknown parameters, namely the depth D at each pixel, and the water parameters, $\phi_a, \phi_b, \phi^\infty$, which can differ between scenes. One way to deal with the unknown depth, is to use a monocular depth estimator in a separate first stage, and then use it as part of the forward model. In the results section we discuss this approach and show that it is suboptimal (method variant termed *DA-Osmosis*).

Another approach is to optimize the unknown parameters, including the depth, during sampling. In some recent work, different methods to do this were proposed [18, 38, 60]. However, these do not work properly when the unknown parameter is high dimensional, highly complex, and strongly correlated with the target image. In our early experimentation we found that in order to separate the image into a clean image and the effects governed by the depth, we need more than a good image prior.

Therefore, one of our main contributions, is to train a joint prior on both the color and depth in clean images, and show how it can be used for underwater image restoration. Following this we define $x_0 = (J, D)$, where J represents a clear image and consists of the 3 color channels, and D is the depth image. This



Fig. 3: **[Left]** Example images from outdoor RGBD datasets used for training our prior. From left to right: DIODE [55], ReDWeb-S [36], HR-WSI [58], KITTI [21]. **[Right]** Samples from the trained RGBD prior. The samples demonstrate the inherent correlation between RGB image and depth in our trained RGBD prior.

allows us to use the following forward model (based on Eq. 1) conditioned on the estimated clean image and depth $\hat{x}_0 = (\hat{J}_0, \hat{D}_0)$:

$$f_{\phi}(\hat{x}_0) = \hat{J}_0 e^{-\phi_a \hat{D}_0} + \phi^{\infty} (1 - e^{-\phi_b \hat{D}_0}) , \quad (8)$$

where $\phi = (\phi_a, \phi_b, \phi^{\infty})$ are the remaining 9 unknown water parameters. We optimize ϕ during sampling, in a similar way to GDP [18], using gradient descent on the image reconstruction loss. The likelihood term used for guidance is defined as a Gaussian around $f_{\phi}(\hat{x}_0)$ with a fixed variance.

Training a joint prior on color and depth, not only allows us to use the above forward model, but also exploits the diffusion model capacity to capture the complexity of the depth image and its correlations with the clean color image.

4.2 Training the Prior

To learn a natural image prior, we train the joint prior model on both color and depth using public RGBD datasets of outdoor scenes that were taken in air. While differing from underwater images, the ability to use large amounts of high quality training data leads to a prior that captures the correlation between color and depth in natural scenes, which as we show, is an important aspect for underwater image restoration.

For the sake of data efficiency and training time, we start with a pretrained diffusion model trained on ImageNet (we take the unconditional model from [17]), and finetune it on RGBD data. We implement this, inspired by DepthGen [45], by replacing the first and last layers of the U-Net to 4 channels rather than 3, and initializing those layers randomly. The datasets we use for finetuning are (see Fig. 3[left]): DIODE [55], RedWeb-S [36], HR-WSI [58], KITTI [21], with 16884, 2179, 20378, 23946 pictures, respectively.

A major challenge in working with RGBD data is to turn the available depth information into a proper image. This includes filling holes and scaling the values to a standard range. Since each of the above datasets were collected in a different manner, we treat each of them differently. This is detailed in the appendix. Fig. 3[right] shows two samples from the prior that show corresponding color and depth images. While prior samples demonstrate an evident domain gap

between in-air and underwater data, our results show that posterior samples are not affected by the gap, and can leverage the strong correlation between depth and color in natural scenes.

4.3 Sampling from the Posterior

Given a trained prior, we perform underwater image restoration by sampling from the posterior with guidance of the image formation model. The sampling process is described in Figs. 2, 4. In each iteration we generate samples of both the image and the depth, while optimizing the water parameters. This results in a gradual update to the estimates of the clean image, and the depth, demonstrated in Fig. 4[right]. We adopt the sampling method in DPS [12], using the reconstruction loss, $\|y - f_\phi(\hat{x}_0)\|_2^2$, which is the negative log-likelihood formed from the model in Eq. 8. Similar to GDP [18], the remaining unknown parameters ϕ are optimized in parallel to the inverse sampling process, using gradient descent on the same loss used for guidance.

Note that one of our main novelties is that the RGBD prior is used as an inherent part of the iterative method and not as a stand-alone depth estimator. The depth is estimated together with the image in *every* iteration, guided by the underwater model, see, e.g., Fig. 4[right]. As demonstrated in the results, this improves the restoration quality over using a fixed estimated depth of the scene.

Running posterior sampling, guided by a reconstruction loss only, we observe two problems. First, for pixels with large depth values, the reconstructed image is dominated by the backscatter, making the estimation of the clean pixel color values unstable. Second, the color values can shift outside the valid range, causing color saturation.

To overcome the first issue we multiply the reconstruction loss in every pixel by the estimated depth value (without passing the gradients of this operation, denoted by sg):

$$\mathcal{L}_{\text{rec}} = \|sg(\hat{D}_0) \cdot (y - f_\phi(\hat{x}_0))\|_2^2. \quad (9)$$

In order to overcome the second issue, we introduce two auxiliary losses. The first penalizes RGB values outside the valid range $[-1, 1]$, and the second encourages the average values of each channel to approach the middle of the color range (in line with the *gray world assumption*). We implement the two auxiliary losses as:

$$\mathcal{L}_{\text{val}} = \lambda_v \sum_{i,c} \max\left(\left|\hat{J}_0(i, c)\right| - T_v, 0\right)^2, \quad \mathcal{L}_{\text{avg}} = \lambda_a \sum_c \left| \sum_i \hat{J}_0(i, c) - T_a \right|$$

where $\hat{J}_0(i, c)$ is the value of color channel c of pixel i , assuming the valid color range is $[-1, 1]$, T_v and T_a are thresholds set to values close to 1 and 0 respectively, and λ_v, λ_a are the scalar weights of both losses. The total loss is then given by:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{val}} + \mathcal{L}_{\text{avg}}. \quad (10)$$

At each iteration we compute the gradient of the loss with respect to x_t , which forms the log-likelihood gradient in the posterior score (Eq. 7), and the gradient

Algorithm 1: Osmosis - Sampling

Input:
 y - Underwater Degraded Image (RGB image)
 $f(\hat{x}_0)$ - Underwater Physical Degradation Model
Initialization of $\phi = \{\phi_a, \phi_b, \phi^\infty\}$ // size = [9]
Output:
 $\hat{x}_0 = [\hat{J}_0, \hat{D}_0]$: \hat{J}_0 - Restored RGB Image,
 \hat{D}_0 - Depth Estimation

```

1  $x_T \sim \mathcal{N}(0, I)$  //  $x_T$  size = [h,w,4]
2 for  $t = T : 1$  do
3    $\epsilon \leftarrow \epsilon_\theta(x_t, t)$ 
4    $\mu = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\alpha_t}} \cdot \epsilon)$ 
5    $\hat{x}_0 = \frac{1}{\sqrt{\alpha_t}}(x_t - \sqrt{1-\alpha_t} \cdot \epsilon)$ 
6    $f_\phi(\hat{x}_0) = \hat{J}_0 e^{-\phi_a \hat{D}_0} + \phi^\infty (1 - e^{-\phi_b \hat{D}_0})$ 
7    $\mathcal{L}$  = loss from Eq. 10
8   if  $\text{Optim}_{\text{start}} \geq t \geq \text{Optim}_{\text{end}}$  then
9     for  $i = 1 : N$  do
10       $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}$ 
11    end
12  end
13   $x_{t-1} \sim \mathcal{N}(\mu - s \cdot \nabla_{x_t} \mathcal{L}, \Sigma)$ 
14 end
```

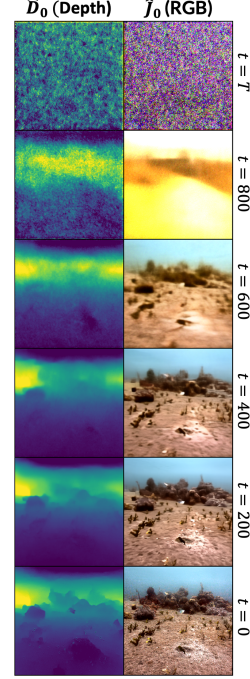


Fig. 4: Our algorithm. [Left] Detailed steps of our algorithm. [Right] Example of how \hat{J}_0, \hat{D}_0 change during the iterations.

with respect to ϕ which is used to update the parameters in the underwater model. In order to stabilize the optimization, we apply gradient clipping by value, to the gradients of x_t , and we run the optimization of ϕ only in some of the sampling steps (defined by the values $\text{Optim}_{\text{start}}$ and $\text{Optim}_{\text{end}}$), running N iterations of gradient descent in each step. When the gradient of x_t is applied in the sampling step, it is multiplied by a guidance scale s . We find that using a smaller scale for the depth channel leads to better results. This makes sense as while all channels are treated the same in the prior, the depth has a different role in the forward model (specifically it is used inside an exponent), and this can lead to a different gradient scale. More details on the implementation are given in the appendix.

