Physic-Informed Auto-Encoder for Ocean Data Compression

Zifu Kou 36920231153203
Shuang Yang 36920231153254
Chengzhi Xiong 36920231153248
Pinhou Yang 36920231153253
XMU University. AI Class.

Abstract

In recent years, with the increase of ocean data, its data scale can reach PB level. How to compress and transmit data quickly and with high quality has attracted more and more attention. Existing lossy or lossless compression algorithms can have good effects, but algorithms that combine with physical mechanisms in the ocean are rare. At the same time, with the rapid development of deep learning, its applications in various fields have significant effects. This topic aims to explore the data compression algorithm that combines physical mechanisms and deep learning methods, and compress and decompress ocean data quickly and with high quality.

1 Introduction

Neural image compression algorithms have made great progress in recent years. Compared with traditional lossy compression algorithms such as JPEG or JPEG2000, deep learning-based algorithms not only have a great improvement in compression rate, but also have strong performance in distortion rate(He et al. 2022a; Jamil et al. 2023). However, most of the algorithms are still based on data-driven mode, learning the spatial features of images, and quantizing and entropy coding the feature maps to achieve the purpose of compressing data. At the same time, with the increasing amount of ocean surface data in recent years, its high-resolution ocean current data, sea surface temperature data, salinity data can reach PB order of magnitude. How to effectively compress ocean data has also become a concern(Wang, Zhou, and Zhou 2023).

Different from the data-driven image compression mode, 1) ocean data has related physical constraints prior knowledge; 2) ocean data has four dimensions including longitude, latitude, depth and time, and it has been shown that adding physical knowledge as prior knowledge into the model can make model more stable and accuracy(Erichson, Muehlebach, and Mahoney 2019). How to add prior knowledge from Ocean science to the model to make the it more stable and reliable has also become a popular direction for the application of AI in the field of natural sciences(Campin et al. 2011).

The problem we are studying is whether we can use the physical constraints in ocean research, and use deep learning methods to propose an effective compression algorithm for ocean data, such as shown in Figure 1, which can compress ocean data more efficiently and with high quality, quickly transfer, and improve the efficiency and quality of ocean researchers to obtain and transmit data.

