2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

# DreamSea: Photorealistic 3D Underwater Terrain Generation by Latent Fractal Diffusion Models

Tianyi Zhang[1] Weiming Zhi[2,3,4] Joshua Mangelson [5] Matthew Johnson-Roberson[4]

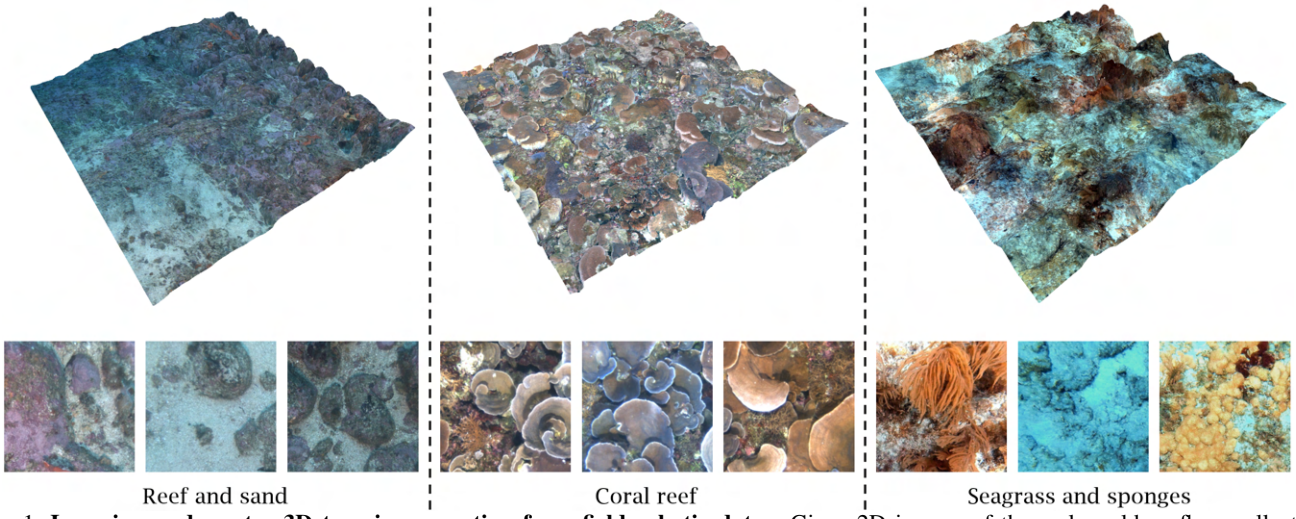


Fig. 1: **Learning underwater 3D terrain generation from field robotic data:** Given 2D images of the real world seafloor collected by robots, DreamSea distills 3D geometry and semantic information from visual foundation models and trains a diffusion model that generates realistic 3D underwater scenes conditioned on latent embeddings from a fractal process. All images and maps shown above are synthesized with DreamSea.

***Abstract*— This paper tackles the problem of generating representations of underwater 3D terrain. Off-the-shelf generative models, trained on Internet-scale data but not on specialized underwater images, exhibit downgraded realism, as images of the seafloor are relatively uncommon. To this end, we introduce DreamSea, a generative model to generate hyper-realistic underwater scenes. DreamSea is trained on real-world image databases collected from underwater robot surveys. Images from these surveys contain massive real seafloor observations and covering large areas. We extract 3D geometry and latent embeddings from the data with visual foundation models, and train a diffusion model that generates realistic seafloor images in RGBD channels, conditioned on novel fractal-distribution-based latent embeddings. We then fuse the generated images into a 3D map, building a 3D Gaussian Splatting (3DGS) model supervised by 2D diffusion priors which allows photorealistic novel view rendering. DreamSea is rigorously evaluated, demonstrating the ability to robustly generate large-scale underwater scenes that are consistent, diverse, and photorealistic. Our work drives impact in underwater robotics, and in particular, underwater robot simulation.**

## I. Introduction

Scene generation is widely studied today, with deep neural networks capable of creating realistic 3D environments trained on large-scale visual data. This technology has a significant impact across various fields, including the film and gaming industries, as well as robotics and autonomous vehicle simulations. In this paper, we explore the application of deep generative models to the unique setting of underwater environments. Without sufficient data and annotations, the following questions for underwater scene generation remain open:

- What kind of data can we use to train an underwater generative model?
- How can we train the underwater 3D generative model without 3D scans?
- How can we control the sampling process while data come with no captions or annotations?
- How can we generate underwater terrain with natural-looking variation in appearance?
- What techniques can we use from off-the-shelf 3D generative models, and what is lacking in current open-source models?

In this work, we tackle the problem from the perspective of robot perception. Underwater robots and autonomous underwater vehicles (AUVs) are designed to travel long distances under the sea, maintaining altitude and route to survey the designated area autonomously. Compared to typical images and videos on the Internet, underwater robotic images cover much larger areas of the terrain. However, the massive amounts of data collected by underwater robots

[1]Aurora Innovation, USA.
[2]School of Computer Science, The University of Sydney, Australia.
[3]Australian Centre for Robotics, Australia.
[4]College of Connected Computing, Vanderbilt University, TN, USA
[5]Department of Electrical and Computer Engineering, Brigham Young University, UT, USA.

present unique challenges: It is difficult to acquire 3D information directly from sensory streams, as depth sensors and LiDARs commonly do not work well underwater. In addition, natural water bodies are highly dynamic, and visibility is low as a result of light scattering and absorption in the medium. Therefore, Structure-from-Motion (SfM) [1] and Simultaneous Localization and Mapping (SLAM) [2] solutions have unstable performance. As a result, a significant amount of robotic data comes with no camera poses, and the cost of expert annotation is extremely high.