5 Results

We use a prior on 256×256 images, with sampling time of about 3 minutes per image on an Nvidia A100 GPU. We present here selected results and analysis. Please refer to the appendix for a complete set.

5.1 Real-World Scenes

We present an extensive comparison on real-world linear images from the datasets SQUID [8], SeaThru [3], SeaThruNerf [32], and additional images acquired in different locations in the world, the Indian and Pacific Oceans, and Mediterranean, and Red Seas. Images are white-balanced as pre-processing. All runs use the same set of parameters. Fig. 1 shows several results of both image restoration and depth estimation.

We compare **image restoration** with the following methods: CWR [25], DM [53], FUnIE-GAN [28], GDGP [41], IBLA [42], MMLE [61], semi-UIR [27], Ucolor [33], USUIR [20], UW-Net [24], waternet [35], and USe-ReDI-Net [54]. For compactness, Fig. 5 summarizes the results for a chosen subset of methods. The complete comparison is shown in the appendix. Among previous methods, we found GDGP to be the most consistent. Our result recovers the full range of the scenes, specifically improving contrast of objects that are further away (note the zoomed-in objects in the far areas, e.g., the diver in row 1). Our method recovers vibrant and consistent colors also in further areas. For example, in rows 2,3,4 note that the background rocks and sand appear bluish in all methods except of ours.

We also compare to a method that restores the RGB using a fixed depth that is pre-estimated using an external SOTA method, *depth anything* (DA) [59]. We use the pre-estimated depth in our method’s pipeline instead of our gradually estimated depth. We term this method *DA-Osmosis*, and the results show that our method restores the far areas with better color, emphasizing the advantage of using our joint RGBD prior.

We compare **depth estimation** with GDGP [41], IBLA [42], unveiling [7], UW-Net [24], and monoUWnet [4], and depth anything (DA) [59]. In GDGP [41] and unveiling [7] we compute the depth from the output transmission maps. Our depth estimations have more details and better explain the scenes (Fig. 5).

The UIEB dataset [35] is sometimes used for quantitative evaluation of image restoration. However, it is collected from various sources and is not linear. Thus it is not suitable for physics-based methods like ours. In addition, the ‘ground truth’ of the UIEB dataset is chosen from results of previous methods and it is not the real ground truth. Fig. 6 shows four examples (out of many) where our result removes more water effects than the ground truth, even despite the non-linearity of the input.

5.2 Simulation

We conducted a simulation following [27, 34] using all the 1449 images from the NYU-v2 RGBD dataset [46] with randomly varying water parameters and generated the underwater images with the image formation model in Eq. 1. For fairness in comparison with other methods we applied $\phi_a = \phi_b$. The quantitative results are summarized in Table 1. Our method substantially outperforms other methods in the image restoration metrics of PSNR, SSIM, and LPIPS. We emphasize that our prior was **not** trained on this data, or indoor data at all. Moreover, two of the methods [27, 34] we outperform were trained on this data.

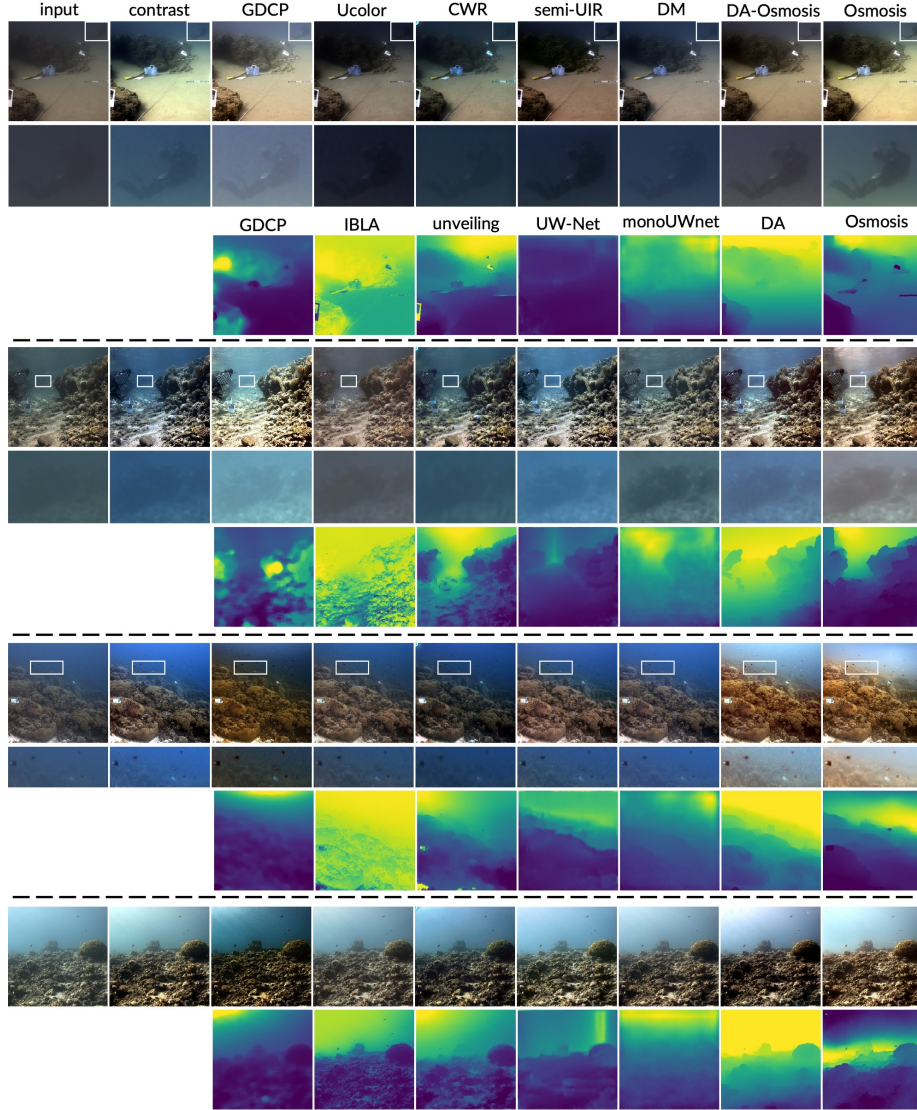


Fig. 5: Real-world restoration results. From left to right: white-balanced input, contrast stretch, GDCP [41], Ucolor [33], CWR [25], semi-UIR [27], DM [53], Depth Anything [59] - osmosis, osmosis (ours). Zoom-in colored rectangles emphasize far objects that have higher contrast in our results. **Real-world depth results.** From left to right: GDCP [41], IBLA [42], unveiling [7], UW-Net [24], monoUWnet [4], Depth Anything [59], Osmosis (ours). Our depth results are smoother and less affected by object gradients. **The reader is encouraged to zoom-in.**

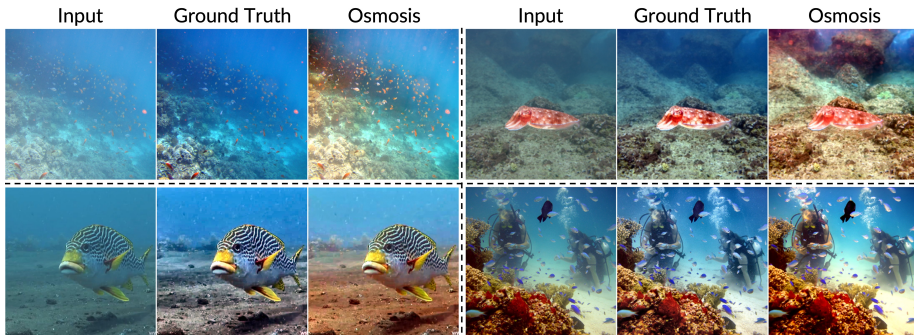


Fig. 6: Example results on UIEB [35], where our method yields **better results than the dataset’s ground truth. See for example the color of the sand, the orange color of the fish in the top-left, and the divers’ authentic skin color in the bottom-right.**

Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
contrast stretch	17.13	0.83	0.11
CWR [25]	16.93	0.79	0.20
DM [53]	17.41	0.82	0.12
FUnIE-GAN [28]	17.64	0.77	0.21
GDCP [41]	12.41	0.71	0.16
IBLA [42]	15.07	0.70	0.19
MMLE [61]	17.00	0.74	0.17
semi-UIR [27]	17.82	0.83	0.12
Ucolor [33]	17.92	0.83	0.10
USUIR [20]	16.76	0.80	0.18
UW-Net [24]	18.04	0.75	0.26
waternet [35]	17.27	0.82	0.11
unveiling [7]	16.34	0.79	0.18
ablation #1	21.09	0.86	0.09
ablation #2	<u>22.17</u>	<u>0.88</u>	0.07
ablation #3	22.00	<u>0.88</u>	0.06
Osmosis (ours)	22.74	0.89	0.06

Table 1: Quantitative comparison on the simulation. Our method achieves best scores in image restoration.

