The Senseiver: attention-based global field reconstruction from sparse observations

Javier E. Santos
Center for Nonlinear Studies
Los Alamos National Laboratory

Zachary Fox
Center for Nonlinear Studies
Los Alamos National Laboratory

Arvind Mohan
Computer, Computational, and Statistical Sciences
Los Alamos National Laboratory

Hari Viswanthan
Earth and Enviromental Sciences
Los Alamos National Laboratory

Nicholas Lubbers
Computer, Computational, and Statistical Sciences
Los Alamos National Laboratory

Abstract

The reconstruction of complex time-evolving fields from a small number of sensor observations is a grand challenge in a wide range of scientific and industrial applications. Frequently, sensors have very sparse spatial coverage, and report noisy observations from highly non-linear phenomena. While numerical simulations can model some of these phenomena in a classical manner, the inverse problem is not well-posed, hence data-driven modeling can provide crucial disambiguation. Here we present the *Senseiver*, an attention-based framework that excels in the task of reconstructing spatially-complex fields from a small number of observations. Building on the *Perceiver IO* model, the Senseiver reconstructs complex n-dimensional fields accurately using a small number of sensor observations by encoding arbitrarily-sized sparse sets of inputs into a latent space using cross-attention, which produces a uniform-sized space regardless of the number of observations. This same property allows very efficient training as a consequence of the being able to decode only a sparse set of observations as outputs. This enables efficient training of data with complex boundary conditions (sea temperature) and to extremely large and complex domains (3D porous media). We show that the Senseiver sets a new state of the art for three existing datasets, including real-world sea temperature observations, and pushes the bounds of sparse reconstruction using a large-scale simulation of two fluids flowing through a complex 3D domain.

1 Introduction

The goal of *sparse sensing* is to take a few sensor values from a field that we cannot fully observe, and use them to reconstruct the global field. Reconstructing a spatial field from sensor data has been a grand challenge in a wide range of industrial applications, medical, and scientific fields, including fluid flow [21], engineering and industrial monitoring [2] , earth systems observations [22], and biomedical engineering [19]. The common feature of these applications is low spatial sensor coverage (typically less than 1%), recording noisy, non-linear, dynamic phenomena. Some of these systems can be fully described by physics-based partial differential equations (PDEs), nevertheless, integrating field observations (for example, sensor measurements) back to the PDEs is challenging.

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

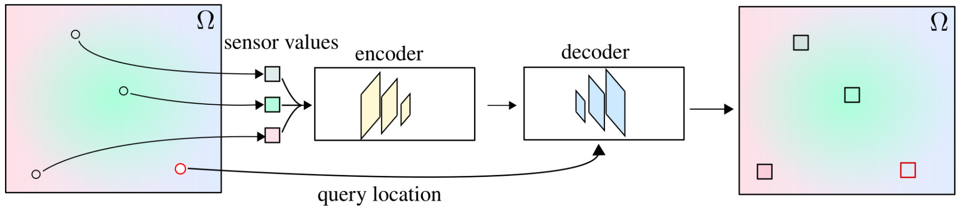

Figure 1: Overview of sparse reconstruction using the Senseiver model. Using sensor values and query locations which are sparse in the field domain. The sensor values are processed by an encoder, and the resulting latent representation is passed along with the query information to a decoder which estimates the field at a new location.

A variety of techniques have been developed using PDE-based [9, 10] and statistical [5, 27, 15] approaches. Still, widespread success has been elusive due to a lack a generic framework to incorporate measured data at arbitrary times and locations. As a result, machine learning models have become an attractive alternative [5, 27, 15], since these models have the capacity to learn complex relationships from non-linear data regardless of its sparsity, structure, and resolution. Machine learning models have the potential to be successful even when the governing PDEs of the system are not available. However, models to reconstruct sparse data should take in account certain restrictions: Real-world sensors are subject to practical constraints, such as physically limited sensor positions (e.g., floating buoys in the ocean [1]), and that covering a field exhaustively can be prohibitively expensive, if not impossible.

Recently, Fukami et al [8] introduced a method based on Voronoi tessellation of observations onto the prediction domain, followed by a refinement using a convolutional network. Their approach has the attractive upsides of allowing arbitrary sensor placement within a 2D mesh, and allowing to perform inference using sensor locations which differ from the ones used during training. Nevertheless, this approach inherits hurdles of deep convolutional networks, like the assumption of a gridded structure and high memory costs (for 3D domains, or in very big 2D arrays) difficulting scaling it to large domains, which are prevalent in real-world problems.

