

# Diffusion model-based generation of sea ice data

Aqsa Sultana<sup>a</sup>, Shaik Nordin Abouzahra<sup>a</sup>, Vijayan K. Asari<sup>\*a</sup>, Theus Aspiras<sup>a</sup>, Ruixu Liu<sup>a</sup>,  
Ivan Sudakow<sup>b</sup>, Lee W. Cooper<sup>c</sup>

<sup>a</sup>University of Dayton, Vision Lab, Dayton, Ohio, USA 45469

<sup>b</sup>The Open University, Milton Keynes, England, MK7 6AA, United Kingdom

<sup>c</sup>University of Maryland, Center for Environmental Science, Maryland, USA 21613

## ABSTRACT

As global warming causes climate change, extreme weather has become more common, posing a significant threat to life on Earth. One of the important indicators of climate change is the formation of melt ponds in the arctic region. Scarcity of large amount of annotated arctic sea ice data is a major challenge in training a deep learning model for the prediction of the dynamics of the melt ponds. In this research work, we use diffusion model, a class of generative models, to generate synthetic arctic sea ice data for further analysis of meltponds. Based on the training data, diffusion models can generate new and realistic data that are not present in the original dataset by focusing on the data distribution from a simple to a more complex distribution. First, simple distribution is transformed into a complex distribution by adding noise, such as a Gaussian distribution and through a series of invertible operations. Once trained, the model can generate new samples by starting from a simple distribution and diffusing it to the complex distribution, capturing the underlying features of the data. During inference, when generating new samples, the conditioning information is provided as input alongside the starting noise vector. This guides the diffusion process to produce samples that adhere to the specified conditions. We used high-resolution aerial photographs of Arctic region obtained during the Healy-Oden Trans Arctic Expedition (HOTRAX) in year 2005 and NASA's Operation IceBridge DMS L1B Geolocated and Orthorectified data acquired in 2016 for the initial training of the generative model. The original image and synthetic image are assessed based on their chromatic similarity. We employed evaluation metric known as Chromatic Similarity Index (CSI) for the assessment purposes.

**Keywords:** Global warming, DDPM, meltponds, diffusion models, generative models, HOTRAX, Sea Operation IceBridge, synthetic images, Chromatic Similarity Index (CSI).

## 1. INTRODUCTION

Modern computer vision revolves around Convolutional Neural Networks (CNNs) and other deep learning architectures, and one such integral method is generative models such as Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), normalizing flows, etc [1]. They have come a long way due to their ability to generate realistic images from scratch or manipulating existing images, thus, making significant strides in recent years. However, due to their limitations, such as training instability, blurriness, mode collapse in latent space, and inability to combine both their specialties, researchers introduced a groundbreaking model: the Denoising Diffusion Probabilistic Model (DDPM). DDPM leverages a powerful generative-based model operating on the principle of denoising the diffusion. This entails adding noise into the image which is mapped accordingly to the Gaussian distribution. By transitioning the image from its initial distribution to a Gaussian distribution, it becomes possible to develop a model which inverts this procedure allowing for the creation of novel images. These novel images are inherently similar to the original images in which the noise was added to, as the process of reconstructing information will fit the image distribution [2]. Therefore, diffusion models represent a burgeoning category of generative models used for generating novel data by passing random noise [3].

Arctic sea ice dynamics play a crucial role in the Earth's climate system, influencing weather patterns, ocean circulation, and global climate trends [4]. Accurate simulation and analysis of arctic sea ice behavior are paramount for understanding its impact on the environment and making informed decisions regarding climate policy and resource management. The absence of thorough annotations for Arctic sea ice data, caused by challenges like logistical issues, high costs, and time constraints, as well as the aggressive nature of the sea and weather, presents a major obstacle in gathering the necessary data for training deep learning models to predict the behavior of melt ponds. DDPM offers a promising approach for generating synthetic sea ice data that closely resemble real-world observations without the need for collection of physical data [5]. By adapting the DDPM to the domain of sea ice data generation, we aim to

overcome the limitations of existing approaches and produce high-quality simulations that accurately represent the complex dynamics of sea ice systems.

In this work, we implement Denoising Diffusion Probabilistic Model (DDPM) for generating synthetic sea ice data through temporal evolution of the sea ice region, thus, allowing for the generation of realistic images that can be used for future applications of segmentation, climate modeling etc. We used high-resolution aerial photographs of Arctic region obtained during the Healy-Oden Trans Arctic Expedition (HOTRAX) in year 2005 and NASA's Operation IceBridge DMS L1B Geolocated and Orthorectified data acquired in 2016.

### 1.1 Healy-Oden Trans Arctic Expedition

The aerial photographs were taken with a Nikon D70 digital camera from a helicopter during the Healy Oden Trans Arctic Expedition (HOTRAX) spanning from 5 August to 30 September 2005. To avoid interference from low clouds, flights were conducted at relatively low altitudes ranging from 150 to 700 meters. A total of 1013 images were captured, with an average resolution of  $3042 \times 2048$  pixels. These images provide detailed insights into ice floes, submerged ice, meltponds, and open water [4,6,7]. Sample images are shown in Fig. 1.