5.3 Ablation

To demonstrate the effect of all parts of our methods, we conduct an ablation study of the following variants: **1.** $\mathcal{L} = \mathcal{L}_{\text{rec}}$ (instead of Eq. 12); **2.** Removing weighting by \hat{D}_0 in Eq. 9; **3.** Same guidance scale s for all channels; **4.** $\phi_a = \phi_b$; **5.** $\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{val}}$ (remove \mathcal{L}_{avg} from Eq. 12); **6.** $\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{avg}}$ (remove \mathcal{L}_{val} from Eq. 12).

Numerical results on the simulation for variants #1-#3 are presented in Table 1. Ablations #4-#6 are shown only on real world images since in the simulation we do not use \mathcal{L}_{avg} , and $\phi_a = \phi_b$. Fig. 7 presents the results of all variants on one of the scenes. We see that the additional losses are important to

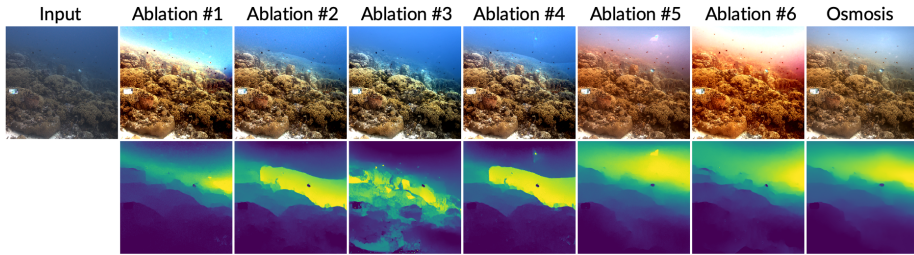


Fig. 7: Ablation. Ablations #1,#5,#6 show the importance of the losses we added to the reconstruction loss. Without \mathcal{L}_{val} (#6), the colors tend to “explode” and oversaturate. Without \mathcal{L}_{avg} (#5) the colors sometimes skew towards pink/purple. In #2, the further areas are not restored well, because the loss is not weighted by the depth D . In #3 the guidance scale is the same for the RGB and depth channels. This harms the depth reconstruction. In #4 we set $\phi_a = \phi_b$ (separately per color channel). Since this is an inaccurate model, the restoration in further areas is harmed.

prevent color saturation and shift. Increasing the loss weight with depth improves restoration in further areas. Separating guidance between the RGB and depth channels improves depth reconstruction. Setting $\phi_a \neq \phi_b$ extends the range of the restoration.

6 Discussion

In this work we demonstrated how to harness the strength of a new RGBD diffusion prior to achieve state-of-the-art results on underwater image restoration. To do this we solved several challenges: i) because of lack of clean underwater image data we use datasets of scenes in air; ii) we notice that the color prior does not suffice to guide restoration, therefore, we add the scene depth to the prior; iii) We use the physical image formation model to guide restoration, and also estimate the water parameters in the process. This results in the most comprehensive single underwater image restoration method to-date. It does not train on any underwater images and therefore does not overfit to any. It gives higher weight to further objects and therefore is superior in reconstructing all the details of the scene. Jointly solving for the depth results in excellent depth estimation from monocular images.

There is a domain gap between the prior (trained on in-air data) and underwater data. This is intentional. The beauty of our method is that by incorporating the underwater model in the sampling it succeeds even *without training* on underwater data (that is very hard to obtain). Thus, it is not specialized for a certain type of water or objects, and it learns the color distribution of natural images not degraded by water.

Like all diffusion models with U-Nets, our method is limited by a fixed resolution and long running time. This could potentially be improved using other network architectures and sampling methods. In the future, we plan to explore this direction and additional tasks that can exploit the RGBD prior.

References

1. Aali, A., Arvinte, M., Kumar, S., Tamir, J.I.: Solving inverse problems with score-based generative priors learned from noisy data. arXiv preprint arXiv:2305.01166 (2023)
2. Akkaynak, D., Treibitz, T.: A revised underwater image formation model. In: CVPR. pp. 6723–6732 (2018)
3. Akkaynak, D., Treibitz, T.: Sea-thru: A method for removing water from underwater images. In: CVPR. pp. 1682–1691 (2019)
4. Amitai, S., Klein, I., Treibitz, T.: Self-supervised monocular depth underwater. In: IEEE Int. Conf. Robotics and Automation (ICRA). pp. 1098–1104 (2023)
5. Balçilar, M.: Full dense depth map image for known positioned camera from lidar point cloud. <https://github.com/balcilar/DenseDepthMap> (2018)
6. Bansal, A., Chu, H.M., Schwarzschild, A., Sengupta, S., Goldblum, M., Geiping, J., Goldstein, T.: Universal guidance for diffusion models. In: CVPR. pp. 843–852 (2023)
7. Bekerman, Y., Avidan, S., Treibitz, T.: Unveiling optical properties in underwater images. In: ICCP (2020)
8. Berman, D., Levy, D., Avidan, S., Treibitz, T.: Underwater single image color restoration using haze-lines and a new quantitative dataset. IEEE TPAMI **43**(8), 2822–2837 (2020)
9. Chan, M.A., Young, S.I., Metzler, C.A.: Sud²: Supervision by denoising diffusion models for image reconstruction. arXiv preprint arXiv:2303.09642 (2023)
10. Choi, J., Kim, S., Jeong, Y., Gwon, Y., Yoon, S.: Ilvr: Conditioning method for denoising diffusion probabilistic models. arXiv preprint arXiv:2108.02938 (2021)
11. Chung, H., Kim, J., Kim, S., Ye, J.C.: Parallel diffusion models of operator and image for blind inverse problems. In: CVPR (2023)
12. Chung, H., Kim, J., Mccann, M.T., Klasky, M.L., Ye, J.C.: Diffusion posterior sampling for general noisy inverse problems. In: ICLR (2023), <https://openreview.net/forum?id=0nD9zGAGT0k>
13. Chung, H., Sim, B., Ryu, D., Ye, J.C.: Improving diffusion models for inverse problems using manifold constraints. NeurIPS **35**, 25683–25696 (2022)
14. Chung, H., Sim, B., Ye, J.C.: Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In: CVPR. pp. 12413–12422 (2022)
15. Chung, H., Ye, J.C.: Score-based diffusion models for accelerated mri. Medical image analysis **80**, 102479 (2022)
16. Daras, G., Shah, K., Dagan, Y., Gollakota, A., Dimakis, A.G., Klivans, A.: Ambient diffusion: Learning clean distributions from corrupted data. arXiv preprint arXiv:2305.19256 (2023)
17. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. NeurIPS **34**, 8780–8794 (2021)
18. Fei, B., Lyu, Z., Pan, L., Zhang, J., Yang, W., Luo, T., Zhang, B., Dai, B.: Generative diffusion prior for unified image restoration and enhancement. In: CVPR. pp. 9935–9946 (2023)
19. Feng, B.T., Smith, J., Rubinstein, M., Chang, H., Bouman, K.L., Freeman, W.T.: Score-based diffusion models as principled priors for inverse imaging. arXiv preprint arXiv:2304.11751 (2023)
20. Fu, Z., Lin, H., Yang, Y., Chai, S., Sun, L., Huang, Y., Ding, X.: Unsupervised underwater image restoration: From a homology perspective. In: Proc. AAAI Conf. on Artificial Intelligence. vol. 36, pp. 643–651 (2022)

21. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. *Int. J. Robotics Research (IJRR)* (2013)
22. Graikos, A., Malkin, N., Jojic, N., Samaras, D.: Diffusion models as plug-and-play priors. In: *NeurIPS* (2022)
23. Guo, L., Wang, C., Yang, W., Huang, S., Wang, Y., Pfister, H., Wen, B.: Shadowdiffusion: When degradation prior meets diffusion model for shadow removal. In: *CVPR*. pp. 14049–14058 (2023)
24. Gupta, H., Mitra, K.: Unsupervised single image underwater depth estimation. In: *ICIP*. pp. 624–628 (2019)
25. Han, J., Shoeiby, M., Malthus, T., Botha, E., Anstee, J., Anwar, S., Wei, R., Armin, M.A., Li, H., Petersson, L.: Underwater image restoration via contrastive learning and a real-world dataset. *Remote Sensing* **14**(17), 4297 (2022)
26. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239* (2020)
27. Huang, S., Wang, K., Liu, H., Chen, J., Li, Y.: Contrastive semi-supervised learning for underwater image restoration via reliable bank. In: *CVPR*. pp. 18145–18155 (2023)
28. Islam, M.J., Xia, Y., Sattar, J.: Fast underwater image enhancement for improved visual perception. *IEEE Robotics and Automation Letters* **5**(2), 3227–3234 (2020)
29. Jalal, A., Arvinte, M., Daras, G., Price, E., Dimakis, A.G., Tamir, J.: Robust compressed sensing mri with deep generative priors. *NeurIPS* **34**, 14938–14954 (2021)
30. Kawar, B., Elad, M., Ermon, S., Song, J.: Denoising diffusion restoration models. *NeurIPS* **35**, 23593–23606 (2022)
31. Kawar, B., Elata, N., Michaeli, T., Elad, M.: Gsure-based diffusion model training with corrupted data. *arXiv preprint arXiv:2305.13128* (2023)
32. Levy, D., Peleg, A., Pearl, N., Rosenbaum, D., Akkaynak, D., Korman, S., Treibitz, T.: Seathru-nerf: Neural radiance fields in scattering media. In: *CVPR*. pp. 56–65 (2023)
33. Li, C., Anwar, S., Hou, J., Cong, R., Guo, C., Ren, W.: Underwater image enhancement via medium transmission-guided multi-color space embedding. *IEEE TIP* **30**, 4985–5000 (2021)
34. Li, C., Anwar, S., Porikli, F.: Underwater scene prior inspired deep underwater image and video enhancement. *Pattern Recognition* **98**, 107038 (2020)
35. Li, C., Guo, C., Ren, W., Cong, R., Hou, J., Kwong, S., Tao, D.: An underwater image enhancement benchmark dataset and beyond. *IEEE TIP* **29**, 4376–4389 (2019)
36. Liu, N., Zhang, N., Shao, L., Han, J.: Learning selective mutual attention and contrast for rgb-d saliency detection. *IEEE TPAMI* (2021)
37. Lu, S., Guan, F., Zhang, H., Lai, H.: Underwater image enhancement method based on denoising diffusion probabilistic model. *J. of Visual Communication and Image Representation* **96**, 103926 (2023)
38. Murata, N., Saito, K., Lai, C.H., Takida, Y., Uesaka, T., Mitsufuji, Y., Ermon, S.: GibbsDDRM: A partially collapsed gibbs sampler for solving blind inverse problems with denoising diffusion restoration. In: *Int. Conf. on Machine Learning* (2023)
39. Özdenizci, O., Legenstein, R.: Restoring vision in adverse weather conditions with patch-based denoising diffusion models. *IEEE TPAMI* (2023)
40. Peng, L., Zhu, C., Bian, L.: U-shape transformer for underwater image enhancement. *IEEE TIP* (2023)
41. Peng, Y.T., Cao, K., Cosman, P.C.: Generalization of the dark channel prior for single image restoration. *IEEE TIP* **27**(6), 2856–2868 (2018)

42. Peng, Y.T., Cosman, P.C.: Underwater image restoration based on image blurriness and light absorption. *IEEE TIP* **26**(4), 1579–1594 (2017)
43. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. pp. 234–241 (2015)
44. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In: *CVPR*. pp. 22500–22510 (2023)
45. Saxena, S., Kar, A., Norouzi, M., Fleet, D.J.: Monocular depth estimation using diffusion models. *arXiv preprint arXiv:2302.14816* (2023)
46. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgb-d images. In: *ECCV*. pp. 746–760 (2012)
47. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: *Int. conf. machine learning*. pp. 2256–2265 (2015)
48. Sohn, K., Ruiz, N., Lee, K., Chin, D.C., Blok, I., Chang, H., Barber, J., Jiang, L., Entis, G., Li, Y., et al.: Styledrop: Text-to-image generation in any style. *arXiv preprint arXiv:2306.00983* (2023)
49. Song, J., Vahdat, A., Mardani, M., Kautz, J.: Pseudoinverse-guided diffusion models for inverse problems. In: *ICLR* (2023), https://openreview.net/forum?id=9_gsMA8MRKQ
50. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) *NeurIPS*. vol. 32 (2019)
51. Song, Y., Shen, L., Xing, L., Ermon, S.: Solving inverse problems in medical imaging with score-based generative models. *arXiv preprint arXiv:2111.08005* (2021)
52. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456* (2020)
53. Tang, Y., Kawasaki, H., Iwaguchi, T.: Underwater image enhancement by transformer-based diffusion model with non-uniform sampling for skip strategy. In: *Proc. ACM Int. Conf. Multimedia*. pp. 5419–5427 (2023)
54. Varghese, N., Kumar, A., Rajagopalan, A.: Self-supervised monocular underwater depth recovery, image restoration, and a real-sea video dataset. In: *ICCV*. pp. 12248–12258 (2023)
55. Vasiljevic, I., Kolkin, N., Zhang, S., Luo, R., Wang, H., Dai, F.Z., Daniele, A.F., Mostajabi, M., Basart, S., Walter, M.R., Shakhnarovich, G.: DIODE: A Dense Indoor and Outdoor DEpth Dataset. *CoRR* **abs/1908.00463** (2019), <http://arxiv.org/abs/1908.00463>
56. Voynov, A., Aberman, K., Cohen-Or, D.: Sketch-guided text-to-image diffusion models. In: *ACM SIGGRAPH 2023 Conference Proceedings*. pp. 1–11 (2023)
57. Wei, M., Shen, Y., Wang, Y., Xie, H., Wang, F.L.: Raindiffusion: When unsupervised learning meets diffusion models for real-world image deraining. *arXiv preprint arXiv:2301.09430* (2023)
58. Xian, K., Zhang, J., Wang, O., Mai, L., Lin, Z., Cao, Z.: Structure-guided ranking loss for single image depth prediction. In: *CVPR* (June 2020)
59. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth anything: Unleashing the power of large-scale unlabeled data. *arXiv preprint arXiv:2401.10891* (2024)
60. Yang, P., Zhou, S., Tao, Q., Loy, C.C.: PGDiff: Guiding diffusion models for versatile face restoration via partial guidance. In: *NeurIPS* (2023)

61. Zhang, W., Zhuang, P., Sun, H.H., Li, G., Kwong, S., Li, C.: Underwater image enhancement via minimal color loss and locally adaptive contrast enhancement. *IEEE TIP* **31**, 3997–4010 (2022)
62. Zhou, J., Yang, T., Zhang, W.: Underwater vision enhancement technologies: a comprehensive review, challenges, and recent trends. *Applied Intelligence* **53**(3), 3594–3621 (2023)

Appendix

In this appendix we give more details about the implementation- prior training and optimization, and provide multiple additional results. All figures are placed at the end of the document.

A Data Preprocessing and Training

Here we describe the different datasets used for training, and the preprocessing performed on each.

KITTI [21] - 23946 images. Depth information is from Lidar measurement and is sparse. We interpolate it into dense depth images using [5]. We then normalize by the maximum measurement value of 80 meters. When computing the loss we mask out depth pixels of remaining holes and non-depth information like the sky.

DIODE [55] - 16884 images. Depth information is from a high quality laser scanner. We normalize by the maximum value of the depth sensor which is 350. Valid depth masks are supplied, and used when computing the loss.

HR-WSI [58]- 20378 images. Computed with stereo cameras. The data is a relative disparity with values between 0 and 1, but without an absolute normalization value. We compute the depth as $1 - \text{disparity}$. Valid mask are provided and used when computing the loss.

Red-Web-S [36] - 2179 images - The depth information is computed from a model’s prediction and is already dense and normalized.

We crop and resize the images to get a size of 256x256. In KITTI we crop-out the upper part of the image which contain only sky to get a 256 height, and then use different random horizontal crops of 256. In all other datasets we resize the smaller dimension into 256, and crop the other dimension at the center. We perform additional data augmentations by horizontal and vertical flips.

B Implementation Details

We give a list of implementation details and specific values used in the experiments. These values are used throughout all experiments except for some different values in the simulation (stated in bold below).

1. Both in training and in sampling we use 1000 sampling steps between 0 to $T = 1$.
2. We used a linear schedule for the diffusion noise variance, in the following range:

$$\alpha_t = 1 - \beta_t, \quad \beta_0 = 1e^{-4}, \quad \beta_1 = 2e^{-2}$$

3. We use a U-Net architecture which was suggested in [16], for 256x256 input. In order to handle RGBD data, we made modifications for the first and last convolution layers. The first convolution layer gets 4 channels as input instead of 3 channels, and the last convolution output is 4 channels instead of 3. These two layers are initialized at random before finetuning.

4. Before the depth map is used in the underwater model, it is linearly scaled from the range $[-1, 1]$ into the range $[0.56, 3.36]$ using the function:

$$g(\hat{D}_0) = 1.4 \cdot (\hat{D}_0 + 1.4)$$

In the **simulation**, this is mapped to a $[0, 1]$ range instead, using $g(\hat{D}_0) = 0.5 \cdot (\hat{D}_0 + 1)$. The model’s depth range is tuned as a standard hyperparameter and is the same for all the real-world images.