Attention mechanisms[23] have greatly improved over other architecture baselines for a variety of problems [24, 3, 14], and recently, the PerceiverIO framework [13] overcame a crucial computational bottleneck using cross-attention with latent arrays, thereby constraining the bulk of the network activations to a fixed-sized space regardless of input length. While this was viewed as a way to handle large sets of inputs (e.g. every pixel in an image), we exploit the fact that it also allows us to scale *down* the quantity of information fed into a network, resulting in a workflow we call the *Senseiver*. Rather than treating field reconstruction as a dense image problem, we encode the observed sensor values into a fixed sized vector, which can then be used with a decoder to provide estimation for the entire field. This workflow is shown in Fig. 1. Crucially, it can treat spatial data which does not live on a fixed, regular, Cartesian-type mesh, by embracing sparsity. These features shows promise in scaling our approach to datasets across a variety of scientific domains with arbitrary size. meshing and geometry.

In this work, we demonstrate examples of these advantages on several datasets, and we compare them to previously-proposed methods, showing drastic improvement in accuracy, scalability and efficiency. Beyond improvements in accuracy, we discuss additional benefits of the sparse processing model of the Senseiver, such as prediction of partial information, reduced memory requirements, and importantly, the ability to treat spatial data which does not live on a fixed, regular, Cartesian mesh. Our model goes further than previous propositions because it treats the problem sparsely, thereby allowing for training to domains of arbitrary size and structure.

2 The Senseiver: an attention-based learning approach

We aim to learn a compact representation of the state of a system from a small number of sensor observations at a given time. This encoded representation can be use to decode the state of the full system from sensor data. The input to our model is a set of N_s sensor observations s_i taken at time t, $\{s_1, s_2, \ldots s_{N_s}\}_t$, with $s_i \in \mathbb{R}^{N_c}$, where N_c corresponds to the number of channels recorded by the sensors. The system has a domain Ω where a set of sensor locations ($\{\chi_{s_1}, \chi_{s_2}, \dots \chi_{s_n}\}$, with $\chi_{s_i} \in \mathbb{R}^N$) are extracted. The details of the computational implementation are shown in Appendix A.

The Senseiver has three main components: 1) a spatial encoder P_E that maps a spatial coordinate χ to an array of spatial encodings a using sine-cosine embeddings [23], 2) an attention-based encoder E that maps the spatial encodings of the sensor positions a and their values s to a latent space z, and 3) an attention-based decoder D which is queried to obtain the value of the system at a specified position χ_q. This architecture can be described by the following equations:

$$
\begin{aligned}
a &= P_E(\chi), \\
z &= E(a_s, s) = E\left(P_E(\chi_s), s(\chi_s, t)\right), \\
\hat{s}_q &= D(z, a_q) = D(z, P_E(\chi_q)).
\end{aligned}
$$

3 Results

We chose datasets representing two major classes of scientific problems with varying degrees of complexity, and our results are shown in Fig. 2, where we reported the L2 norm (ϵ). **Category 1: Cyclic and quasi-cyclic phenomena**: These datasets exhibit periodicity or seasonality, as seen in some applications in engineering and climate sciences. In this context, the model is able to generalize extremely well with a very small amount of data. We tested two datasets of this kind, a) a 2D unsteady flow pass a cylindrical obstacle [4] which results in a von Kármán vortex street, an alternating shedding of left- and right-handed vortices in the flow field behind the cylinder. We trained our model to reconstruct the simulation based on eight sensor locations as proposed by [8] and sensor locations at the inlet/outlet boundaries, which represents a more realistic configuration, for example, in a laboratory experiment. b) The NOAA sea surface temperature [18]. This real-world dataset was collected from satellite and ship-based observations through time. The data comprise weekly observations of the sea surface temperature of the planet Earth. During training we do not use any information about field values on the continents, because their is no recorded value to reconstruct, which saves computational time (since the continents are 32% of the computational domain). We tested the trained model in data spanning from 2001 to 2018 to show the models ability to extrapolate.