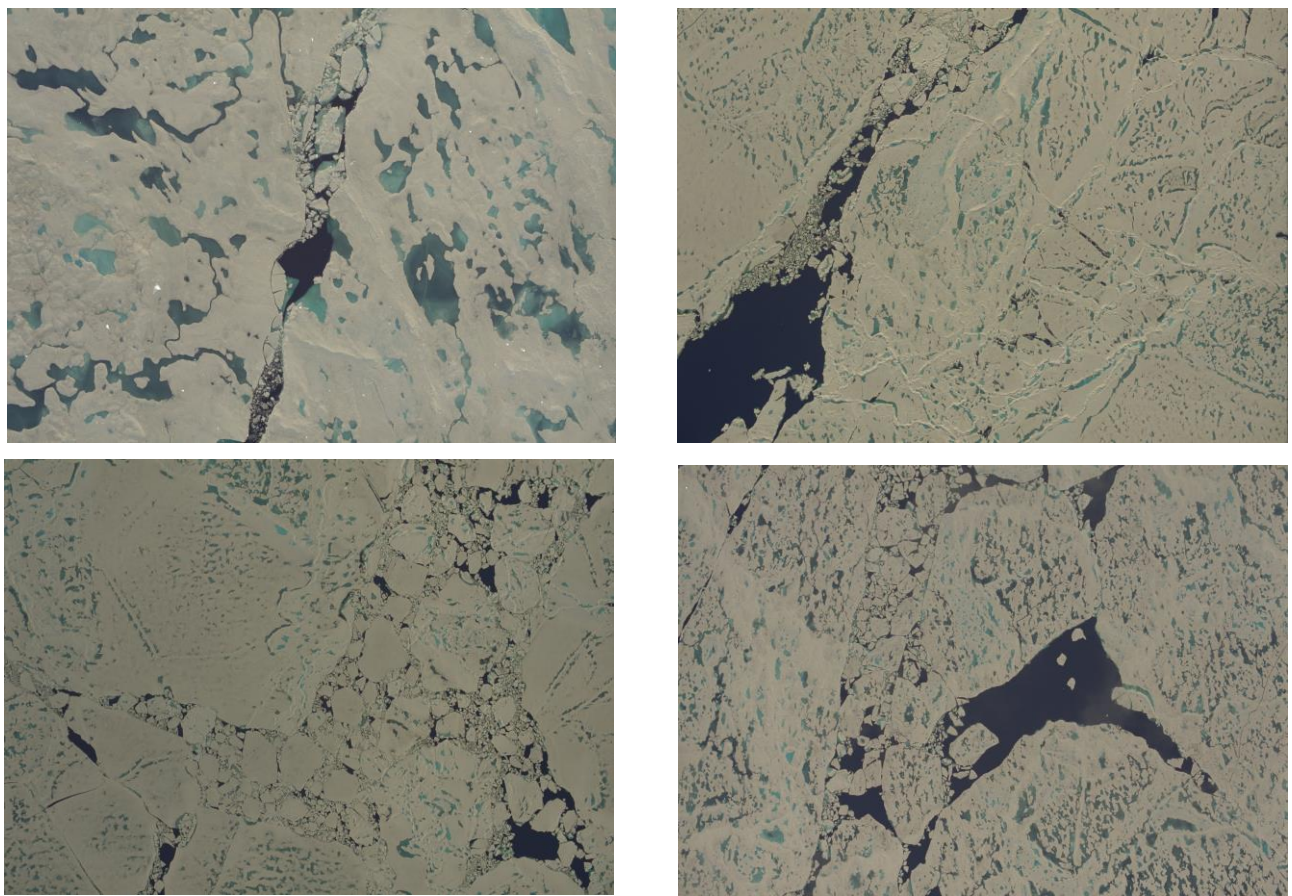


Figure 1. Image samples from HOTRAX database

### 1.2 NASA's Chukchi Sea Operation IceBridge

The IceBridge dataset comprises Level 1B images captured by the Digital Mapping System (DMS) over Greenland and Alaskan waters in 2016. In July, the aircraft flew over the Chukchi Sea, likely during the melting period of sea ice. The imagery collected had a resolution of  $10 \text{ cm} \times 10 \text{ cm}$  and varied in temporal frequency. This operation received funding from aircraft survey campaigns under NASA's Operation IceBridge initiative [8]. The sample images are shown in Fig. 2.