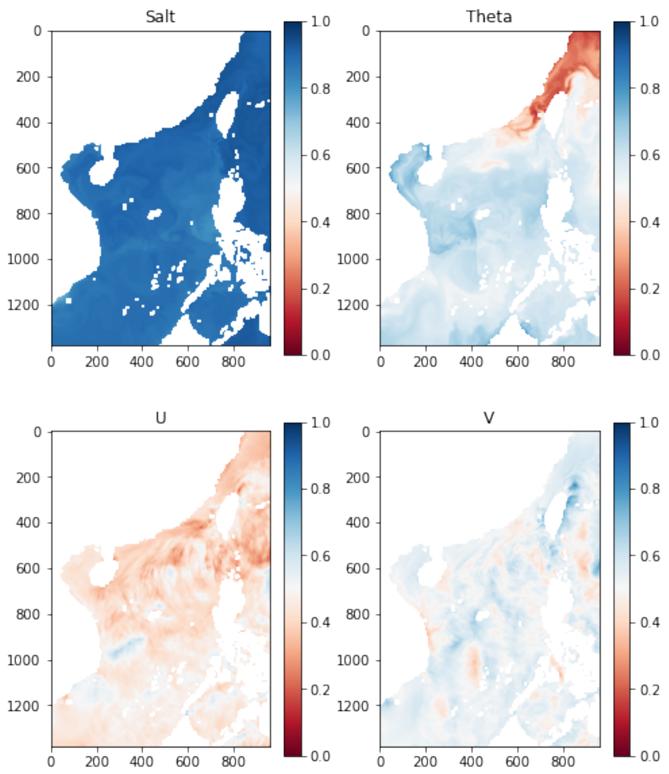

Figure 1: Ocean Data, The South Sea of China

2 Related Work

Image compression algorithm. A compression algorithm based on spatial channels has been proposed(He et al. 2022a), and it has good spatial channel information capture ability and good compression quality and speed. The authors proposed a compression method based on model parameters(Dupont et al. 2022), using the fitted model parameters and the trained offset vector to represent the image information. (Ballé, Laparra, and Simoncelli 2016) proposed to use AutoEncoder for image compression, and proposed the GDN normalization method and the model-based compression data quantization method, which have a significant improvement in data effect compared with JPEG and JPEG2000. (Zhang, Xu, and Yang 2022) proposes an image compression model based on Conditional Generative Adversarial Networks (HPIC) and an overprior probability model is first used to encode and quantize the original image in this model.(Zhu, Li, and Zhang 2022) introduces an innovative multi-source data preprocessing method, which uniformly converts the DN value of multispectral images into reflectivity and get a notable improvement in the quality of reconstructed images. The multispectral image compression framework utilizes 1 * 1 convolution to reduce inter-spectral redundancy, self-encoder to reduce image dimension, Gaussian mixture entropy coding to estimate the code rate, and rate-distortion optimization to jointly optimize the code rate and distortion.

Masked autoencoders introduce a masking techniques, encouraging the model to focus on key image features, efficiently capturing image structural information and improving compression-decompression quality (He et al. 2022b). Furthermore, masked autoencoders exhibit scalability, suitable for large-scale image datasets, and offer flexibility in the masking process, allowing adjustments for different compression settings, enhancing the method's versatility and it has been shown that masked autoencoder has great advancements in the field of image decompression. Receptance Weighted Key Value (RWKV) is a novel model introduced in 2023(Peng et al. 2023), aimed at addressing the challenges associated with handling long sequences. This model combines the efficient training methods of Transformers with the efficient inference process of RNNs, leveraging a linear attention mechanism for model construction. Empirical results have demonstrated that RWKV can perform on par with similarly sized Transformers, while exhibiting linear memory and computational requirements(Bo Peng, and Eric Alcaide 2023).

Compression algorithm with natural science. As in (Wang, Zhou, and Zhou 2023), a matrix compression algorithm based on SVD was used, and experiments were carried out based on the data of Yangtze River Delta in China and HYCOM high-resolution ocean data, and achieved good results, but its calculation method was more complex. The authors proposed to use video compression method(Berres et al. 2017) to compress ocean database data, based on temporal dimension t, and apply video compression technology to compress temporal group images. (Tarasiou, Chavez, and Zafeiriou 2023) used the ViT model based on Transformer structure to process high-resolution temporal satellite remote sensing images.

Algorithm combining physical knowledge. (Thuerey et al. 2021) provided a deep learning method based on physical framework and its application. (Campin et al. 2011) also provided a method of adding physical constraints into the model for training, transforming physical constraints into part of the loss function and adding them into the model training, so that the model can learn the knowledge of physical constraints. (Erichson, Muehlebach, and Mahoney 2019) Based on AutoEncoder framework and added Lyapunov condition to enhance the robustness of the model in prediction, and used Gulf of Mexico sea surface data to verify the model effect. The method of combining physical prior knowledge and deep learning image compression algorithm has not been searched for temporarily, and plans to further conduct literature research in the future. (Fang 2022) provides us with an approach in which a hybrid physics-informed neural network for partial differential equations (PDEs) is proposed, drawing inspiration from convolutional neural networks and finite volume methods. It employs an approximation of the differential operator to solve PDEs, which has been proven to have a convergent rate. This method can be adapted for solving PDEs in ocean data, improving the accuracy of compression algorithms.

(Wang et al. 2023) model the learning and decision-making process of CNN with a statistical physical percolation model. Based on the differentiation degree and vulnerability of percolation. Propose the concept of CNN differentiation degree and summarize the empirical formula to quantify it. The relationship between the differentiation degree and vulnerability is analyzed from both adversarial attack and adversarial training perspectives to explain the decision-making mechanism of CNN and classification reliability.(de Haan et al. 2020)By adding a variety of physical constraints in the optimization process, better image reconstruction results can be achieved. Image Reconstruction and Enhancement Based on Deep Learning This paper provides an overview of efforts in the field of computational microscopy and optical sensing systems using deep neural network propulsion microscopy. Deep learning has proven to be one of the leading machine learning techniques for a variety of inference tasks.