5. In the reconstruction loss, the weight is computed according to the same linearly scaled depth using a ‘stop-gradient’ operator.

$$g(\text{sg}(\hat{D}_0)) = 1.4 \cdot (\text{sg}(\hat{D}_0) + 1.4)$$

This is used both for real world and the simulation experiments.

6. For all real world experiments we use a guidance scale, separated to each of the RGBD channels, as following: red: 7, green: 7, blue: 7, depth: 0.9. In the **simulation** the values are: red: 4, green: 4, blue: 4, depth: 1.
7. The weights of the auxiliary losses are:

$$\lambda_v = 20, \quad \lambda_a = 0.5$$

In the **simulation** we do not use \mathcal{L}_{avg} :

$$\lambda_v = 40, \quad \lambda_a = 0$$

8. The thresholds used in the auxiliary losses are: $T_v = 0.7$, $T_a = 0.5$
9. The threshold of gradient clipping we used is 0.005. In the **simulation** The threshold of gradient clipping we used is 0.001.
10. We initialized the water parameters to:
 - (a) $\phi_a : 1.1, 0.95, 0.95$
 - (b) $\phi_b : 0.95, 0.8, 0.8$
 - (c) $\phi^\infty : 0.14, 0.29, 0.49$

In the **simulation** we use the simpler model where $\phi_a = \phi_b$, and initialize according to ϕ_a above, and ϕ^∞ is initialized to $[0.2, 0.4, 0.7]$.

11. The optimization schedule of ϕ was set to run from step $t = 0.7$ down to step $t = 0$, with 20 gradient descent iterations at each step.

$$\text{Optim}_{\text{start}} : 0.7, \quad \text{Optim}_{\text{end}} : 0, \quad N : 20$$

C Real World Results

In this appendix we present a total of **48** results of Osmosis on challenging real world scenes. For 16 real-world scenes (8 of them were presented in the main paper) we provide extensive comparisons with other methods in Figs. 8, 9, 10, 11. Additionally, we provide 16 additional scenes of real-world restored RGB and depth maps generated by Osmosis in Fig. 13. We also include results on additional 16 images in Fig. 16.

C.1 Full comparisons

In Figs. 8, 9, 10, 11 we present results of the complete suite of comparison methods on the all the real-world scenes presented in the main paper and on additional scenes: a) Input, b) contrast stretch, c) GDCP [41], d) IBLA [42], e) unveiling [7], f) UW-Net [24], g) waternet [35], h) Ucolor [33], i) MMLE [61], j) CWR [25], k) FUnIE-GAN [28], l) USUIR [20], m) semi-UIR [27], n) DM [53], o) DA-Osmosis [59], p) Osmosis (ours). For USe-ReDI-Net [54] there is no released code, therefore comparison is shown in Fig. 12 on the 3 linear scenes that were presented in [54].

C.2 More results

We show results on additional scenes from SQUID [8] and Seathru [3] in Fig. 13.

D Extended Ablations

In addition to the example provided in Fig. 7 of the main paper, three additional examples are presented here for the same ablation study. To demonstrate the effect of all parts of our methods, we conduct an ablation study of the following variants:

1. $\mathcal{L} = \mathcal{L}_{\text{rec}}$ (instead of Eq. 10).
2. Removing weighing by \hat{D}_0 in Eq.9.
3. Same guidance scale s for all channels.
4. $\phi_a = \phi_b$
5. $\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{val}}$ (remove \mathcal{L}_{avg} from Eq. 10).
6. $\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{avg}}$ (remove \mathcal{L}_{val} from Eq. 10).

Numerical results on the simulation for variants #1-#3 are presented in Table 1 in the main paper. Ablations #4-#6 are shown only on real world images since in the simulation we do not use \mathcal{L}_{avg} , and $\phi_a = \phi_b$. Fig. 14 presents the results of all variants on 3 scenes presented in Fig. 1 in the main paper. We see that the additional losses are important to prevent color saturation and shift. Increasing the loss weight with depth improves restoration in further areas. Separating guidance scales between the RGB and depth channels improves depth reconstruction. Setting $\phi_a \neq \phi_b$ extends the range of the restoration.

E Simulation

Fig. 15 depicts a method comparison on several scenes from the simulation. We can see that our method cleans the entire range of the image, while previous methods recover mostly the nearby areas.

F Consistency Analysis

As diffusion is a random process, we test our method’s consistency in two different experiments. Fig. 16 demonstrates a consistency experiment, where we ran our method on several images of the same scene from several viewpoints. We can see the resulting scene has similar appearance, and the depth is consistent.

Fig. 17 shows multiple results on the same image using a different random seed each time. We see that our results are very similar even with different seeds, showing the strength of our formulation and losses.

G Results on Haze

Fig. 18 shows our results on several haze images. Although our method was not designed and optimized for haze, the haze is removed in the restored images, the scenes have vivid colors and the depth maps are good. To run our method on these images we performed a “degamma” operation on them ($I^{2.2}$), and set $\phi_a = \phi_b$ constant for all color channels (one parameter instead of 6).

H Non-linear images

Every physics-based method expects as input linear images. We demonstrate the effect of inputting non-linear images in Fig. 19. We took the uncompressed non-linear images of the linear images used as input in Fig. 1 in the main paper, performed white-balance on them, and used them as input for our method. We can see in the results that the range of the restoration is smaller (i.e., the restoration stops at some point), for both color and depth. In addition, the colors are skewed.

I Failure Cases

Fig. 20 demonstrates 3 failure cases on real-world linear images. In example 1, there is an artifact in the top-right corner. In example 2 the restored colors are reddish and saturated. In example 3 there is a pinkish hue, especially in the horizon (“sky”) area.

We have noticed that, sometimes, the sky is not recognized as the most far area of the restored depth map (e.g., example 2 in Fig. 20). A possible reason is that during training, in some of the data, the sky is masked out and replaced by a value of 0. Although these pixels are masked out in the loss as well, they can still affect the prior through the input images. The effect can also have been amplified by the vertical flipping in our data augmentation, resulting in many cases where the top of the image has smaller depth.

J Negative Results and Abandoned Directions

We give here a list of directions that were tried and either gave worse results to the ultimate method we use, or did not show any promise in early experimentation and was therefore abandoned. The goal of this section is to share more information in order to give a bigger picture of our experimentation process. Many of the results here were not thoroughly examined and therefore should be treated as such.

1. We tried computing the gradient w.r.t x_0 rather than x_t for the likelihood score. This is similar to the Backward Universal Guidance suggested in [6]. This resulted in noisy images.
2. We ran a few initial experiments using Dynamic Guidance Scheme [56, 60], and changing the guidance scale during the sampling process as suggested in [6, 22]. We did not see significant improvement, but we believe further experimentation in this direction could be fruitful.
3. We tried to completely stop the guidance for the last few iterations as suggested in [56], because we noticed that in some cases the restored colors in early stages of the sampling looked better than in the end. This resulted in improvements to the colors, but the geometric details were less preserved.
4. We tried an inner optimization of \hat{x}_0 for several iterations at each as suggested in [6]. This resulted in the generated image being too noisy, which can perhaps be explained as the sampling process going too far off the manifold of the prior.
5. We tried running several iterations of sampling and ϕ optimization for the same time step [38], or per-step self-recurrence [6]. The aim of this feature is to keep the restored image committed to the prior and in addition strengthening the guidance. In our case, this resulted in somewhat distorted colors and in a worse estimation of the depth geometry.
6. We considered different annealing scheduling of the sampling time [22]. We did not perform vast experimentation with this. In our results the color restoration was more conservative, i.e. more stable but fixing less of the water effects.
7. We tried to clip the \hat{x}_0 image values in the forward model, instead of clipping the gradients. This lead to color saturation or de-saturation (colors getting closer to black). We found that gradient clipping in addition to the \mathcal{L}_{val} auxiliary loss gave better stability to the color restoration.