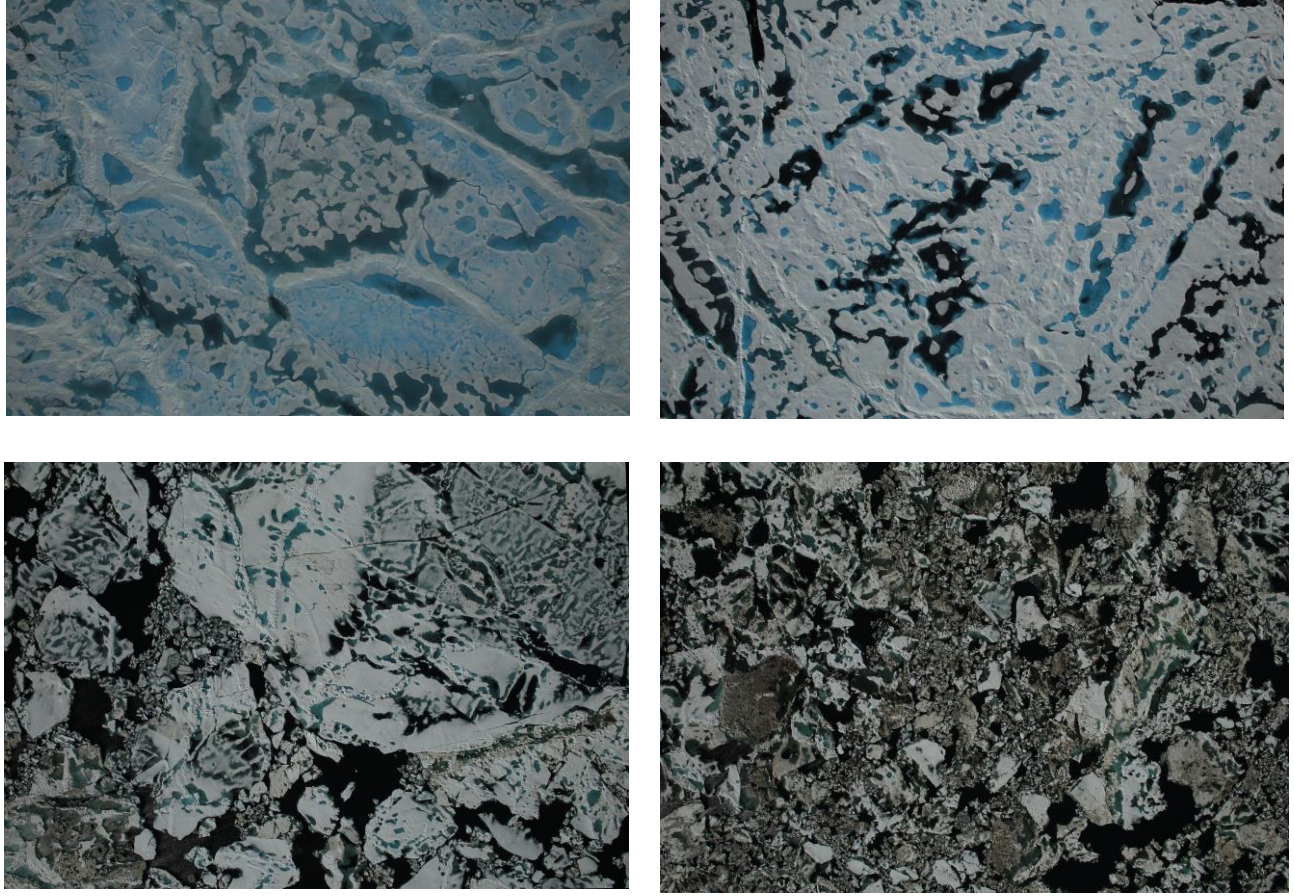


Figure 2. Image samples from Operation IceBridge database

## 2. MODEL ARCHITECTURE

### 2.1 Forward and backward processes

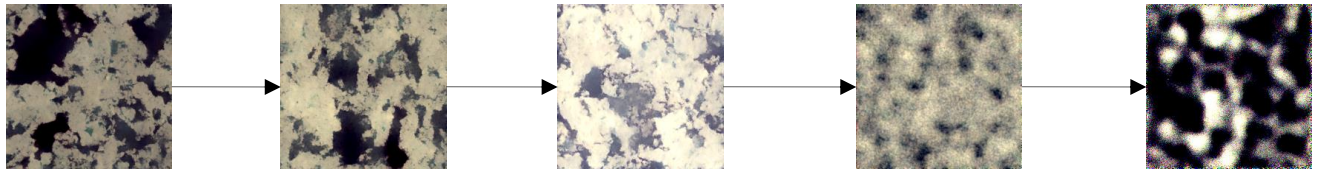


Figure 3. Markov chain manifested for our image data for forward process

DDPM follows two essential processes: forward process and reverse process. The fundamental idea behind the forward process in DDPM is to begin with an original and clear image. Through a defined number of steps represented by 'T,' a minute level of noise is methodically introduced, adhering to a Gaussian distribution as shown in Fig. 3. For the introduction of minute noise at each timestep, we employ Markov chain with an implication that to generate the image at current timestep, the image from the previous timestep is required. The stochastic nature of Markov chain makes the likelihood of transitioning to any specific state depend solely on the current state and the time elapsed, rather than on the sequence of events that occurred previously. This characteristic simplifies the noise addition process. The forward process is also known as fixed or non-learnable step [2,9].

The next goal is to conduct a backward process to denoise the images known as backward diffusion process. The model forecasts the average of the noise introduced between the current and previous timestamps. This approach enables the model to effectively eliminate the noise, attaining the desired result. To sum it up, the forward process allows to perform the prediction of the noise added from the original image to the image at timestamp ‘T’ and traversing backward in Markov’s chain to generate new data from noise [2]. Figure 4 shows an example of a reverse process to generate new data.

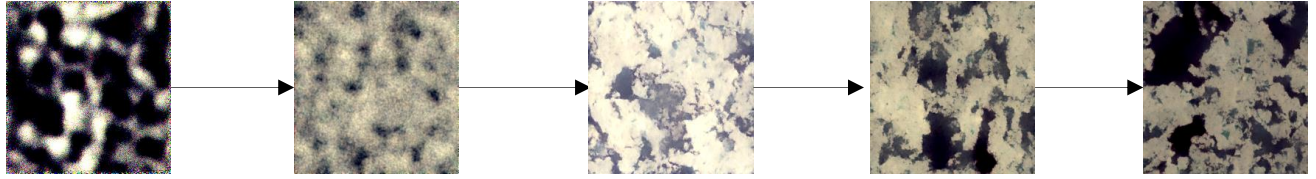


Figure 4. Traversing Markov chain for backward process to generate new image data

## 2.2 Model implementation

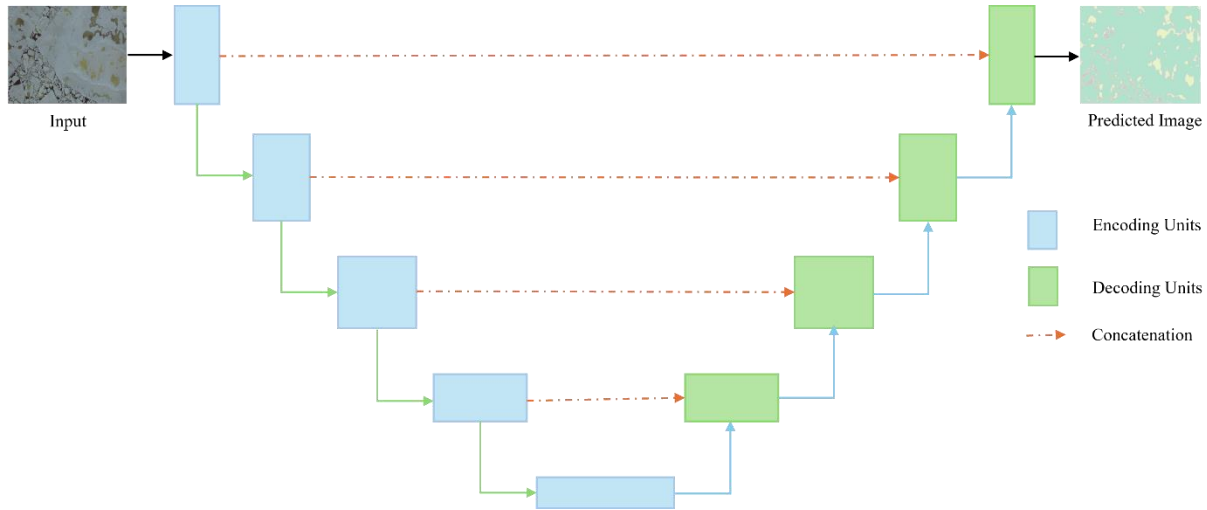


Figure 5. Architectural structure of ‘U’ shaped segmentation network: UNet

We employed a UNet architecture for our diffusion model. Figure 5 shows the overall architecture of UNet. The UNet architecture comprises four encoders and four decoders arranged in the shape of the letter 'U'. Each encoder block consists of 3x3 convolutions followed by Rectified Linear Unit (ReLU) activation functions. The contracting path, which involves stepping through the encoders, doubles the feature channels while halving the spatial dimensions via 2x2 max-pooling operations, effectively down-sampling the data. To maintain information flow between the encoder and decoder, a bridge is formed by a 3x3 convolution followed by ReLU activation functions.