Category 2: Acyclic and non-linear chaotic phenomena: These datasets are prevalent in several applications and are characterized by highly chaotic dynamics that lack periodicity in time or space and present spatial structures that are complex and vary greatly throughout time. The goal in this section is to demonstrate the capability of the model to learn complex time-dependent features and to scale to large domains. The first dataset on this section a) turbulent fluid flow through a channel[7]. The flow field data is obtained by a slicing a three-dimensional numerical simulation of incompressible flow in a channel at a Reynolds number of 180. The target of interest in this case is the velocity of the middle slice in the direction of flow. The second dataset b) two immiscible fluids flowing through a complex 3D medium comprised of spherical obstacles. A simulation was run using the lattice-Boltzmann method for 4 days in 120 CPU cores to generate this dataset.The goal of this test case is to assess the capabilities of our model to train with very large domains, which so far have been a challenge for machine learning methods. The computational domain is 128x128x512 and we collected 100 frames (over 1.6 billion points). Similarly to the sea temperature dataset, in this domain around 70% of the grid cells have no property value to reconstruct (solid boundaries), hence the training is sped-up by a significant factor compared to current approaches, as our method identifies these areas of no data.

Discussion: The flexibility of the proposed architecture allows to explore many use cases that were not possible until now. Although we tried to cover as much ground as possible, there were many things left unexplored. For instance, non-Cartesian or unstructured grids can be used during training and/or inference, on the same vein, the resolution of the field prediction can be increased by computing the desired property at intermediate intervals. The sensors can can have more than 1 channel which can record different things. Also, multiple decoding heads can be trained to predict outputs with different boundary conditions or different downstream tasks (segmentation, classification, inference of a different property). The positional encodings could be used to train a model to have forecasting capabilities. During the development of this project, an attempt was made to encode time using sine-cosine encodings without success. On the other hand, we tried utilizing a trainable array where each time increment (dt) corresponded to one vector, this was successful but we found it unpractical