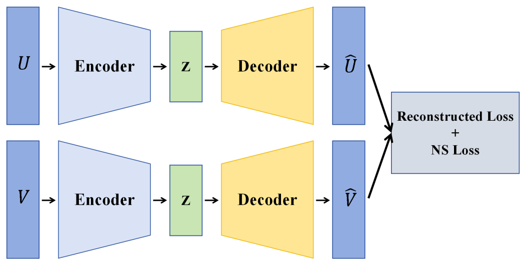

Figure 2: Model Structure

3 Proposed Solution

In this section, we will introduce our solution in three parts. First we will show our basic temporal model structure for image compression as Figure 2. Then we will introduce generalized divisive normalization(GDN). Finally we will introduce the NS equation and how do we imply it into our loss function.

Temporal Model

Most ocean data compression methods are focusing on singular snapshot(Jamil et al. 2023; Momenifar et al. 2022; Glaws, King, and Sprague 2020). Although these methods outperform than the original methods such as SVD and JPEG, they do not take fully advantage of the information in time dimension. In order to get more features in time dimension, we use 3D convolutional network to extract the time information. Inspired by the breakthroughs in deep learning in the field of imaging, rapid progress has been made in feature learning in recent years. Various pre-trained convolutional network (ConvNets) models are available for extracting image features. To delve further into research, we have adopted deep 3D ConvNet(Tran et al. 2015), which excels in processing video data, to learn spatio-temporal features.

However, one limitation of C3D is that, although it is based on 3D convolution, its feature extraction capability is severely constrained by computation and parameter quantity. The C3D model has only 11 layers but a model size of 321M. In contrast, the deeper resnet-152 network is only 235M in size. This limitation affects not only the feature extraction capability of C3D but also the training scope of 3D networks. Therefore, we adopted the P3D (Pseudo-3D Residual Networks) convolutional network to enhance the performance of model training. P3D draws from the concept of asymmetric convolution proposed in InceptionNets, decomposing a k×k×k 3D convolution into a 1×k×k convolution for extracting spatial features and a k×1×1 convolution for merging temporal features, and integrating these two decomposed convolutions in a p3d block. P3D not only significantly reduces the size of the model but also fully utilizes the scene and object knowledge learned from images, making the P3D ResNet, based on 2D spatial convolution plus 1D temporal convolution, significantly outperform the C3D using direct 3D spatio-temporal convolution.

In our research project on ocean data compression, we combined the unique advantages of C3D and P3D models to effectively improves the processing efficiency of time and spatial features, and our model structure is shown in Figure3. We use the Autoencoder structure, and the compressed datas are the latent vectors processed by the encoder, and we use a decoder to reconstruct data. In the details of Encoder and Decoder, we use a temporal 3D convolution to downsample the time dimension, and use a spatial 3D convolution to downsample the spatial dimension. Then we use a GDN module to normalize feature map. As for Decoder, we use transpose convolution to upsample and inverse GDN(IGDN) to renormalization.

GDN module

Follow (Ballé, Laparra, and Simoncelli 2016), we imply the GDN module to better reconstruct image. The GDN module is a widely used normalization method in convolutional neural networks, which introduces gating mechanisms to regulate feature propagation and activation. The GDN module consists of two steps: channel-wise gating and feature map gating. Channel-wise gating generates a control signal with the same number of channels as the input through a 1x1x1 convolutional layer, which is used to adjust the feature response of each channel. Feature map gating multiplies the channel-wise gating signal element-wise with the feature map to control feature propagation. This gating mechanism can adjust the spatial and channel correlations of features in the network, thereby improving feature representation. The GDN module possesses adaptive and highly non-linear spatial characteristics, which distinguishes it from batch normalization. Batch normalization normalizes across the channel dimension, so different samples would obtain the same normalization result for the same channel. On the other hand, the GDN module is adaptive in the spatial domain, so even within the same channel, different positions may receive different normalization results. Additionally, the GDN module exhibits highly non-linear characteristics, allowing it to better adapt to different tasks and data.