In the expansive path, the decoder employs 2x2 transpose convolution techniques to up-sample the image sizes. Each decoder is concatenated with its corresponding encoder to preserve information from earlier layers. Finally, two convolutions of 3x3 with ReLU activation functions are applied. The output layer of the decoder utilizes 1x1 convolutions with either sigmoid or softmax activation functions, depending on the number of classes. This activation function generates the segmentation mask, representing pixel-level classification [10, 11, 12].

The diffusion process is employed on UNet by modifying its architecture as it requires the same model at every timestep. The procedure involves numerous changes within consistent conditional Gaussian distributions as the image generated are integer pixel values. This requires discrete (log) probabilities for every potential pixel value across all pixels. Thus, instead of using vanilla UNet outlined earlier, we designate the final step in the reverse diffusion

sequence as an independent discrete decoder. At every time step, the UNet model predicts the noise that is required to be subtracted from the input image. Furthermore, the modified UNet architecture also incorporates ConvNext Block [13], attention block [14], MultiLayer Perceptron (MLP) [15] to create a more robust form of UNet. The downsample and upsample blocks used in the model behaves as a standard downsample and upsample process from the UNet. For awareness of the timesteps used, we introduce sinusoidal embedding into the upsampling and downsampling processes [2]. The addition of sinusoidal timestep embedding as one of the building blocks allows for the encoding of a specific time step while preserving relevant, current time information necessary for decoding. This feature ensures the model's adaptability across various time steps it encounters [18]. The engendering of the processes above provides reinforcement and maintains model's capability in retaining same weights throughout the process of denoising fully noised and slightly noised images. The consolidated UNet is shown in Fig. 6

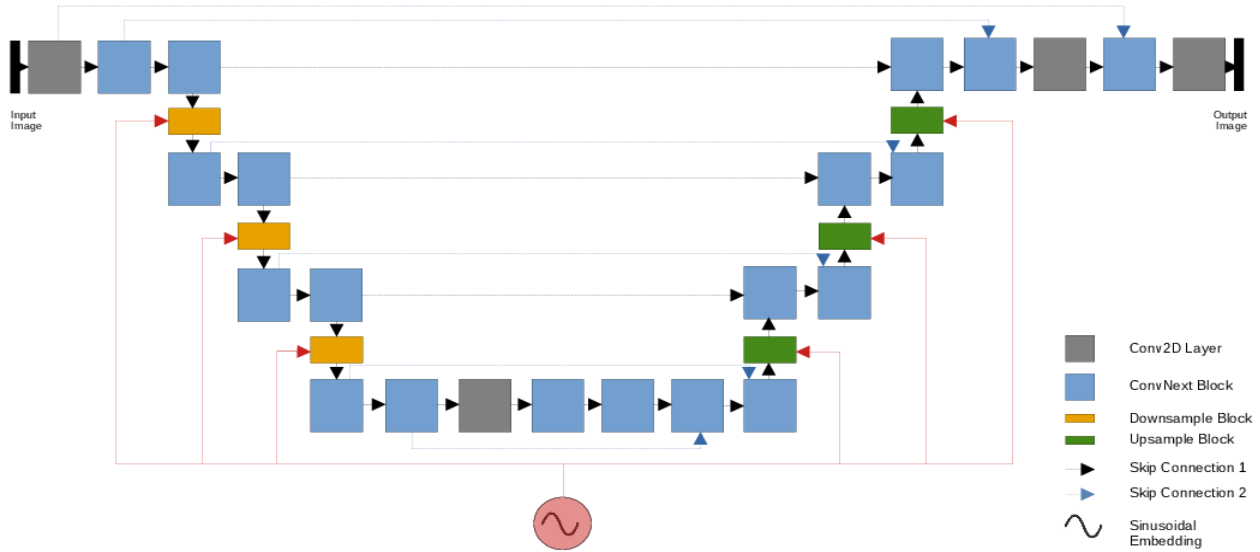


Figure 6. Modified UNet architecture for diffusion model-based

UNet modified diffusion model uses ConvNext infused with attention instead of basic convolutional blocks. To retain the information and prevent loss, we add skip connections. There are two residual connections in each layer instead of one to prevent the overfitting problem due to the complexity of the model (indicated by black and blue dotted lines). To incorporate the time step ( $T$ ) information into the model, we add sinusoidal embedding to all the downsample and upsample blocks (red lines). The final layer of the architecture consists of Conv2d layer with the kernel size of 1 to obtain the predicted noise in image dimension [18].

### 3. EVALUATION METRICS

Color perception is crucial in the process of distinguishing the properties between open water (dark blue rectangle), snow (gray rectangle) and meltponds (light blue rectangle) as shown in Fig. 7. Open water typically appears darker depending on factors such as depth, clarity, and light. The water is darker due to the absorption of longer wavelengths of light i.e., red band and infrared spectrum. Meltponds are pools of water formed atop sea ice due melting. The presence of impurities, angle of observation and depth can vary in meltponds. They appear lighter in color due to the light reflecting from the underlying ice. Their color can vary between light blue to turquoise, sometimes greenish hue due to the algae bloom and sediments. Snow appears white due to its high reflectivity in the visible light. However, snow sometimes appears to have a bluish hue due to impurities or scattering of light. To gauge the resemblances between the original image and the generated image, we suggest an assessment measure built-up on Euclidean distance called the Chromatic Similarity Index (CSI). We created ten patches for each category: snow, meltponds, and open water. Then, we calculated the RMS between the corresponding pixels in the red, blue, and green channels as well as for grayscale variants of these samples. Equation below illustrates the CSI between images  $a$  and  $b$ .



$$CSI_{(a,b)} = \frac{1}{m} \sum_{i=1}^m \sqrt{\frac{1}{n} \sum_{j=1}^n (a_{ji} - b_{ji})^2}$$

Where  $CSI_{(a,b)}$  is the chromatic distance between two images  $a$  and  $b$ .  $n$  and  $m$  are the number of pixels and number of samples respectively, and  $i$  is the individual sample and  $j$  is the index between the original and synthetic pixels.