This paper introduces *DreamSea*, a diffusion-based generative model that can infinitely generate photorealistic 3D underwater scenes. **DreamSea is trained on RGB images captured by underwater robots without any 3D sensory information, SfM poses or human annotations.** After training, scenes generated by DreamSea are spatially consistent in geometry with natural-looking variations in appearance. The contributions of this paper are as follows:

1) A novel approach that leverages a *fractal* distribution of latent embeddings to control the appearance of generated terrains;
2) Integration of visual foundation models (VFMs) on unseen underwater images to exploit semantic and 3D geometric information for scene generation; and
3) A pipeline that integrates developments in image diffusion, inpainting, VFMs and 3DGS [3], allowing the generation of photorealistic 3D terrains from unannotated images.

## II. Related Work

### A. Procedural Terrain Generation

Early studies on procedural terrain generation focus on generating elevation maps that resemble the 3D structure of real-world terrain [4]. In particular, explicit mathematical models such as fractional Brownian motion (fBm) [5], the diamond square algorithm [6], and Perlin noise [7] are commonly used to approximate natural variations. Modern approaches have enabled the generation of 3D scenes consisting of a variety of assets procedurally and rendered with photorealistic quality [8]. Similar procedural strategies have also been applied to generate room layouts [9], object-level [10] layouts and scenes [8]. However, those modern approaches are based on pre-modeled 3D assets. While it is feasible to specify these assets in advance for commonly seen objects and scenes, e.g. indoor environment, this is not the case for unseen environments such as the deep sea.

### B. Deep Generative Models

Given an image dataset, an image generation model learns the distribution of this dataset. Unseen image samples can be generated as samples drawn from this distribution. Early techniques such as Variational Autoencoders (VAEs) [11] and Generative Adversarial Networks (GANs) [12] are able to generate realistic images. In recent years, models such as DDPM [13], Stable Diffusion [14], TRELLIS [15] and DiT [16] allow high-quality generation that can be conditioned on language inputs.

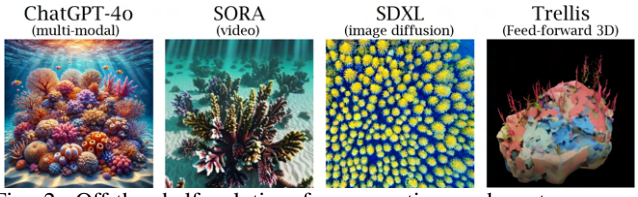

Fig. 2: Off-the-shelf solution for generating underwater scenes: Generalist generative models [14], [15] are able to generate scenes with diverse appearances, but present heavy artificial effects even though prompted with the "photorealistic style" keyword.

Some of these technologies have also led to commercialized models such as ChatGPT and SORA. While these models are capable of creating arbitrary scenes, we find, empirically, that the quality of generated underwater scenes is significantly lower than other more common environments. It can be hypothesized that the training data for underwater scenes is scarce and unbalanced. The development of specialized models with curated data for underwater scenes remains an open problem. Our DreamSea model leverages a DDPM [13] network with the RePaint [17] framework as a backbone generation and inpainting model.

### C. 3D Scene Representation and Generation

Three-dimensional scenes are often represented as point clouds, meshes or implicit functions, and generative models can be trained on 3D datasets such as ScanNet [18] to create 3D assets and scenes. Recent advancements in neural radiance fields (NeRFs) [19] techniques enable 3D scene reconstruction with photorealistic quality by optimizing directly over photometric loss. Building upon NeRFs, 3DGS [3] developed an explicit representation which enables efficient training and rendering at 100+ fps, making it a great fit for creating 3D scenes and visual simulation. In this work, 2D diffusion priors [20] are used to support generation of 3D assets.

### D. Visual Foundation Models

Underwater robotic field tests typically result in massive amounts of images that are extremely challenging to annotate and often lack 3D information. In this work, we leverage visual foundation models, which are trained on internet-scale data to infer semantic and geometric information by the images collected by our robots. CLIP [21] is a vision-language model (VLM) trained on internet-scale image-caption pairs and generalizes to unseen images. DINOv2 [22] is another foundation model that encodes an RGB image in a vector representation. In this work, we train the image diffusion model conditioned on DINO v2 representations, so the diffusion can be controlled in the latent space. Depth Anything v2 [23] is a depth foundation model that predicts depth from RGB images. In many cases this is used to generate RGB+Depth (RGBD) images from RGB image inputs.

## III. DreamSea

At the center of DreamSea is a terrain generation model that varies in spatial coordinates. This model can then generate a set of consistent images spanning a desired spatial

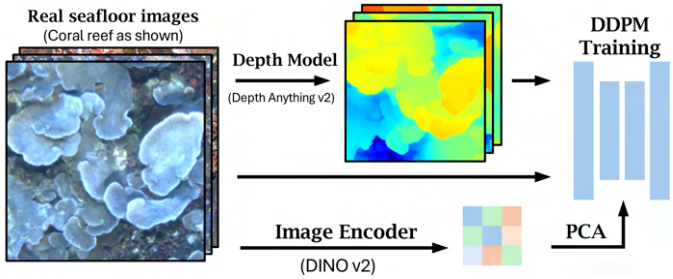

Fig. 3: **Overview of Training:** Given RGB-only images collected from underwater surveys, we generate depth channels and embeddings with visual foundation models [23], [22]. A DDPM network is then trained with an RGBD image as input conditioned on embeddings.

region, which can be used to construct 3DGS representations. Particular care needs to be taken to ensure that the generated images reflect both the biological and landscape diversity of marine environments, while being logically consistent across space.