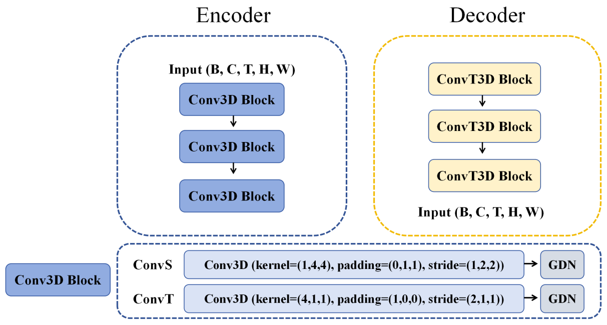

Figure 3: Encoder and Decoder Structure

Our analysis transformation process, denoted as g_a , consists of three stages: convolution, sub-sampling, and divisive normalization. Specifically, the transformation is expressed as $u_i^{(k)}(t,m,n)$ for the ith input channel of level k at the spatial position (t,m,n). The input image vector x corresponds to $u_i^{(0)}(t,m,n)$, and the output vector y corresponds to $u_i^{(3)}(t,m,n)$. Each stage begins with an affine convolution:

$$v_i^{(k)}(t,m,n) = \sum_j (h_{k,ij} * u_i^{(k)})(t,m,n) + c_{k,i} \qquad (1)$$

Where $*$ represents 3D convolution, followed by downsampling:

$$w_i^{(k)}(t,m,n) = v_i^{(k)}(s_k t, s_k m, s_k n) \qquad (2)$$

Where s_k is the downsampling factor for stage k. Each stage concludes with a GDN operation:

$$u_i^{(k+1)}(t,m,n) = \frac{w_i^{(k)}(t,m,n)}{\sqrt{\beta_{k,i} + \sum_j \gamma_{k,ij}(w_i^{(k)}(t,m,n))^2}} \qquad (3)$$

The full set of h, c, β, and γ parameters (across all three stages) constitute the parameter vector ϕ to be optimized.

Similarly, the synthetic transformation g_s also consists of three stages, with the order of operations in each stage reversed, the downsampling replaced by the upsampling, and the GDN replaced by an approximate inverse function we call IGDN. We define $\hat{u}_i^{(k)}(t,m,n)$ as the input to the kth synthesis stage, so that the $\hat{y}$ corresponds to $\hat{u}_i^{(0)}(t,m,n)$, and $\hat{x}$ corresponds to $\hat{u}_i^{(3)}(t,m,n)$. Each stage consists of an IGDN operation:

$$\hat{w}_i^{(k)}(t,m,n) = \hat{v}_i^{(k)}(t,m,n)\sqrt{\hat{\beta}_{k,i} + \sum_j \hat{\gamma}_{k,ij}(\hat{w}_i^{(k)}(t,m,n))^2} \tag{4}$$

Which is followed by upsampling:

$$\hat{v}_i^{(k)}(t,m,n) = \begin{cases} \hat{w}_i^{(k)}(t/\hat{s}_k, m/\hat{s}_k, n/\hat{s}_k) \\ \qquad \text{, if } (t/\hat{s}_k, m/\hat{s}_k, n/\hat{s}_k)\text{not int} \\ 0 \\ \qquad \text{,otherwise} \end{cases} \tag{5}$$

Where $\hat{s}_k$ is the upsampling factor for stage k. Finally, this is the followed by affine convolution:

$$\hat{u}_i^{(k+1)}(m,n) = \sum_j \left(\hat{h}_{k,ij} * \hat{u}_i^{(k)}\right)(m,n) + \hat{c}_{k,i} \tag{6}$$