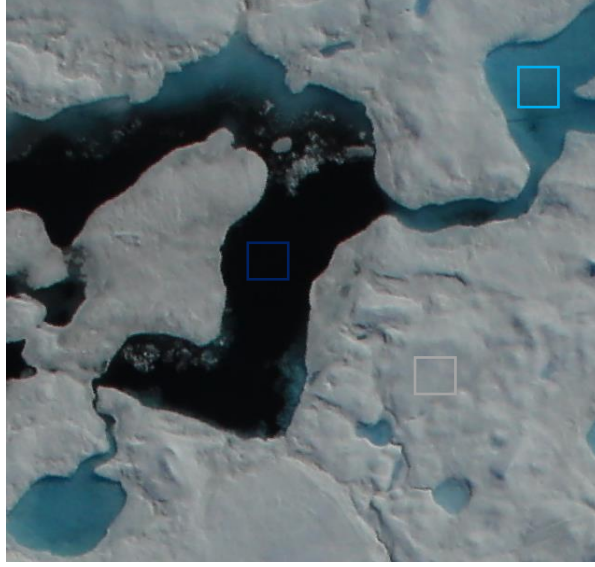


Figure 7. An image of arctic area marked based on its regions

## 4. TRAINING AND EXPERIMENTAL RESULTS

### 4.1 Training method

As mentioned in Sec. 2, diffusion model trains by traversing backward by identifying the reverse Markov transitions that optimize the likelihood of the training dataset. In other words, for each available image in the data, we arbitrarily select a time step ranging between  $[0, T]$ . This will compute the forward process and produce noisy images. And, in backward process, the model is utilized to forecast the added noise in the image. For this purpose, as we are dealing with probabilistic latent space, we incorporate Kullback-Leibler (KL) Divergence loss function, an asymmetric statistical distance measuring the two probability distributions' difference.

Table 1. Configuration for the training, UNet and diffusion model

| Training                |                    | UNet       |           | Diffusion         |        |
|-------------------------|--------------------|------------|-----------|-------------------|--------|
| Batch Size              | 16                 | Input size | 64        | Timesteps         | 100    |
| Learning Rate           | $1 \times 10^{-4}$ | Batch size | 16        | Beta scheduler    | Linear |
| Number of steps         | 50000              | Dimension  | [1,2,4,8] | Output image size | 224    |
| Save samples every      | 1000               | Channels   | 3         |                   |        |
| Number of samples saved | 9                  |            |           |                   |        |

Table 1 displays the configuration of training, UNet and diffusion processes respectively. We evaluated our diffusion model on HOTRAX and Operation IceBridge database. The model was trained for 50,000 steps with a batch size of 16. The image size obtained was 224x224 due to hardware limitations. Using Exponential Moving Average, samples were generated every 1000 steps and model was saved. The experiment was carried out utilizing an NVIDIA GEFORCE RTX Titan GPU with 24 GB of VRAM and 128 GB of RAM. The PyTorch framework in Python was employed for the experiment.