This section elaborates on the design consideration and methodology details of DreamSea, and is structured as follows. In section III-A, we outline the extraction of relative depth from diverse underwater data from different expeditions. In section III-B, we introduce our diffusion-based generative model that is conditioned on *zero-shot visual features*, enabling the controlled generation on varied underwater environments. In section III-C, we introduce our novel fractal-based generation approach, which enables a set of spatially *consistent* underwater images to be generated and allows explicit control of the diversity of the generated terrain. Finally, in section III-D, we leverage the terrain generated by our generative model to construct a 3DGS representation supervised by the 2D diffusion prior. An overview of training and generation procedure are outlined in Figure 3 and 4 respectively.

### A. 3D Structure from Depth Foundation Model

To build more consistent 3D structures underwater, we seek to incorporate depth into the diffusion-based generative model. This, however, can be challenging. While traditional 3D reconstruction and mapping methods such as SfM and SLAM have been demonstrated on underwater data, the community struggles to scale up the application of these methods due to challenging underwater environments. These challenges often manifest via low visibility, dynamic surroundings, heavy motion blur under low light, and different sensor set-ups between expeditions to collect data. In this paper, we use the depth foundation model, Depth Anything v2 [23], to generate a depth map from 2D image data. Depth foundation models are good at predicting the relative depth distribution in single frames. We normalize this prediction to $[0, 1]$. In this work, we consider depths up to a scale factor, and do not require absolute metric depth. The metric scale can be recovered with additional sensors or classic stereo-matching methods. The estimated depths are used as additional depth channels for the real-world RGB training data.

### B. Conditional Diffusion on Zero-shot Features

Underwater robotic images do not come with captions. Additionally, annotating underwater data is also exceedingly challenging and requires a massive expert-level effort. Relying on manual labels would both be costly and difficult to scale. In light of this, we leverage the foundation visual model, DINOv2 [22], to extract zero-shot features from underwater images: for the image data set, we first generate DINO v2 features and then apply Principal Component Analysis (PCA) on the feature set to project high-dimensional features to the low-dimensional space. This reduced dimensional feature vector then acts as a descriptor of the contents within the image. Similar ideas have been explored in LangSplat [24] in which a Variational Autoencoder (VAE) [11] is trained to project CLIP [21] features onto a low-dimension space. Early work by Zhang et al. [25] takes a similar approach on seafloor mapping data with self-supervised training. However, here, by integrating foundation models, we are not required to train large neural networks from scratch to extract features, and can instead apply weights pre-trained on Internet-scale data. After obtaining a reduced-dimensional feature vector for each image, we train a diffusion model conditional on feature vectors, to generate both RGB and depth images. Let us denote the feature vector as

$$\boldsymbol{\phi} \leftarrow \texttt{PCA}(\texttt{DINOv2}(\mathbf{I})), \tag{1}$$

where $\mathbf{I}$ is an image and $\texttt{PCA}(\texttt{DINOv2}(\cdot))$ indicates applying PCA to the feature vector outputted by the DINO model, reducing dimensionality. During inference, our conditional generative model can be expressed as, $\mathbf{I} \sim P(\mathbf{I}|\boldsymbol{\phi})$, where $\boldsymbol{\phi}$ is a visual feature vector we condition upon. Generating spatially-consistent and yet diverse landscape images, requires controlling the evolution of $\boldsymbol{\phi}$ over the spatial domain, which alters the generative distribution of the terrain.

### C. Fractal Latent Terrain Generation

An inherent property of naturally-occurring terrains is that coordinate points that are close in geometric distance should have similar attributes. The spatial distribution of natural terrain is often modeled using fractal processes to approximate natural-looking variations. We imbue this inductive bias into DreamSea through a novel **latent fractal framework**, which assumes that the latent vectors over the spatial domain follow fractal processes.

We begin by initializing the latent vectors at the corners of an arbitrary square region for which we seek to generate terrain. We seek to sample a latent function $\boldsymbol{\Phi} : \mathbb{R}^2 \rightarrow \mathbb{R}^d$, where $d$ is the dimensionality of the latent vector after PCA reduction. Specifically, $\boldsymbol{\Phi}(\cdot)$ outputs a latent vector $\boldsymbol{\phi}$ for a given coordinate $(x, y)$, which can then be used to control the image generation.

The latent function can be seen as a sample from a fractal process, generated from the *Diamond-Square* Algorithm (Figure 5) applied to estimate the function output over a dense grid that covers the desired region. Here, the outputs are estimated recursively through a recursive two step