Similarly, the sets of parameters for $\hat{h}$, $\hat{c}$, $\hat{\beta}$, and $\hat{\gamma}$form the optimizable vector $\hat{\theta}$. Downsampling/upsampling operations can be implemented in conjunction with their adjacent convolution, thus increasing computational efficiency

NS equation

The Navier-Stokes equations are fundamental equations describing fluid motion. Proposed by Claude-Louis Navier and George Gabriel Stokes in the mid-19th century, this system of equations elucidates the evolution of velocity and pressure fields within a fluid. It represents the motion equation embodying the conservation of momentum for viscous fluids. The Navier-Stokes equations form the cornerstone of fluid mechanics and are crucial for understanding and simulating fluid motion.

The NS equations are now widely used to simulate various physical systems, such as water flowing from a tap or the airflow over an aircraft wing. From a physics perspective, the NS equations operate effectively and seem to possess reliable predictive capabilities. Of course, the NS equations can also be employed to simulate the movement of oceans. Therefore, we consider introducing the NS equations as constraints with the expectation of further enhancing the performance of neural networks.

The NS equations are a type of partial differential equation, and depending on the specific situation, they can take various forms. Generally, there are three common cases: constant viscosity conditions, incompressible fluid, and incompressible fluid under constant viscosity conditions. Here, we choose the more general case of incompressible fluid under constant viscosity conditions as the prerequisite. Equation represents the NS equations under this condition.The NS equations in two dimensions can be expressed as Eq8 and Eq9.

$$\frac{\partial \vec{v}}{\partial t} + \vec{v}\nabla\vec{v} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\vec{v} + \vec{f} \tag{7}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = f_x - \frac{1}{\rho}\frac{\partial p}{\partial x} + \nu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) \tag{8}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = f_y - \frac{1}{\rho}\frac{\partial p}{\partial y} + \nu(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}) \tag{9}$$

Here,$\vec{v}$ represents the velocity vector, t represents time,ρ represents fluid density,ν represents kinematic viscosity,$\vec{f}$ represents body force, and ∇ represents the del operator.

To combine the NS equation as a constrain with data reconstruction, we calculate the NS value in original data and reconstructed data, and use it as a constrain to let the model better rebuild the data. Our NS Loss is given in Eq10

$$L_{NS} = MSE(NS_U, \hat{NS_U}) + MSE(NS_V, \hat{NS_V}) \tag{10}$$

$$NS_U = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - \nu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) \tag{11}$$

$$NS_V = \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} - \nu(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}) \tag{12}$$

we use Euler's forward difference to calculate the derivative items in the Eq12, and we set the kinematic viscosity $\nu = 1$. So our final loss is given in Eq13

$$L = L_R + \alpha L_{NS} \tag{13}$$

$$L_R = MSE(u, \hat{u}) + MSE(v, \hat{v}) \tag{14}$$

4 Experiment

Data Processing

We use the dataset of The South Sea of China, and because there exsist the Solid Wall conditions in the nearby of the coast which is a complex physical procession, we clip a 200×200 patch as shown in Figure 4. Our final dataset composed by the ocean current velocity in longitude direction(V) and latitude direction(U), with 10272 length on time dimension in which the interval is one hour. Considering the calculation ability, we set $t = 10, batch = 32$, and we train our model on two GTX 1080Ti with 11GB Memory.