Fig. 8: Comparisons with all the methods on the scenes presented in Fig. 1 in the main paper. a) Input, b) contrast stretch, c) GDCP [41], d) IBLA [42], e) un-veiling [7], f) UW-Net [24], g) waternet [35], h) Ucolor [33], i) MMLE [61], j) CWR [25], k) FUnIE-GAN [28], l) USUIR [20], m) semi-UIR [27], n) DM [53], o) DA-Osmosis [59], p) **Osmosis (ours)**. Our restorations have the best colors and recovery range. **The reader is encouraged to zoom-in.**

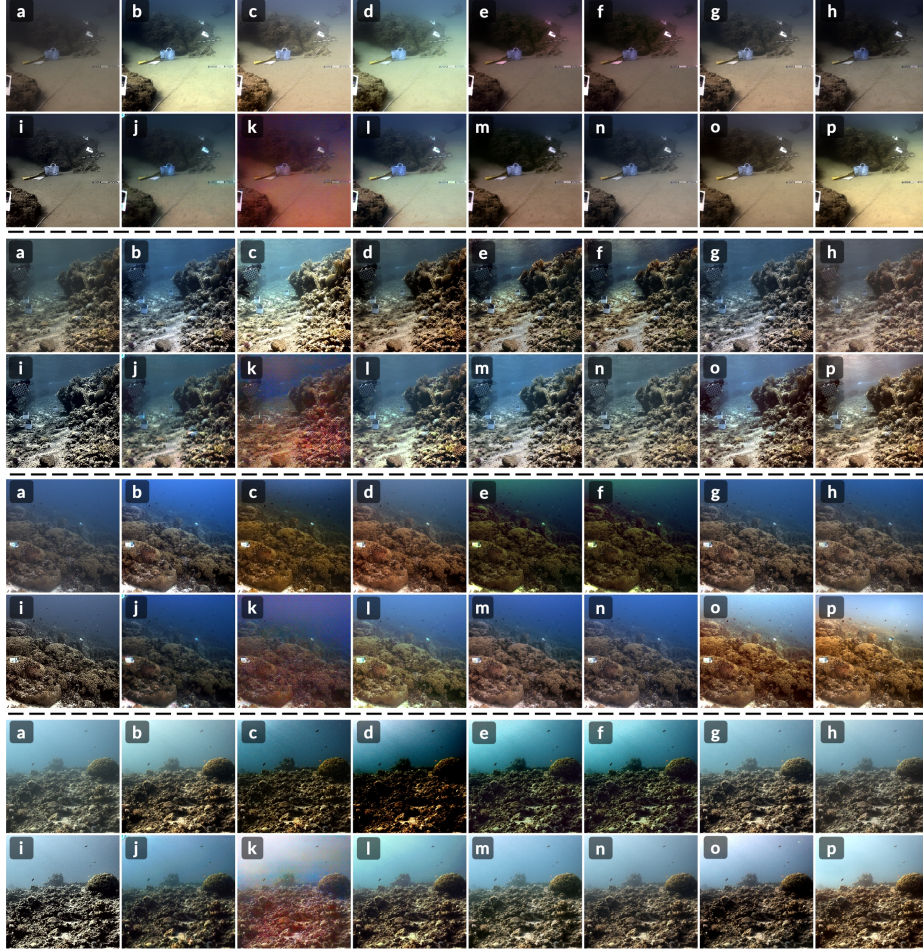


Fig. 9: Comparisons with all the methods on the scenes presented in Fig. 5 in the main paper. a) Input, b) contrast stretch, c) GDCP [41], d) IBLA [42], e) un-veiling [7], f) UW-Net [24], g) waternet [35], h) Ucolor [33], i) MMLE [61], j) CWR [25], k) FUnIE-GAN [28], l) USUIR [20], m) semi-UIR [27], n) DM [53], o) DA-Osmosis [59], p) **Osmosis (ours)**. Our restorations have the best colors and recovery range. **The reader is encouraged to zoom-in.**



Fig.10: Comparisons with all the methods on additional scenes. a) Input, b) contrast stretch, c) GDCP [41], d) IBLA [42], e) unveiling [7], f) UW-Net [24], g) waternet [35], h) Ucolor [33], i) MMLE [61], j) CWR [25], k) FUnIE-GAN [28], l) USUIR [20], m) semi-UIR [27], n) DM [53], o) DA-Osmosis [59], **p) Osmosis (ours)**. Our restorations have the best colors and recovery range. **The reader is encouraged to zoom-in.**

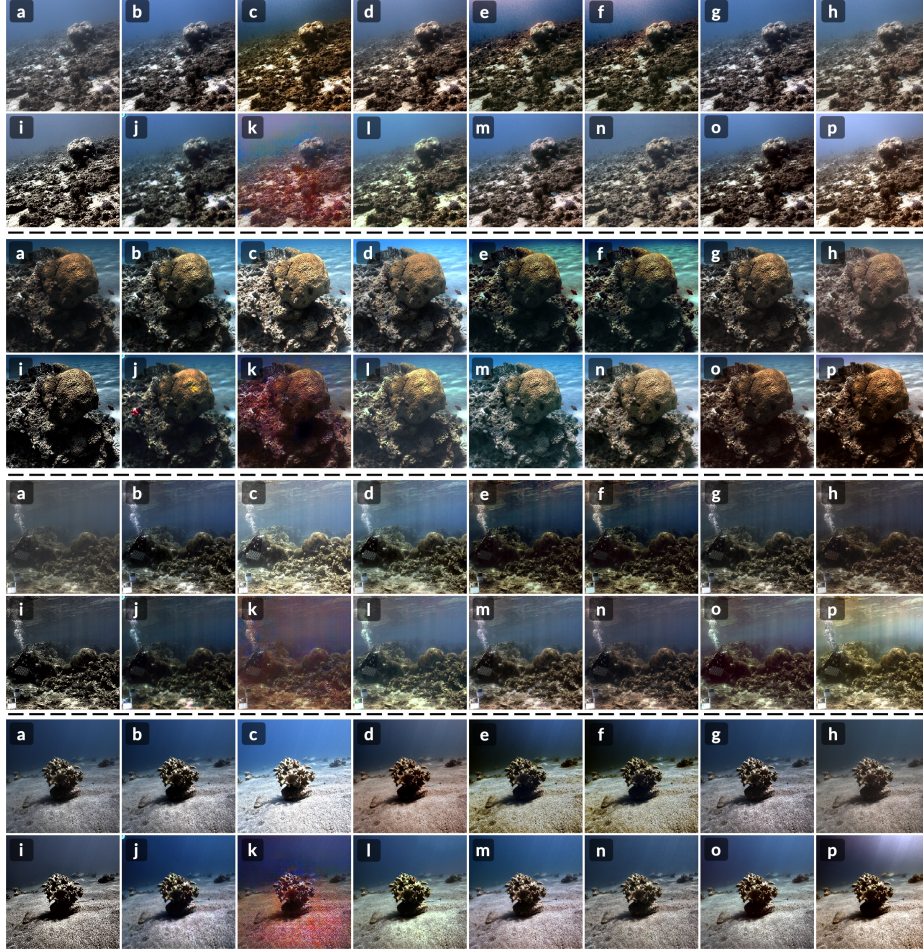


Fig.11: Comparisons with all the methods on additional scenes. a) Input, b) contrast stretch, c) GDCP [41], d) IBLA [42], e) unveiling [7], f) UW-Net [24], g) waternet [35], h) Ucolor [33], i) MMLE [61], j) CWR [25], k) FUnIE-GAN [28], l) USUIR [20], m) semi-UIR [27], n) DM [53], o) DA-Osmosis [59], **p) Osmosis (ours)**. Our restorations have the best colors and recovery range. **The reader is encouraged to zoom-in.**

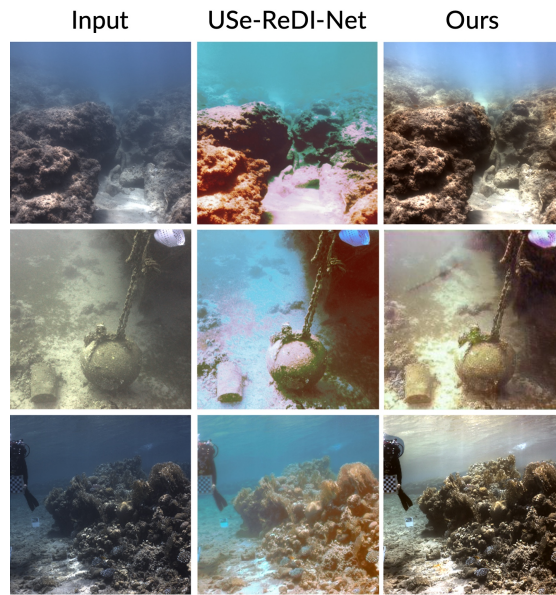


Fig. 12: Comparisons with [54]. Since there is no published code for [54] we can only compare with results published in the paper on linear scenes. Our restorations have consistent colors across the scenes.