#### 4.2 Experimental results

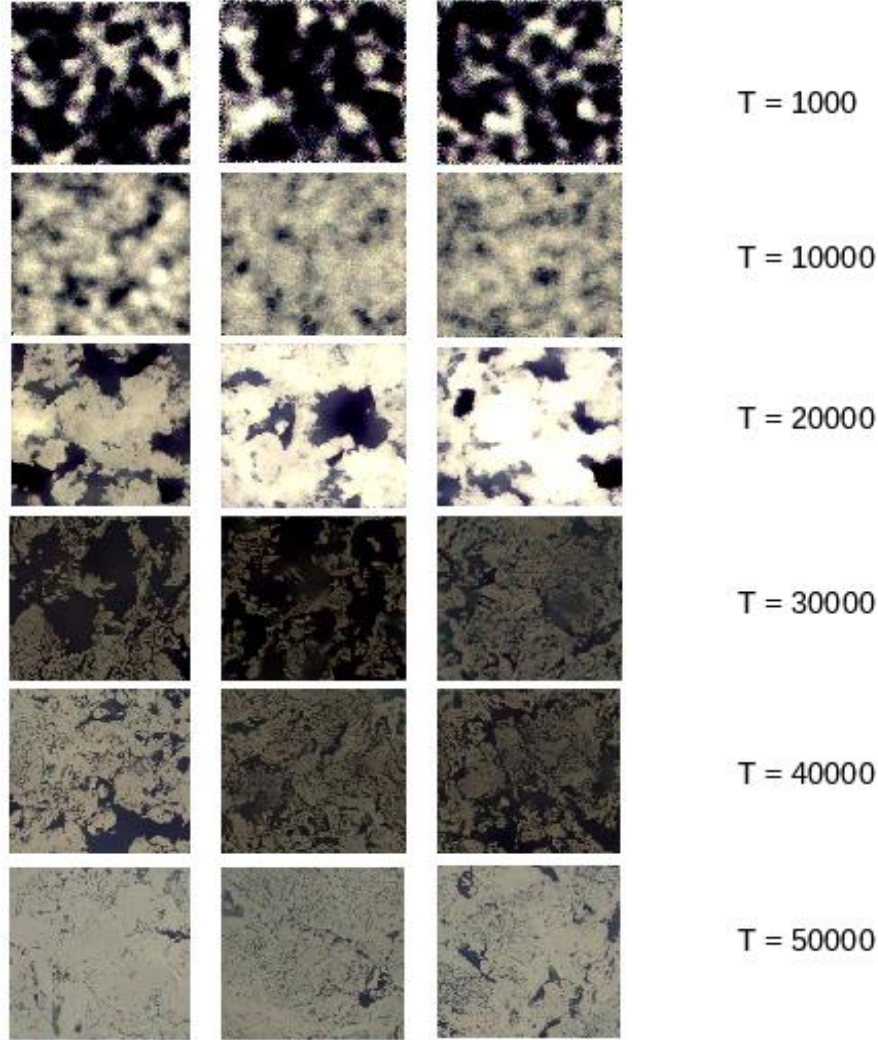


Figure 8. Generated images at different time steps for HOTRAX

In Fig. 8, for HOTRAX data, the top row illustrates the images acquired during the initial 1000 steps, revealing primarily noise with minimal discernible features. Progressing to 10000 and 20000 timesteps, the model begins to gradually capture some features. However, the most substantial advancements in feature development occur around 30000 and 40000 steps. By the conclusion of 50000 steps, we observe promising results produced by the model, with a notable enhancement in sample quality compared to earlier stages. Similarly, generated images for Operation IceBridge based on timesteps are illustrated in Fig. 9. Furthermore, Fig. 10 and Fig. 11 illustrate more samples generated by DDPM for HOTRAX and Operation IceBridge databases respectively.

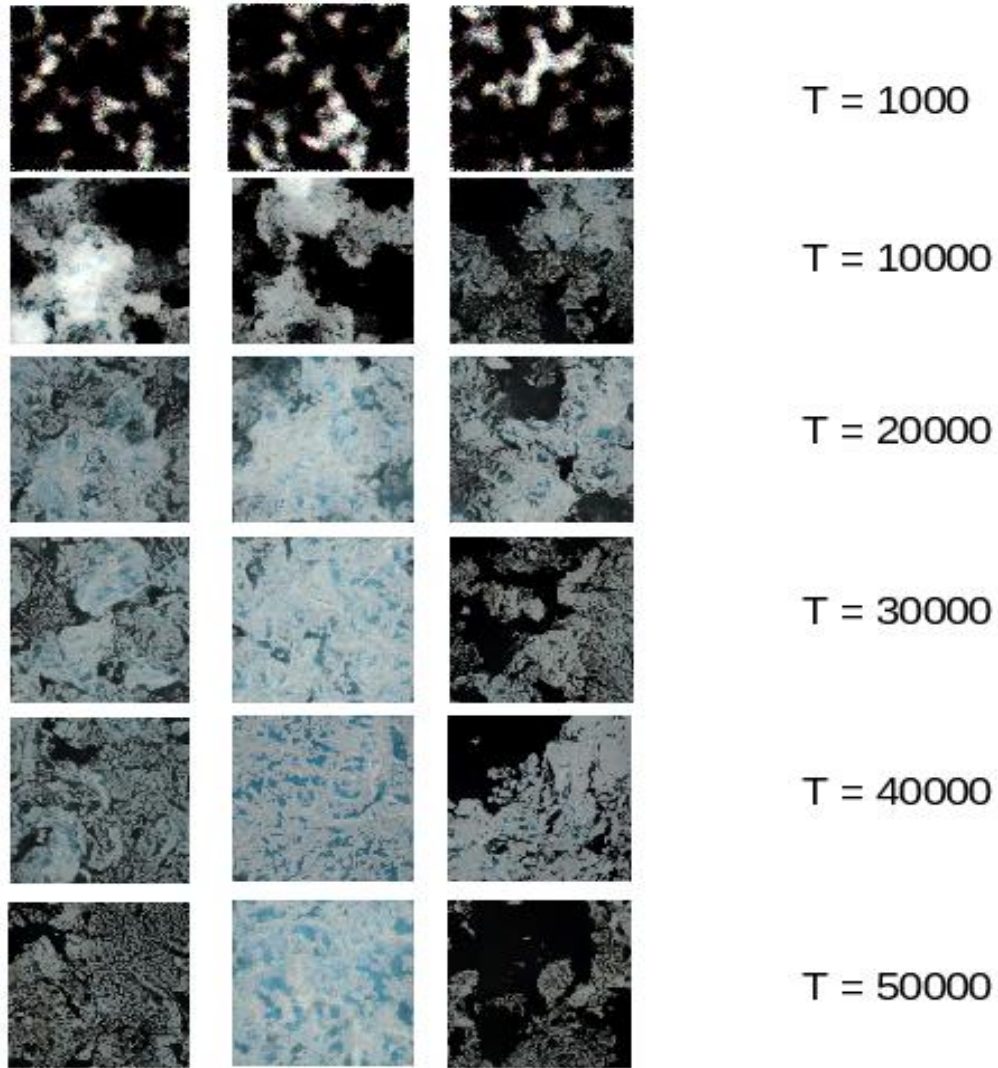


Figure 9. Generated images at different time steps for Operation IceBridge

To assess the chromatic similarity between the original and generated images, we first created patches for three classes: meltponds, snow, and open water. Next, we normalized both sets of class patches and computed CSI for all the samples. The resulting chromatic differences between the corresponding pixels in the original class patches (reference) and generated class patches (testing) are presented in Table 2 and Table 3 for both the databases. Lower the CSI, similar are the original and synthetic images. We observe that the color sample distances between original snow and synthetic snow are the smallest compared to the original and synthetic pairs of snow-open water and snow-meltpond, which exhibited larger disparities between their corresponding pixels. Likewise, the original and synthetic pairs, such as open water - snow and open water - meltponds, showed greater disparities in distances compared to open water - open water pairs. Moreover, concerning meltponds in both the original and synthetic pairs, such as meltponds - snow and meltponds - open water, the differences were more pronounced when contrasted with meltponds - meltponds, exhibiting the lowest differences. Therefore, as the CSI decreases, the original and synthetic images become increasingly alike.