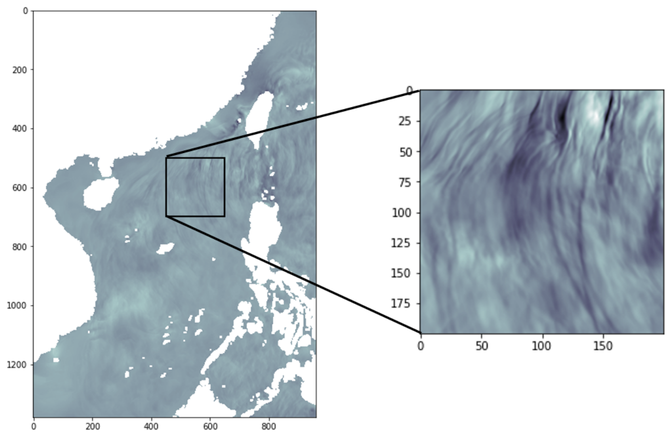

Figure 4: Data Clipping

Evaluation Metrics

To explore our model's ability of reconstruction quality, we set the Compression Rate(CR, given in Eq15) at 4 and 16. Follow the (Glaws, King, and Sprague 2020), we use mean average errors(MAE), mean square errors(MSE) and peak signal-to-noise ratio(PSNR, bigger is better).

$$CR = \frac{uncompressed\ size}{compressed\ size} \tag{15}$$

$$MAE(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{N}\sum_{i=1}^{N} |(x_i - \hat{x}_i)| \tag{16}$$

$$MSE(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{N}\sum_{i=1}^{N} (x_i - \hat{x}_i)^2 \tag{17}$$

$$PSNR(\mathbf{x}, \hat{\mathbf{x}}) = 10\log_{10}\left(\frac{\max(x, \hat{x})^2}{MSE(x, \hat{x})}\right) \tag{18}$$

Table 1: Compression Result

Model	CR	Test MSE	Test MAE	Test PSNR
Ours	8	34.82±2.10	0.97±0.09	6.65±0.32
Ours NS	8	**35.89±2.11**	**0.76±0.07**	**5.99±0.24**
Ours(100)	8	35.25±2.15	0.883±0.08	6.42±0.27
SVD	8	23.66±2.11	4.44±1.72	15.75±3.09
Ours	16	31.64±1.32	1.96±0.3	9.581± 0.642
Ours NS	16	31.71±1.54	1.904±0.25	9.55±0.584
Ours(100)	16	**32.25±2.02**	**1.781±0.182**	**9.208±0.452**
SVD	16	19.91±2.08	10.38±3.69	24.21±4.406

Compression Experiment

The compression result is shown in Table1, we can see that our model is outperformed than the original SVD method. NS represents as use NS Loss, and 100 represents star using NS loss on the 100 epochs. We can see that using NS loss can have better performance. Although the NS Loss calculation method is coarse, it can provide a limitation to the model when it reconstruct the data. And the reconstructed data is shown in Fig5. We can see that some small-scale vortices are not well reconstructed, we consider it is because the depth of model is too shallow to capture the finely detailed features in the data.

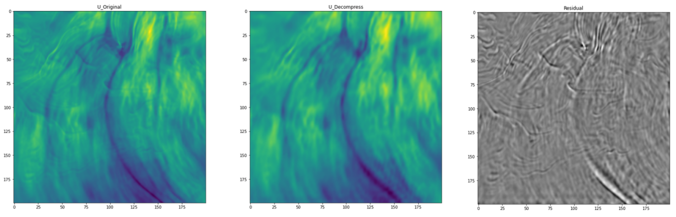

Figure 5: Decompression U data on CR=16

5 Conclusion

With the continuous development of deep learning, a large number of excellent compression algorithms have emerged in the field of image compression. However, in the field of ocean research, ocean data has the characteristics of floating-point type, which is different from the traditional integer type image, so the traditional image compression methods cannot be well applied to compress ocean data. We use the powerful learning ability of deep learning models to explore its application in ocean data compression. At the same time, considering that ocean data is different from traditional data, it not only includes the time dimension t, but also includes the inherent physical information. We use the NS equation in the ocean as prior information, and transform the NS equation into a computable equation and add it to the model training, and achieve good results. However, due to the low accuracy of our numerical calculation method, there is still room for improvement in improving the effect.

In order to further improve our model performance, we will improve it in the following aspects in the future: 1) Use more refined NS loss, such as high-order Runge-Kutta method to calculate the derivative. 2) Update the Backbone or add Transformer modules, such as NonLocal operators, Transformer feature extraction modules, etc. 3) Use the video compression algorithm framework to further improve the compression quality

References

Ballé, J.; Laparra, V.; and Simoncelli, E. P. 2016. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*.