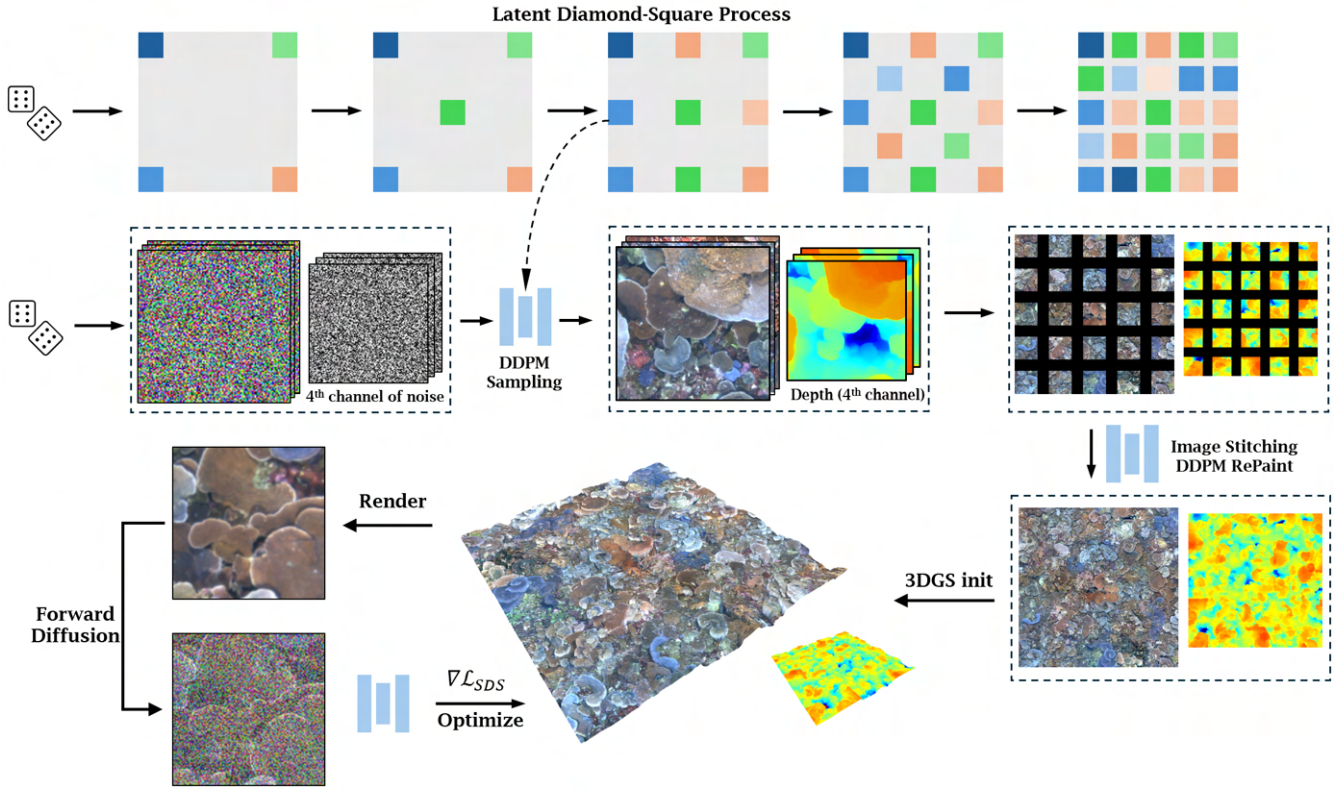


Fig. 4: **Overview of Generation:** Our approach generates fractal embedding with the diamond-square method first, then generates images conditioned on these embeddings. We use RePaint [17] to stitch the images together into a dense RGBD map. The RGBD map can be converted into a 3D point cloud and initialized as a 3DGS model [3]. The 3DGS model is further refined with 2D diffusion priors using Score Distillation Sampling (SDS) loss allowing realistic rendering from novel views.

process. First, in the *diamond step* we estimate the function value at the spatial mid-points of each square regions using the four corners of each square - forming four new diamonds.

Next, we apply a *square step*, to estimate the mid-points of diamond regions from the corner points of each diamond — forming squares that subdivided the original square. In each step, we compute the latent vector values at the centers of square and diamond shape patterns as the mean of the corner points of the regions plus some random noise. Let us denote the set of vertices of a square or diamond shape as the set $K$, and the center point of the square or diamond as $\mathbf{r}_c$, the latent vector value at the center is given by

$$\mathbf{\Phi}(\mathbf{r}_c) = \frac{1}{|K|} \sum_{\mathbf{r} \in K} \mathbf{\Phi}(\mathbf{r}) + s\boldsymbol{\sigma}, \qquad \boldsymbol{\sigma} \sim \mathcal{N}(0, \mathbf{I}). \tag{2}$$

Here, $s$ is a scaling factor that controls the variability of the landscape. This factor $s$ is gradually decayed. Therefore, starting with latent vector values at the vertices of a square, we can recursively estimate latent vector values over the entire square region.

A single iteration of this process, along with illustrated vertices, is shown in Fig. 5. The end result of this step is a 2D spatial field of latent fractal embeddings that can be used to conditionally generate a set of images with strong spatial dependency. After each iteration, $s$ will be multiplied by a dampening factor $ds \in [0, 1]$. To accomplish this, we train a diffusion model using RGB images from real

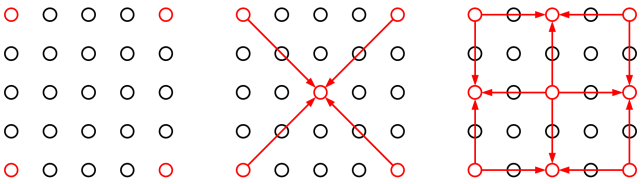


Fig. 5: The Diamond-Square algorithm, which recursively interpolates on a spatial grid, is used to generate latent embeddings. The red arrows start from the vertices of the existing square and diamond shapes from the previous iteration, and point towards the new center points.

underwater imagery augmented with depth generated using Depth Anything v2 [23]. The resulting model is used to generate an RGBD image for each vertex in the spatial latent field and then RePaint [17] is used to in-fill any gaps between each pair of neighboring images, to form a spatially consistent map in the form of an RGBD point cloud.

Here, we highlight that the function of images over the 2D spatial domain is drawn from a *doubly stochastic process*. The set of generated images, $\{\mathbf{I}_\mathbf{x}\}_{\mathbf{x} \in \mathbb{R}^2}$, can be considered as a function drawn from the conditional diffusion model, which itself is dependent on a latent function, $\mathbf{\Phi}(\mathbf{x})$, drawn from a fractal process, governed by the scale factor $s$. Specifically,

$$\{\mathbf{I}_\mathbf{x}\}_{\mathbf{x} \in \mathbb{R}^2} \sim \underbrace{P(\mathbf{I}|\mathbf{\Phi}(\mathbf{x}))}_{\text{Diffusion Model}}, \quad \mathbf{\Phi}(\mathbf{x}) \sim \underbrace{P(\mathbf{\Phi}|s)}_{\text{Fractal Process}}. \tag{3}$$

We note that the doubly stochastic nature of our image generation enables highly diverse terrains to be generated.