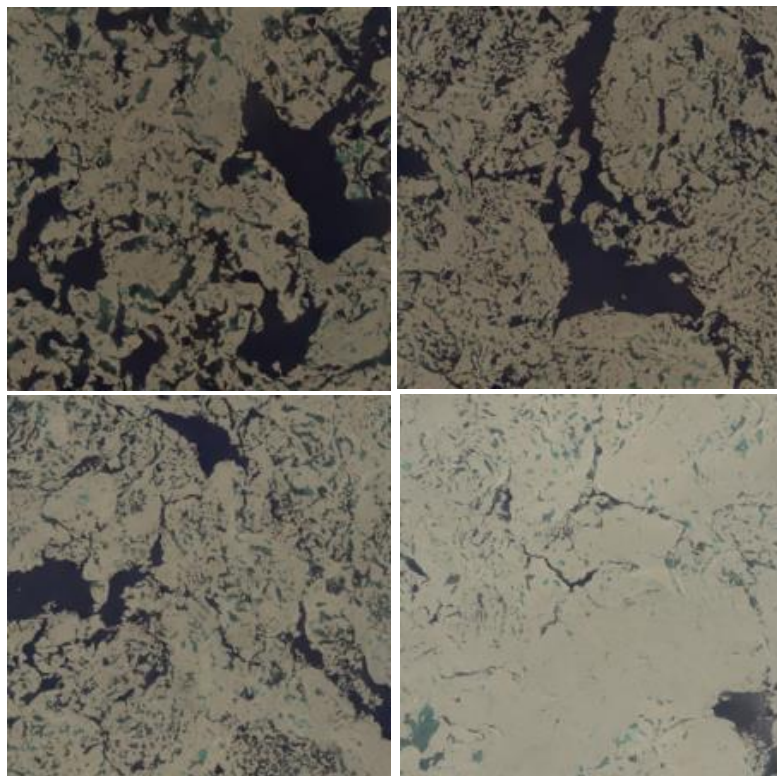


Figure 10. More diffusion model-based generated samples at T=50000 for HOTRAX

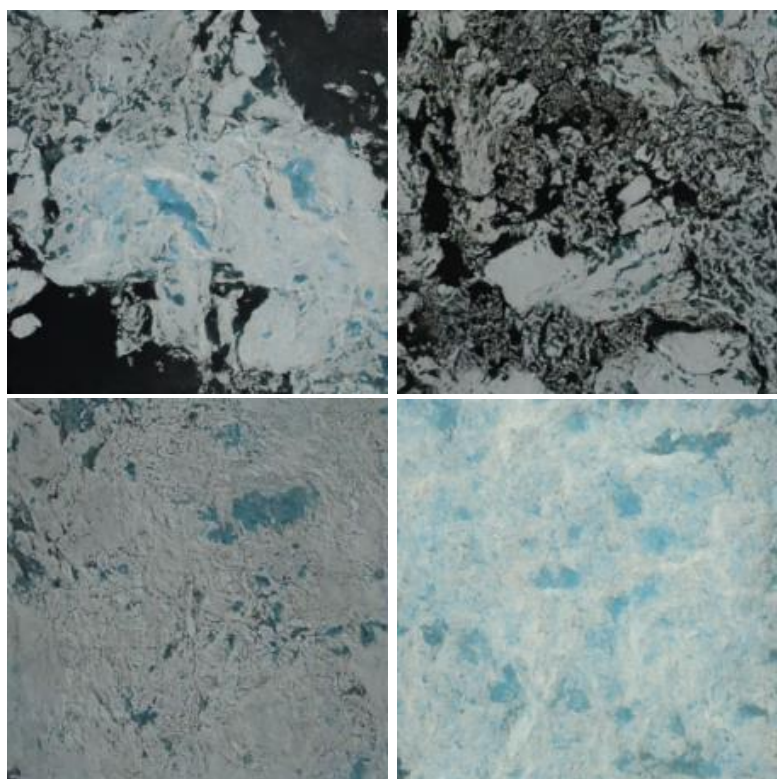


Figure 11. More diffusion model-based generated samples at T=50000 for Operation IceBridge

Table 2. CSI for HOTRAX database

| Reference | Snow           |            |           | Open water |                |           | Meltpond |            |                |
|-----------|----------------|------------|-----------|------------|----------------|-----------|----------|------------|----------------|
| Testing   | Snow           | Open water | Melt pond | Snow       | Open water     | Melt pond | Snow     | Open water | Melt pond      |
| $\bar{R}$ | <b>0.04191</b> | 0.11889    | 0.06328   | 0.13857    | <b>0.02390</b> | 0.05427   | 0.09256  | 0.06739    | <b>0.03286</b> |
| $\bar{G}$ | <b>0.04152</b> | 0.11881    | 0.06331   | 0.13838    | <b>0.02395</b> | 0.05407   | 0.09227  | 0.06748    | <b>0.03276</b> |
| $\bar{B}$ | <b>0.04125</b> | 0.11927    | 0.06355   | 0.13873    | <b>0.02386</b> | 0.05419   | 0.09239  | 0.06748    | <b>0.03276</b> |
| Grayscale | <b>0.15022</b> | 0.47681    | 0.20935   | 0.53535    | <b>0.09003</b> | 0.18714   | 0.34621  | 0.27691    | <b>0.09697</b> |

Table 3. CSI for Operation IceBridge database

| Reference | Snow           |            |           | Open water |                |           | Meltpond |            |                |
|-----------|----------------|------------|-----------|------------|----------------|-----------|----------|------------|----------------|
| Testing   | Snow           | Open water | Melt pond | Snow       | Open water     | Melt pond | Snow     | Open water | Melt pond      |
| $\bar{R}$ | <b>0.00819</b> | 0.07212    | 0.04624   | 0.07596    | <b>0.00760</b> | 0.03620   | 0.04487  | 0.02946    | <b>0.00902</b> |
| $\bar{G}$ | <b>0.00834</b> | 0.07210    | 0.04616   | 0.07586    | <b>0.00769</b> | 0.03633   | 0.04484  | 0.02927    | <b>0.00889</b> |
| $\bar{B}$ | <b>0.00828</b> | 0.07218    | 0.04608   | 0.07586    | <b>0.00763</b> | 0.03638   | 0.04497  | 0.02918    | <b>0.00868</b> |
| Grayscale | <b>0.02513</b> | 0.23820    | 0.14701   | 0.24748    | <b>0.02167</b> | 0.11082   | 0.14270  | 0.08644    | <b>0.02509</b> |