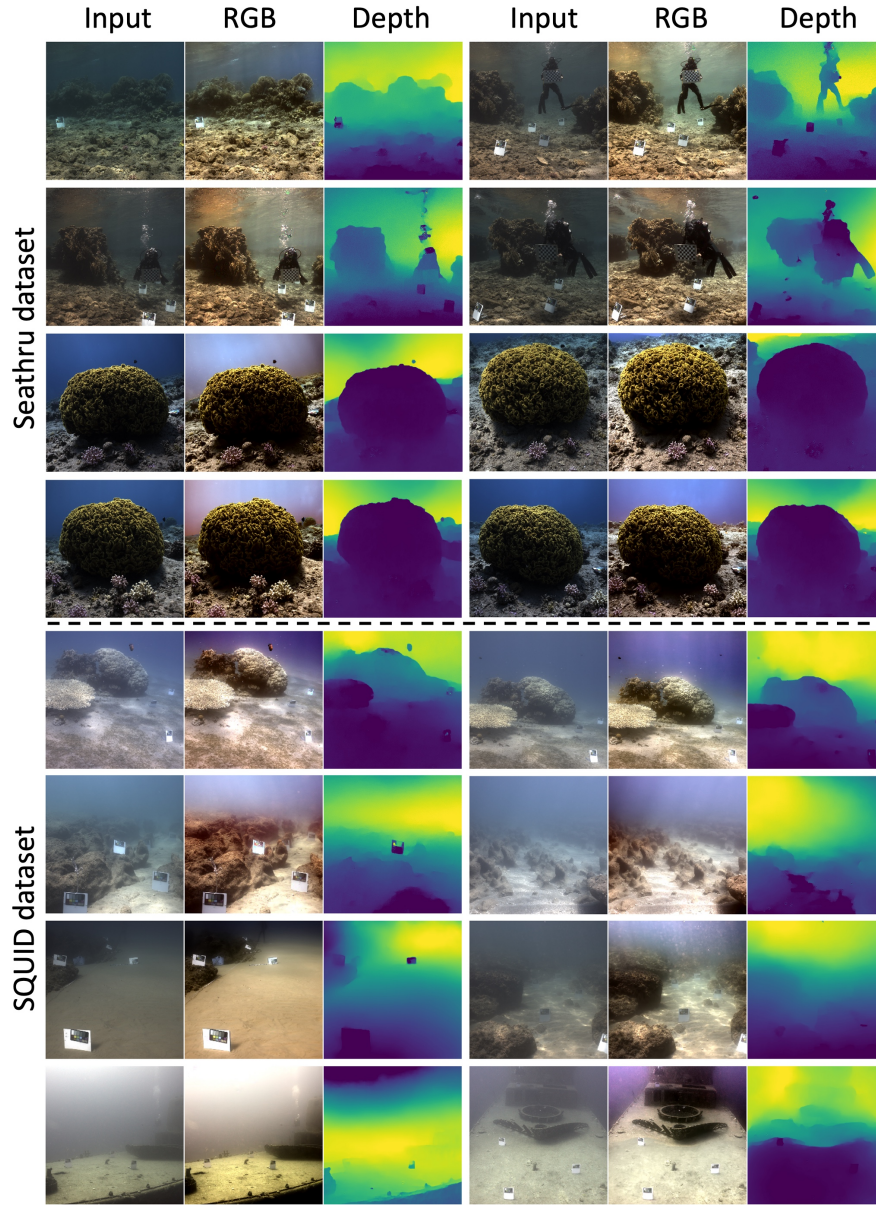


Fig. 13: Additional Results. Osmosis results on several scenes from Seathru [3] and SQUID [8] datasets. The restored image displays significantly less water effects, while maintaining consistent colors. We note that the estimated depth for very bright objects (e.g. the color boards) tends to be too close. This could be a result of the irregularity of having such objects within natural scenes, considering the training data. In any case this does not have a noticeable effect on the restored image. **The reader is encouraged to zoom-in.**

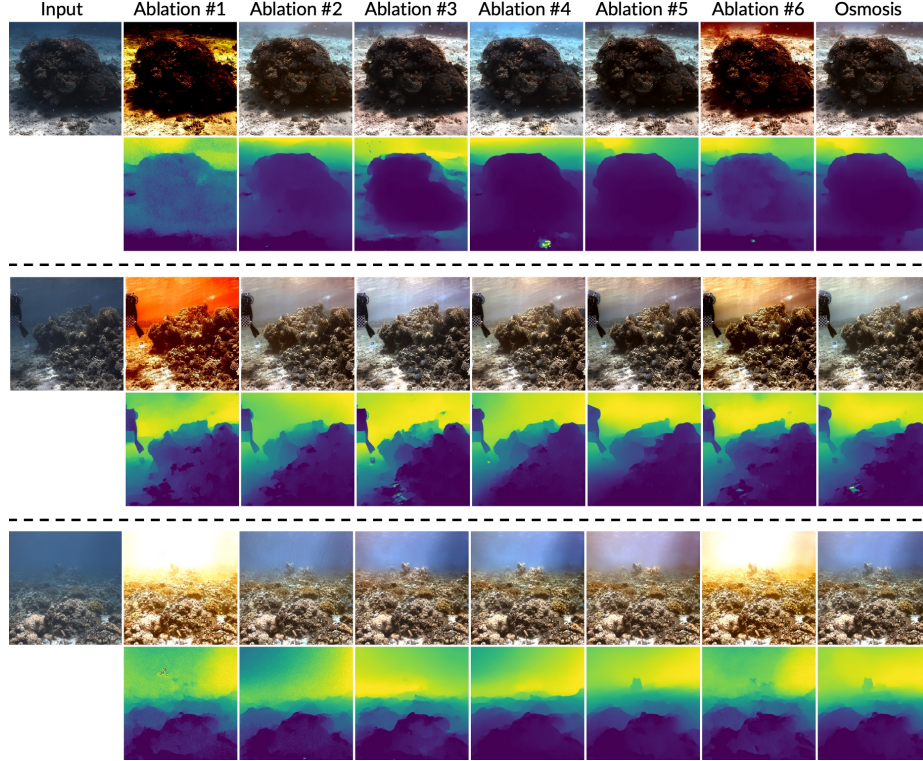


Fig. 14: Ablation results on several real-world images. Ablations #1,#5,#6 demonstrate the importance of the losses we added to the reconstruction loss. Without \mathcal{L}_{val} (#6), the colors tend to "explode" and oversaturate. Without \mathcal{L}_{avg} (#5) the colors sometimes skew towards pink/purple. In ablation #2, we can see that the further areas are not restored well, this is because the loss is not weighed by the depth D . In ablation #3 the guidance scale is the same for the RGB and depth channels. This harms the depth reconstruction. In ablation #4 we set $\phi_a = \phi_b$ (separately per color channel). Since this is an inaccurate model, the restoration in further areas is harmed.

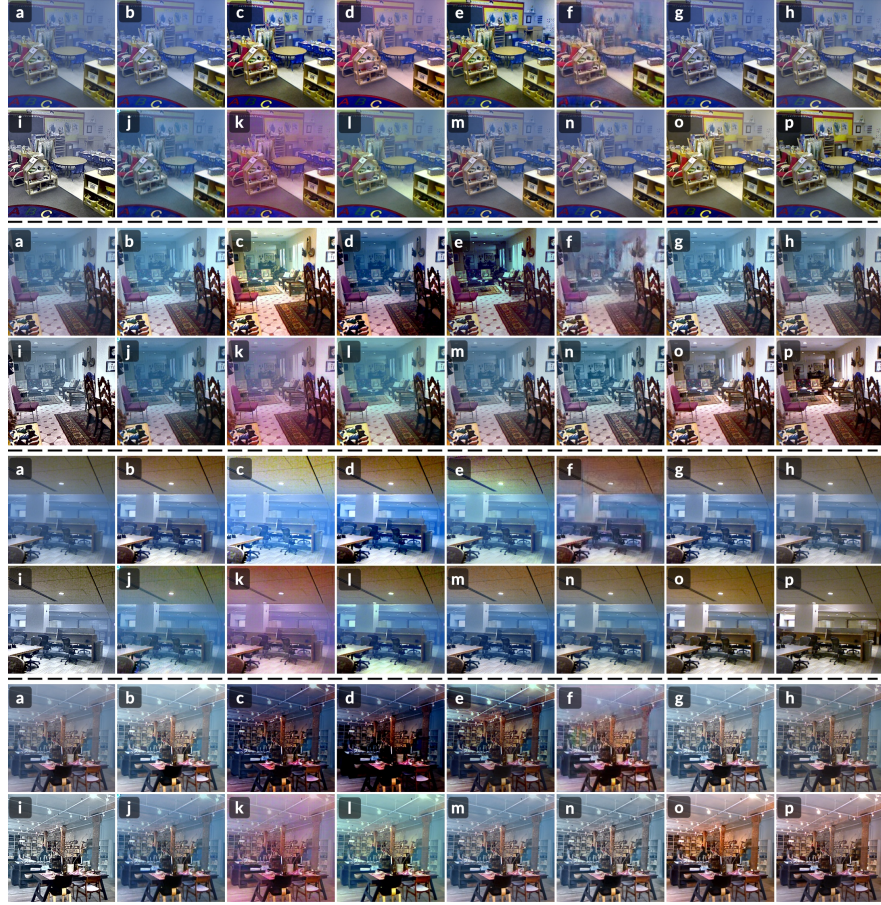


Fig. 15: Several color restoration results on simulated images. a) Input, b) contrast stretch, c) GDCP [41], d) IBLA [42], e) unveiling [7], f) UW-Net [24], g) water-net [35], h) Ucolor [33], i) MMLE [61], j) CWR [25], k) FUnIE-GAN [28], l) USUIR [20], m) semi-UIR [27], n) DM [53], o) **Osmosis (ours)**, p) Ground-truth. Our color restoration achieves highest PSNR, and cleans more areas in the scenes.