### D. 3D Scene Generation via Gaussian Splatting

In this section, we convert the RGBD point cloud generated in the previous step into a geometrically-consistent 3DGS model that uses the generated images as a strong prior. The resulting model provides us with a 3D structure that is dense and allows for the generation of novel images from arbitrary viewing poses.

We begin by using the depth channels from the generated images to initialize 3D Gaussians following the default method [3]. Then we freeze the 3D positions of the Gaussian cloud and refine the appearance with 2D diffusion priors. Given a cloud of Gaussians $\mathbf{G}$ initialized, each Gaussian $g_i$ includes the following attributes: position $\mathbf{p}_i$, covariance $\Sigma_i$, opacity $\alpha_i$ and radiance $\mathbf{c}_i$, that $g_i = \{\mathbf{p}_i, \Sigma_i, \alpha_i, \mathbf{c}_i\} \in \mathbf{G}$. With a subset of Gaussians $\mathcal{N} \in G$ ordered along a camera ray, the pixel value in an image can be rendered from 3DGS models with the following rendering equation:

$$C = \sum_{i \in \mathcal{N}} \mathbf{c}_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{4}$$

Here $\mathbf{p}_i$ is initialized from the depths of the generated images. We use the *Score Distillation Sampling* (SDS) loss introduced in DreamFusion [20] to optimize the 3D Gaussians from 2D diffusion prior:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\mathbf{I}^r) \triangleq \mathbb{E}_{t,\epsilon} \left[ w(t) \left( \hat{\epsilon}(t) - \epsilon \right) \frac{\partial \mathbf{I}^r}{\partial \theta} \right] \tag{5}$$

here $\theta$ is the parameters of Gaussian cloud $\mathbf{G}$ to be optimized, $\mathbf{I}^r$ is the rendered image; $\hat{\epsilon}$ and $\epsilon$ are predicted noise and added noise; $t$ is the timestep in the diffusion process and $w(t)$ is the weighting function following the implementation in [20] (parameter $y$ and $\mathbf{z}_t$ in the original paper are omitted here for brevity).

## IV. Experiments

### A. Datasets

The results presented throughout the paper are trained on real-world data collected from four different locations with three different robot platforms, spanning a time from 2009 to 2024 (see Figure. 6). The *Scott Reef* and *Batemans datasets* were collected from 2009 to 2015 with a Seabed-class AUV, Sirius, which features a dual-hull design for stabilized imaging underwater. We post-process the raw images, hosted on Squidle.org, to have normal exposure. The *Hawaii dataset* was collected in April 2024 with an Iver AUV, the torpedo design allowed it to travel long distances and sample images from the seafloor. The *Florida dataset* was collected in August 2023 with a customized remotely operated vehicle (ROV) equipped with ZED cameras. Each location presents a unique benthic appearance and is reflected in our model.

### B. Implementation Details

Our model's implementation is adapted from DDPM networks. We train each model on a single NVIDIA RTX4090 GPU with 24GB VRAM for 2000 epochs, with a batch size of 12. Although the size of each data set differs, it usually takes $\sim$ 200 hours to train on a dataset with 10k images, at the resolution of $224 \times 224$. We use the first two main components from PCA results on DINO v2 embeddings. From our empirical study, we find it to be sufficient to describe the variation in appearance of underwater environments. This is consistent with the practice in [25], [24].

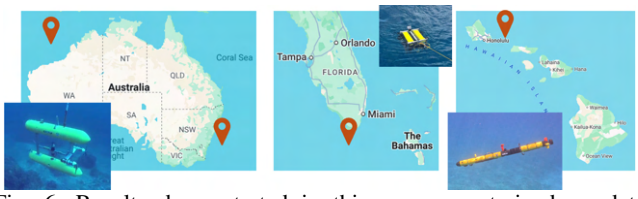


Fig. 6: Results demonstrated in this paper are trained on data collected from 4 different sites with 3 different robot platforms.

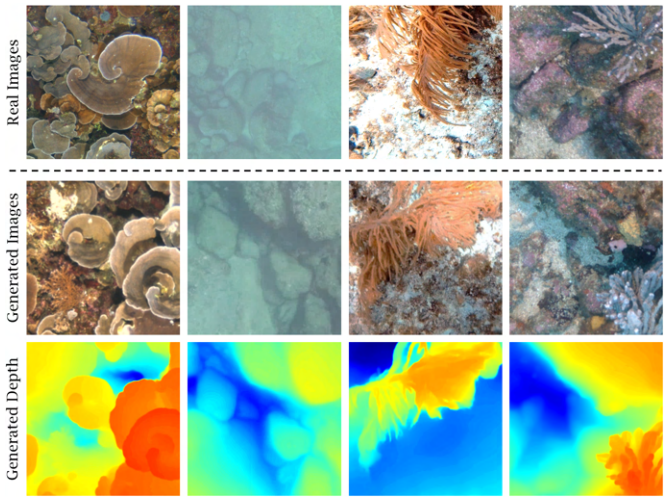


Fig. 7: Our diffusion model is able to output realistic images as well as relative depth estimation.

### C. Qualitative Evaluation

We train the diffusion model on the dataset collected from various locations capturing diverse underwater appearances. At a glance, the generated images closely resemble the real images from the training set well, as shown in Figure 7. The generated relative depth also aligns well with human perception, indicating that our training pipeline successfully learns the visual distribution of real underwater datasets and distills the 3D information from the depth foundation model. This model that generates realistic RGB-D data serves as the cornerstone of the rest of this work.