## 5. CONCLUSION

In this work, we utilize Denoising Diffusion Probabilistic Model (DDPM), a category of generative models, to create synthetic data representing Arctic sea ice. Diffusion model follows the Markov chain process consisting of forward diffusion and reverse diffusion. These models are adept at generating novel and realistic data beyond what is included in the original dataset by transitioning from a basic to a more intricate data distribution through the addition of Gaussian noise for the forward process. A UNet based architecture was used in the reverse diffusion process to predict the subtractive noise to generate noise-free synthetic images

We introduced an evaluation metric, the Chromatic Similarity Index (CSI), to gauge the chromatic similarities between the original images and the generated images. This metric entailed computing the normalized RMS for the RGB channels, along with their grayscale counterparts for the ten patches. We then assessed the chromatic resemblances based on the results after calculating the mean differences between the corresponding pixels. In Sec.4, Tables 2 and 3 revealed that the chromatic difference between the corresponding pixels in the original and generated images of snow and snow, open water and open water, meltpond and meltpond were minimal suggesting that generated images are closer to the original images. Therefore, we can assert that the generated images exhibit high quality, promising characteristics, and realism, making them suitable for future analysis.

In the future, our aim is to apply our pretrained model to segment the generated images. Additionally, we intend to integrate positional embedding into the Diffusion Model to generate sequential patches. This approach will enable us to achieve high-resolution images by seamlessly stitching the patches together. We would also like to create high precision masks using in-painting which features the use of tokens.

## ACKNOWLEDGEMENTS

This work was supported by the Division of Physics at the National Science Foundation (NSF), Grant No. PHY 2102906.

## REFERENCES

- [1] Man K, Chahl J. A Review of Synthetic Image Data and Its Use in Computer Vision. *J Imaging*. 2022 Nov 21;8(11):310. doi: 10.3390/jimaging8110310. PMID: 36422059; PMCID: PMC9698631.
- [2] Ho, J., Jain, A., & Abbeel, P. (2020, June 19). *Denoising diffusion probabilistic models*. arXiv.org. <https://arxiv.org/abs/2006.11239>
- [3] Chang, Z., Koulieris, G. A., & Shum, H. P. H. (2023, June 7). *On the Design Fundamentals of Diffusion Models: A Survey*. arXiv.org. <https://arxiv.org/abs/2306.04542>
- [4] Sultana, A., Asari, V. K., Sudakow, I., Aspiras, T., Liu, R., & Demchev, D. (2023, June). R2UNet for melt pond detection. In *Pattern Recognition and Tracking XXXIV* (Vol. 12527, pp. 206-218). SPIE.
- [5] Li, W., Hsu, C.-Y., and Tedesco, M.: Advancing Arctic sea ice remote sensing with AI and deep learning: now and future, *EGUsphere* [preprint], <https://doi.org/10.5194/egusphere-2023-2831>, 2024.
- [6] I. Sudakow, V. K. Asari, R. Liu and D. Demchev, "MeltPondNet: A Swin Transformer U-Net for Detection of Melt Ponds on Arctic Sea Ice," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 8776-8784, 2022, doi: 10.1109/JSTARS.2022.3213192.
- [7] Ivan Sudakow, Vijayan Asari, Ruixu Liu, & Denis Demchev. (2022). Melt pond from aerial photographs of the Healy–Oden Trans Arctic Expedition (HOTRAX) (1.0) [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.6602409>
- [8] Dominguez, R. (2010). IceBridge DMS L1B Geolocated and Orthorectified Images, Version 1 [Data Set]. Boulder, Colorado USA. NASA National Snow and Ice Data Center Distributed Active Archive Center. <https://doi.org/10.5067/OZ6VNOPMPRJ0>. Date Accessed 03-15-2024.
- [9] Tim Salimans, Diederik Kingma, and Max Welling. Markov Chain Monte Carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pages 1218–1226, 2015
- [10] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015
- [11] Aqsa Sultana, Vijayan K. Asari and Theus Aspiras "Residues in succession recurrent U-Net for segmentation of retinal blood vessels", *Proc. SPIE 12527, Pattern Recognition and Tracking XXXIV*, 1252706 (13 June 2023); <https://doi.org/10.1117/12.2664876>
- [12] Sultana, Aqsa. *Residues in Succession U-Net for Fast and Efficient Segmentation*. 2022. University of Dayton, Master's thesis. *OhioLINK Electronic Theses and Dissertations Center*, [http://rave.ohiolink.edu/etdc/view?acc\\_num=dayton1659016279233472](http://rave.ohiolink.edu/etdc/view?acc_num=dayton1659016279233472).
- [13] Liu, Z., Mao, H., Wu, C., Feichtenhofer, C., Darrell, T., & Xie, S. (2022, January 10). *A ConvNet for the 2020s*. arXiv.org. <https://arxiv.org/abs/2201.03545>
- [14] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., Glocker, B., & Rueckert, D. (2018, April 11). *Attention U-Net: Learning where to look for the pancreas*. arXiv.org. <https://arxiv.org/abs/1804.03999>
- [15] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020, October 22). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. arXiv.org. <https://arxiv.org/abs/2010.11929>
- [16] DC. He and L. Wang (1990), "Texture Unit, Texture Spectrum, And Texture Analysis", *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 28, pp. 509 – 512.
- [17] [https://gis.cdc.gov/grasp/diabetes/data/tutorials/analysis/equations\\_eucdist.html](https://gis.cdc.gov/grasp/diabetes/data/tutorials/analysis/equations_eucdist.html).
- [18] Kemal Erdem, (Nov 2023). "Step by Step visual introduction to Diffusion Models.". <https://erdem.pl/2023/11/step-by-step-visual-introduction-to-diffusion-models>