Berres, A. S.; Turton, T. L.; Petersen, M. R.; Rogers, D. H.; Ahrens, J. P.; Rink, K.; Middel, A.; Zeckzer, D.; and Bujack, R. 2017. Video Compression for Ocean Simulation Image Databases. In *EnvirVis@ EuroVis*, 49–53.

Campin, J.-M.; Hill, C.; Jones, H.; and Marshall, J. 2011. Super-parameterization in ocean modeling: Application to deep convection. *Ocean Modelling*, 36(1-2): 90–101.

de Haan, K.; Rivenson, Y.; Wu, Y.; and Ozcan, A. 2020. Deep-Learning-Based Image Reconstruction and Enhancement in Optical Microscopy. *Proceedings of the IEEE*, 108(1): 30–50.

Dupont, E.; Loya, H.; Alizadeh, M.; Goliński, A.; Teh, Y. W.; and Doucet, A. 2022. COIN++: Neural compression across modalities. *arXiv preprint arXiv:2201.12904*.

Erichson, N. B.; Muehlebach, M.; and Mahoney, M. W. 2019. Physics-informed autoencoders for Lyapunov-stable fluid flow prediction. *arXiv preprint arXiv:1905.10866*.

Fang, Z. 2022. A High-Efficient Hybrid Physics-Informed Neural Networks Based on Convolutional Neural Network. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10): 5514–5526.

Glaws, A.; King, R.; and Sprague, M. 2020. Deep learning for in situ data compression of large turbulent flow simulations. *Physical Review Fluids*, 5(11): 114602.

He, D.; Yang, Z.; Peng, W.; Ma, R.; Qin, H.; and Wang, Y. 2022a. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5718–5727.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022b. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16000–16009.

Jamil, S.; Piran, M. J.; Rahman, M.; and Kwon, O.-J. 2023. Learning-driven lossy image compression: A comprehensive survey. *Engineering Applications of Artificial Intelligence*, 123: 106361.

Momenifar, M.; Diao, E.; Tarokh, V.; and Bragg, A. D. 2022. A physics-informed vector quantized autoencoder for data compression of turbulent flow. In *2022 Data Compression Conference (DCC)*, 01–10. IEEE.

Peng, B.; Alcaide, E.; Anthony, Q.; Albalak, A.; Arcadinho, S.; Cao, H.; Cheng, X.; Chung, M.; Grella, M.; GV, K. K.; et al. 2023. RWKV: Reinventing RNNs for the Transformer Era. *arXiv preprint arXiv:2305.13048*.

Tarasiou, M.; Chavez, E.; and Zafeiriou, S. 2023. ViTs for SITS: Vision Transformers for Satellite Image Time Series. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10418–10428.

Thuerey, N.; Holl, P.; Mueller, M.; Schnell, P.; Trost, F.; and Um, K. 2021. Physics-based deep learning. *arXiv preprint arXiv:2109.05237*.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 4489–4497.

Wang, K.; Chen, Z.; Zhu, M.; Yiu, S.-M.; Chen, C.-M.; Hassan, M. M.; Izzo, S.; and Fortino, G. 2023. Statistics-Physics-Based Interpretation of the Classification Reliability of Convolutional Neural Networks in Industrial Automation Domain. *IEEE Transactions on Industrial Informatics*, 19(2): 2165–2172.

Wang, Y.; Zhou, M.; and Zhou, F. 2023. Ocean data compression based on block SVD. In *Journal of Physics: Conference Series*, volume 2486, 012024. IOP Publishing.

Zhang, X.; Xu, H.; and Yang, M. 2022. . , 43(6): 783.

Zhu, M.; Li, G.; and Zhang, W. 2022. Research on UAV remote sensing multispectral image compression based on CNN. In *2022 3rd International Conference on Geology, Mapping and Remote Sensing (ICGMRS)*, 619–625.