### D. Quantitative Evaluation

We also quantitatively evaluate our diffusion model on an unseen dataset from CoralNet using FID [26]. CoralNet contains coral observations collected by researchers around the world and does not include any training data we use. We compare our model trained on Scott Reef dataset which contains rich Coral observations with today's text-2-image models, gpt-image-1, gemini-2.0 and SDXL [14]. We acknowledge that CoralNet is a domain-specific dataset for marine science related research, which is

| | FID↓ |
|---|---|
| gpt-image-1 | 308.65 |
| gemini-2.0 | 295.12 |
| SDXL [14] | 383.06 |
| **ours** | **120.11** |

TABLE I: FID evaluated on an unseen CoralNet dataset.

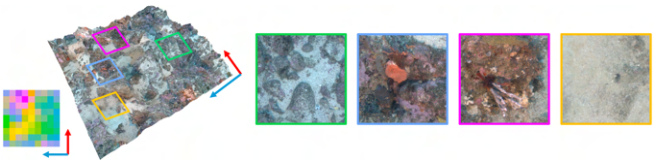

Fig. 9: Latent Controlled Generation on fractal embeddings with $s = 1.0$ and $ds = 0.6$.

different from the internet data that large models are trained on.

We tabulate FID scores of dreamsea relative to baselines. It shows that our model outperforms other generalist models, implying that our specialist model performs better on generating underwater terrains and generalizes beyond training data. We also notice that the FID score in this study is significantly higher than research in other domains. We believe that this is due to the high disparity between our training data and unseen testing data. Regardless of the high FID value, we show evidence of our model generating realistic results, even side-by-side compared with unseen dataset.

### E. Justification of High FID

In the qualitative evaluation, although our method performs better in terms of FID, the absolute value of FID is higher than normal. Here in Fig. 8 we show that regardless of high FID, our model is capable of generating visual content that resembles unseen real-world data, which means that our model can create generalizable results. Our hypothesis is that the high FID is caused by the diverse nature of real-world underwater images. We also identify that the difference in color and lighting effects can be a significant factor leading to diversified appearance and influencing the evaluation.

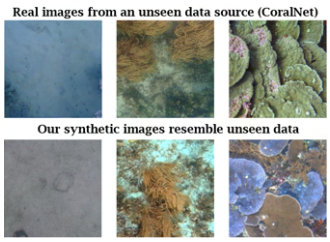


Fig. 8: DreamSea creates generalizable results.

### F. Latent Controlled Generation

Generating images and maps with latent embedding control plays a critical role in creating terrain with appearance aligned with human preference and natural variation. We show the 2D map generated from a fractal latent field. In Figure 9, where the latent field is generated with scaling factor $s = 1.0$ and dampening factor $ds = 0.6$. We observe that the stochasticity injected into the latent process visibly enhances the diversity of the generated terrain. The effects of different $s$ and $ds$ values in the fractal process are further investigated in IV-I.

We also demonstrate smooth image transitions over the latent space: Figure 10 shows diverse underwater scenes from different locations, in which we can see how the appearance of the images smoothly transits along axes and we can recognize how the content of the image shifts gradually as latent shifts. The interpolation demonstrated that latent embeddings from VFMs controls underwater image generation smoothly and can be well aligned with human perception. More example is shown in Figure 12.

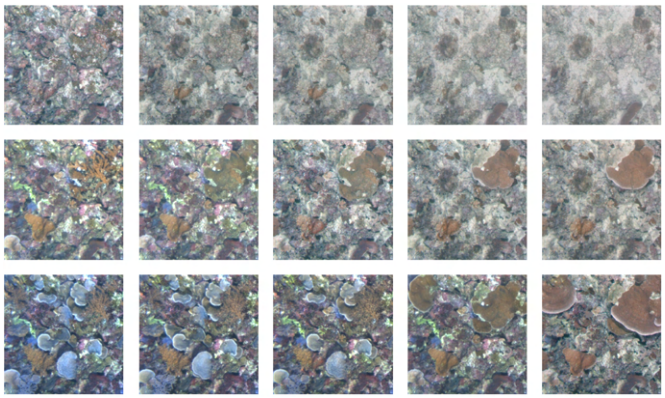

Fig. 10: Interpolating on 2D latent space: we generate diverse images conditioned latent embeddings interpolated in 2 directions, and can observe the appearance of generated images gradually transitioning from sand to reef to corals of different kinds.

### G. Ablation: Image stitching by inpainting

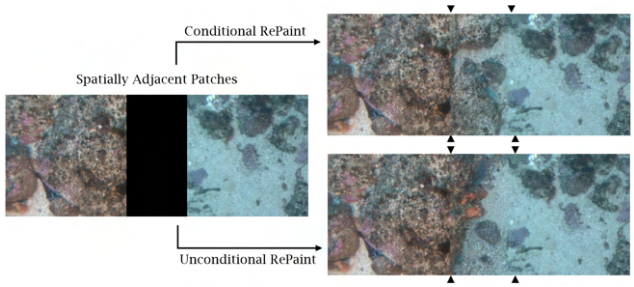


Fig. 11: We find conditional repaint generates heavier boundary effects than unconditional repaint when blending images together.