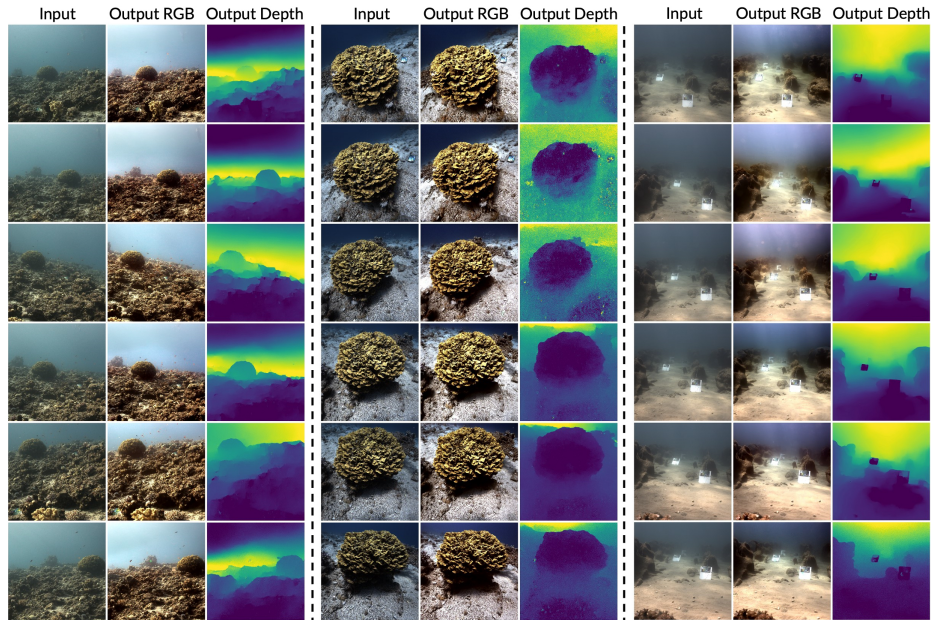


Fig. 16: Consistency of results. We ran our method on several images of the same scene from several viewpoints. The resulting restorations have similar appearances in terms of color, and the depth is consistent.

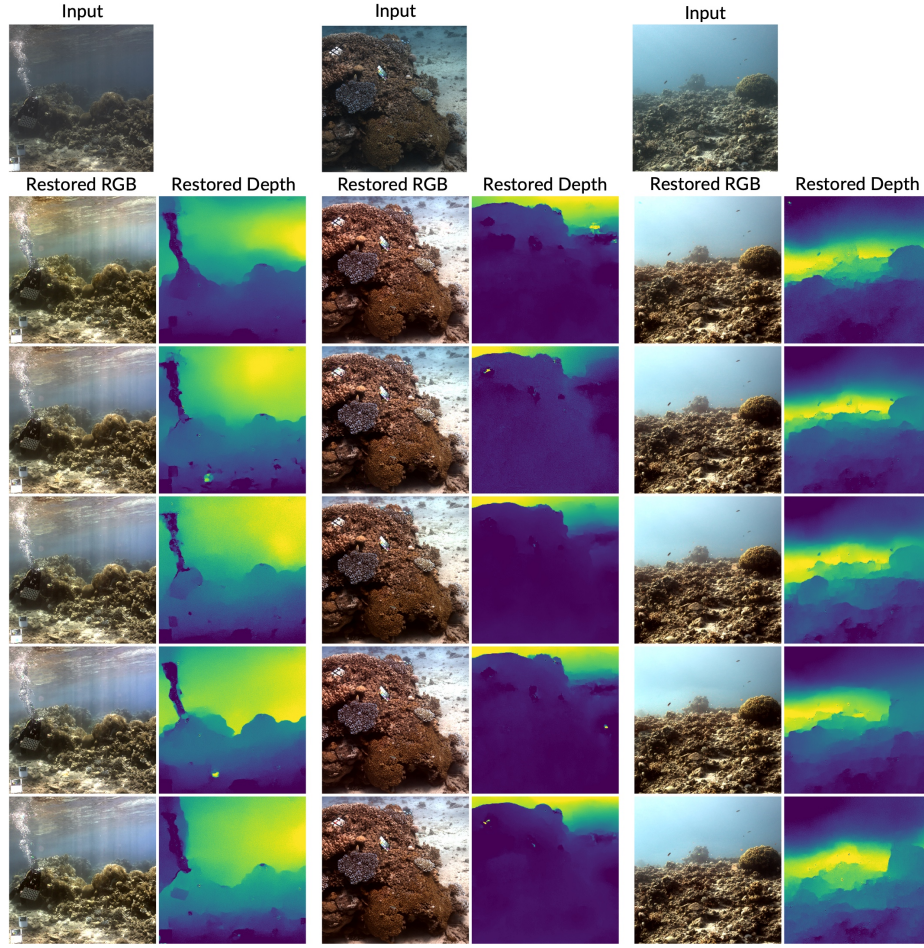


Fig. 17: Consistency of results given different random seeds for 3 different scenes. Each row presents results with different random seeds. We see that the results are very similar in both color and depth, showing the strength of our method's optimization procedure.



Fig. 18: Results on haze. Though not our main goal, we demonstrate that our method can potentially work also on haze images, when setting $\phi_a = \phi_b$ (identical for all color channels) in Eq. 8 in the main paper. Note the vivid colors and the depth maps.

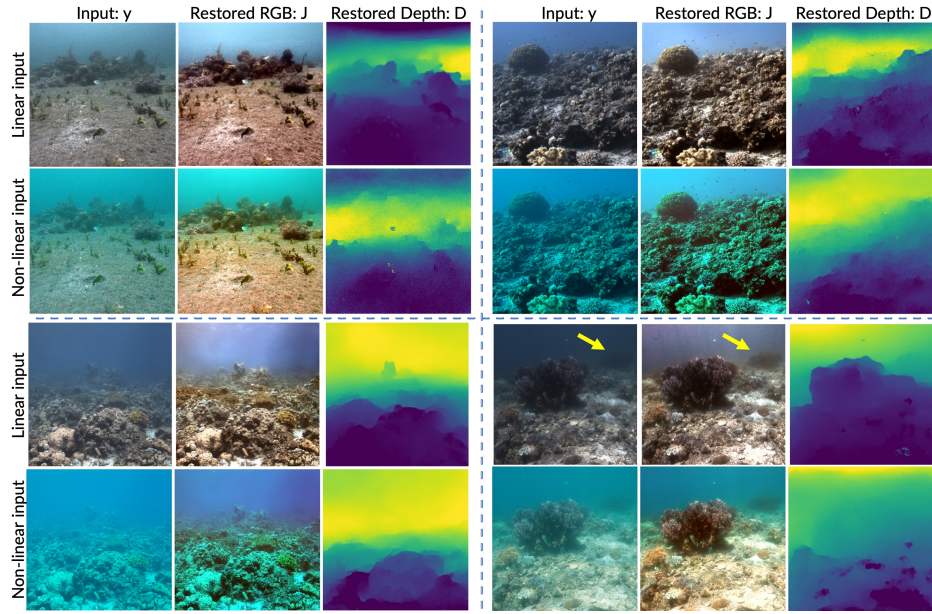


Fig. 19: Effect of non-linear input. We took the non-linear camera jpgs for the scenes presented in Fig. 1 in the main paper and ran our method after white-balancing. We see that the reconstruction range of both the colors and the depth maps is limited, and the restored colors are skewed.

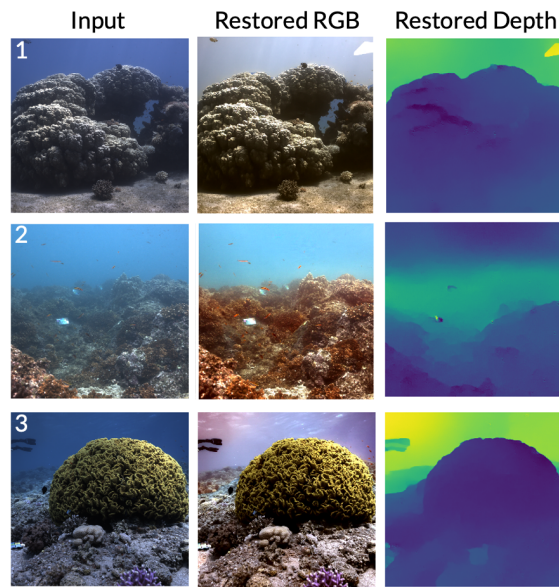


Fig. 20: Real-world failure cases. In example 1, there is an artifact in the top-right corner. In example 2 the restored colors are reddish and saturated. In example 3 there is a pinkish hue, especially in the horizon (“sky”) area.