Given two generated images spatially adjacent to each other, we stitch them together with RePaint [17]. Within the RePaint model, we investigate two approaches: 1) using the same conditional DDPM network used for generation; 2) training a new unconditional DDPM. The result shows that both methods can accomplish inpainting on the generated images. However, the conditional inpainting model creates heavier boundary effects in the image, while unconditional inpainting creates fewer artifacts, as shown in Figure 11. Our hypothesis on this observation is that, for the conditioned inpaint approach, the neural network inpaints the image conditioned on both the existing part of the image as well as the latent embedding. Although they are sampled conditioned on the same latent embeddings, the actual appearance of the existing part may be shifted, creating inconsistencies when inpainting. The unconditional approach depends on the existing part of the image, so fewer artifacts are exhibited at the boundaries between images. The final results we present integrate an unconditional model to blend the images together, alongside the conditional image generation model.

### H. Ablation: Spatial Inpainting Patterns

We further compare our inpainting pattern with most intuitive and commonly used patterns, i.e. raster scan pattern [27] and lawn mowing pattern [28]. The raster scan pattern updates the image space row by row in one direction.

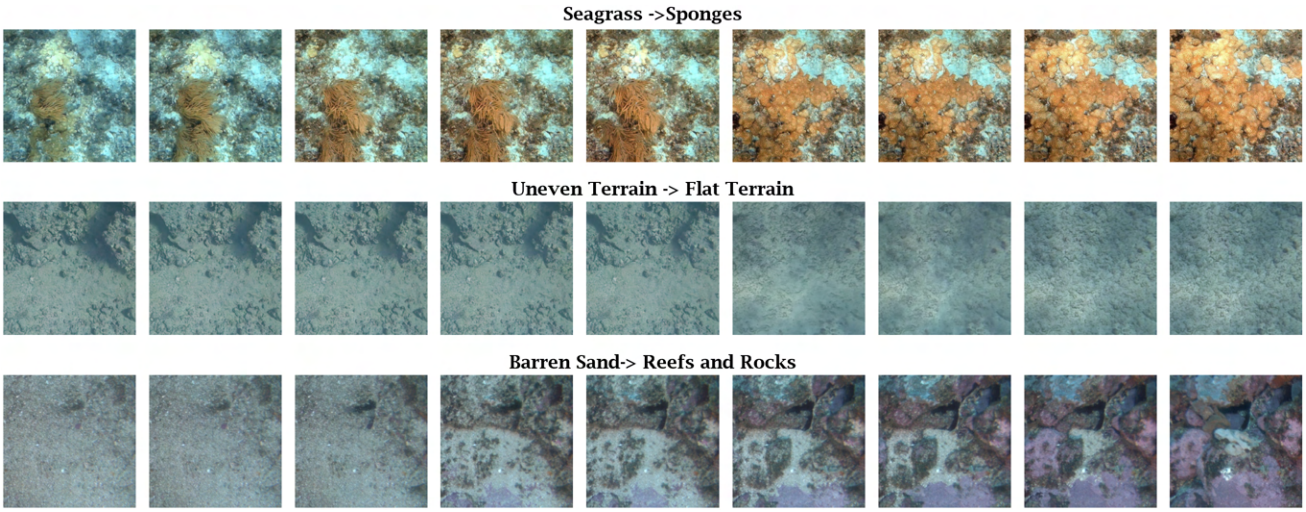


Fig. 12: Examples of image generation conditioned on interpolated DINOv2 embeddings. A smooth transition can be observed.

The lawn mowing pattern updates the image space row by row but in alternating direction, which is commonly used in robot mapping [28]. In comparison, the inpaint method introduced in this paper is parallizable since the new patches are less dependent on previous generated patches. Furthermore, we demonstrate that such dependency reduces latent control accuracy by evaluating the CLIP and DINO latent of generated image patches (Reference embedding of DINO is given; For CLIP embedding we generate a batch of reference image and extract the CLIP embedding as reference). As shown in Table II, which tabulates mean-squared error (MSE) between reference latent and predicted latent.

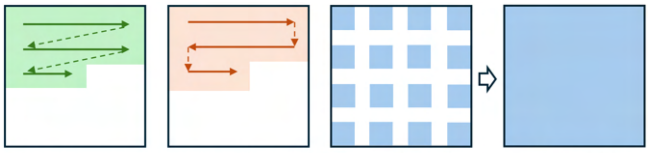


Fig. 13: Our inpainting pattern is parallelizable, comparing to common patterns in image generation and robot mapping, i.e. raster pattern [27] and lawn mowing pattern [28].

Image patches are generated conditioned on input latent. We observe that by leveraging fractal embeddings, DreamSea consistently outperforms baselines that utilize raster scan and lawn mowing patterns which are sequential. These sequential in-painting patterns implicitly assume that the generated terrain contains auto-regressive dependencies while our fractal embeddings explicitly accounts for spatial dependencies along both $x$ and $y$-axes.

TABLE II: MSE↓ on CLIP [21]/DINO [22] embedding space evaluated on individual dataset Florida (FL), Hawaii (HI), Batemans (BM) and Scott Reef (SR). DreamSea outperforms as it does not generate images in a sequentially conditioned order.

| | FL | HI | BM | SR | Ave. |
|---|---|---|---|---|---|
| Raster Order [27] | 0.055/**3.44** | 0.049/3.63 | 0.039/3.66 | 0.055/5.34 | 0.049/4.02 |
| Lawn Mowing [28] | 0.054/3.65 | 0.053/3.34 | 0.043/4.77 | 0.066/5.28 | 0.061/4.24 |
| **DreamSea** | **0.035**/3.46 | **0.029/2.12** | **0.030/2.95** | **0.041/4.48** | **0.034/3.34** |

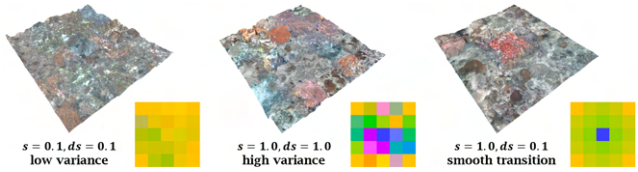


Fig. 14: Effects of scaling factor $s$ and dampening factor $ds$ in fractal process: higher $s$ yields higher variance in generated appearance, and lower $ds$ will smooth the generate scene.

*I. How does scaling $s$ and damping $ds$ control variation in generated terrain?*

In the fractal process described in III-C, two important hyperparameters are the scaling factor $s$ and the damping factor $ds$. $s$ determines the magnitude of randomness applied to the latents and $ds$ is the factor that dampens $s$ over iterations. Figure 14 shows how different values of $s$ and $ds$ affect the generated 3D terrain. With a small $s$ (Figure 14 left), the latent map is smooth. Conditioned on this smooth latent mat which generates a 3D terrain with low variance in appearance. With large $s$ (Figure 14 mid and right), spatial variance is observed and $ds$ controls transition smoothness.

## V. Conclusion and Limitations

This paper introduce *DreamSea*, a diffusion-based generative model that creates realistic benthic terrain with natural variations. *DreamSea* shows how unannotated robotic data collected from field deployments can be employed in concert with generalist foundation models to train a domain-specific deep generative model.

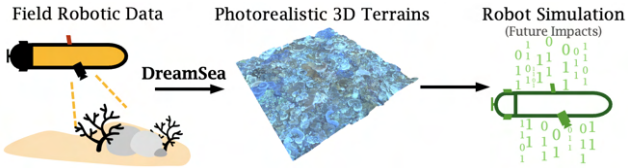


Fig. 15: DreamSea, trained on large-scale unannotated underwater data collected by robots, has potential impacts on field robot simulators in the future.

DreamSea conditions generation upon visual latent embeddings extracted using foundation models. Furthermore, it imbues spatial-awareness into the generative model via a novel latent fractal procedure. The resulting terrain generation allows for the generation of highly diverse underwater environments, while considering spatial-dependencies. The resulting terrain visuals and estimated depths are integrated as priors to construct 3DGS models, which provide both 3D geometry and enables novel-view images to be produced.

However, our proposed method has following limitations: First, it is restricted to generating 3D models solely from a top-down view. This is because the dataset collected by robots is mostly top-down. Second, aligning the model's output with domain-specific expert knowledge, e.g. marine biology, remains an unresolved challenge. Addressing these limitations presents directions for future work.

## ACKNOWLEDGMENT

ChatGPT was used for minor grammar corrections in this paper. This work was partially supported by the National Oceanic and Atmospheric Administration (NOAA) under grant NA22OAR0110624. It was also supported by the Office of Naval Research and NAVSEA under awards: N00178-23-1-0006, N00014-24-1-2301, and N00014-24-1- 2503. The authors thank Corina Barbalata and team at LSU for their exceptional contributions in developing robot platforms.

## REFERENCES

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, "Building rome in a day," *Communications of the ACM*, vol. 54, pp. 105–112, 2011.

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[3] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, July 2023.

[4] G. S. P. Miller, "The definition and rendering of terrain maps," in *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, (New York, NY, USA), p. 39–48, Association for Computing Machinery, 1986.

[5] B. Mandelbrot, *The Fractal Geometry of Nature*. Einaudi paperbacks, Henry Holt and Company, 1983.

[6] A. Fournier, D. Fussell, and L. Carpenter, *Computer rendering of stochastic models*, p. 189–202. New York, NY, USA: Association for Computing Machinery, 1998.

[7] K. Perlin, "An image synthesizer," in *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '85, (New York, NY, USA), p. 287–296, Association for Computing Machinery, 1985.

[8] A. Raistrick, L. Lipson, Z. Ma, L. Mei, M. Wang, Y. Zuo, K. Kayan, H. Wen, B. Han, Y. Wang, A. Newell, H. Law, A. Goyal, K. Yang, and J. Deng, "Infinite photorealistic worlds using procedural generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12630–12641, 2023.

[9] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, K. Ehsani, J. Salvador, W. Han, E. Kolve, A. Kembhavi, and R. Mottaghi, "Procthor: Large-scale embodied ai using procedural generation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5982–5994, 2022.

[10] K. Greff, F. Belletti, L. Beyer, C. Doersch, Y. Du, D. Duckworth, D. J. Fleet, D. Gnanapragasam, F. Golemo, C. Herrmann, *et al.*, "Kubric: A scalable dataset generator," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3749–3761, 2022.

[11] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *stat*, vol. 1050, p. 1, 2014.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[13] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[14] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

[15] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang, "Structured 3d latents for scalable and versatile 3d generation," *arXiv preprint arXiv:2412.01506*, 2024.

[16] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023.

[17] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11461–11471, June 2022.

[18] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[19] 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," *Commun. ACM*, vol. 65, p. 99–106, Dec. 2021.

[20] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," in *The Eleventh International Conference on Learning Representations*.

[21] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[22] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. HAZIZA, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "DINOv2: Learning robust visual features without supervision," *Transactions on Machine Learning Research*, 2024. Featured Certification.

[23] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[24] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, "Langsplat: 3d language gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20051–20060, 2024.

[25] T. Zhang and M. Johnson-Roberson, "Learning cross-scale visual representations for real-time image geo-localization," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5087–5094, 2022.

[26] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[27] T. Li, Y. Tian, H. Li, M. Deng, and K. He, "Autoregressive image generation without vector quantization," in *Advances in Neural Information Processing Systems* (A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, eds.), vol. 37, pp. 56424–56445, Curran Associates, Inc., 2024.

[28] M. Johnson-Roberson, O. Pizarro, S. B. Williams, and I. Mahon, "Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys," *Journal of Field Robotics*, vol. 27, no. 1, pp. 21–51, 2010.