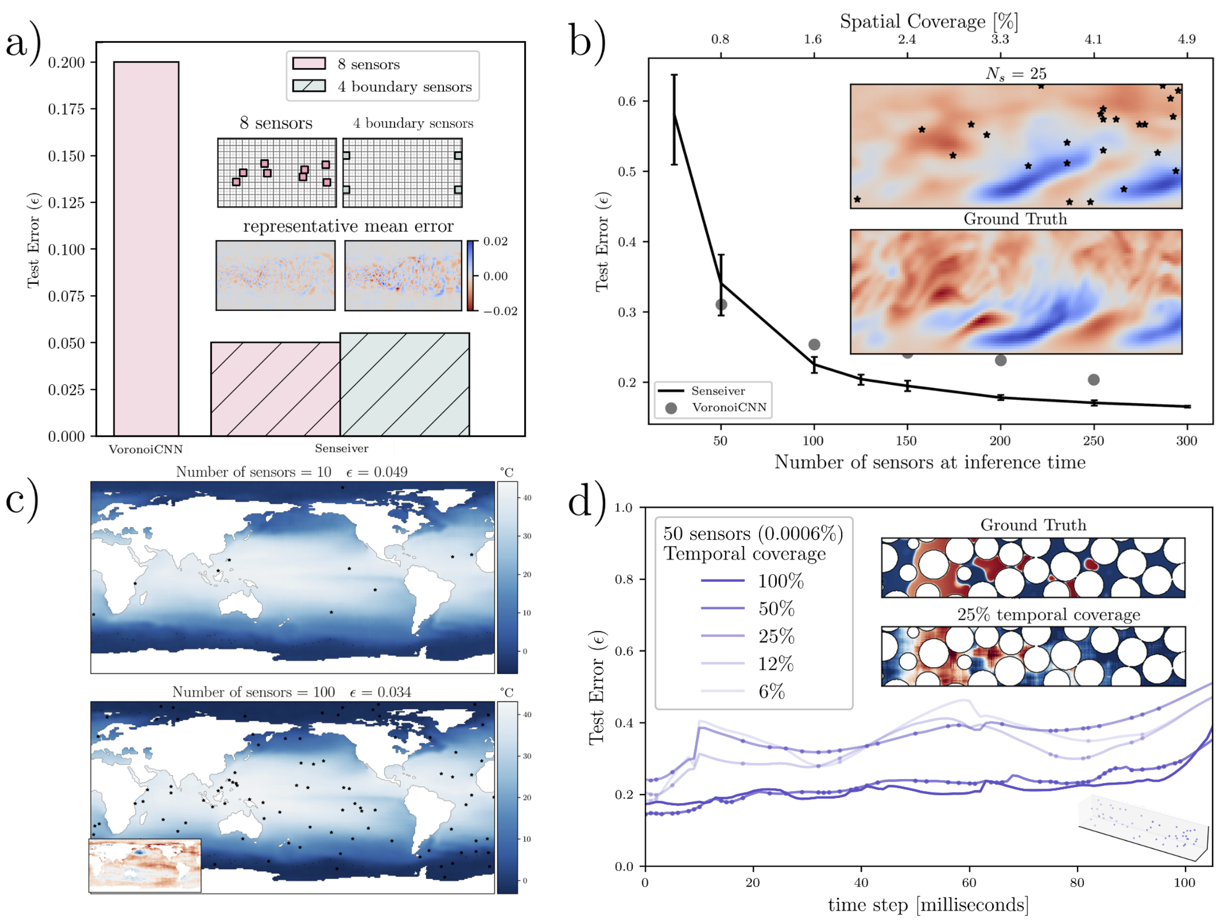

Figure 2: Senseiver performance in the different datasets. **a)** Even with just 4 sensors at the boundary, the Senseiver is able to reconstruct the the entire simulation faithfully with a negligible drop in accuracy compared to the eight sensor configuration. **b)** Performance of the model on the turbulent dataset varying the number of sensors and their locations at inference time. We tested our trained model with ten different random sensor locations. The plot sows the 10th and 90th percentile as bounds of the error plot and the average of the with a line. Predictions are shown for the 25 sensor reconstruction, finer details are reconstructed when more sensors are added. **c)** In the sea temperature forecasting, just ten sensors allow for a very strong reconstruction performance of $\epsilon = 0.04$. Ten sensors constitute a total spatial coverage of 0.0154 %. By adding more sensors yields the overall test error goes down and missing details are added to the local temperature field as seen in subpanel. **d)** Performance of the model versus temporal coverage in the large 3D simulation. Each line represents a trained model and the points represent the training data. The Reconstructions of the density field for time step 95 using 25 sensors are shown.

since it requires the model to visit every time increment (dt) during training. One interesting avenue for future work could be to enforce known physics through the positional encodings.

4 Conclusion

From an information theoretic perspective, sparse sensing is an inverse modeling problem which maps sparse, low-dimensional measurements to a dense high-dimensional state. The goal of sparse sensing algorithms is to obtain the best possible estimates useful enough to inform practical applications, since there are few other viable alternatives. We propose an attention-based neural network architecture, the *Senseiver*, to encode a compact representation of large systems. We validated the effectiveness of our method with extensive demonstrations on different datasets of interest to the sparse-sensing community, and also on a complex, realistic three-dimensional fluids dataset for the first time in literature. Our approach offers new capabilities for large, practical applications compared to the state-of-the-art convolutional neural network architectures [8] by demonstrating higher accuracy with lower memory footprint. Three examples of global flow reconstruction from local sensor measurements demonstrated the accuracy and robustness of our method. Sparse sensing of fluid flow data, especially turbulence, is extremely challenging due to non-linearity and chaos. Additionally, a low sensor coverage makes the task harder since the sensors can have non-unique reconstructions. Compared to the previous efforts, our model scales effectively in large domains of high dimensionality.

Sparse sensing of fluid flow data, especially turbulence, is extremely challenging due to non-linearity and chaos. Four examples of global flow reconstruction from local sensor measurements demonstrated the accuracy and robustness of our method. Compared to the previous efforts, a key advantage of the Senseiver is using a query based decoder, allowing us to predict domains of arbitrary sizes in a *sequential* manner. This decoupling of the query process from the dimensionality of the dataset makes it extremely memory efficient and allows our model to scale effectively to large domains. It is possible to tailor the Senseiver to a particular applications by incorporating physics constraints. Another potential application could be to use it as a super-resolution model, since the model has more information available to it than in a sparse-sensing problem. In summary, this work scratches the surface of what is possible with attention-based architectures for sparse sensing.

5 Acknowledgments

JES and ZF gratefully acknowledge the support of the U.S. Department of Energy through the LANL/LDRD Program and the Center for Non-Linear Studies (CNLS) for this work. JES would like to thank Andrew Jaegle and João Carreira for their useful suggestions and Jack Burdick and Jack Moriarty for the useful discussion on the attention mechanism. Finally, we are grateful to the developers of the many software packages used throughout this project including, but not limited, to PyTorch [20], Numpy [11], Vedo [17], Matplolib [12], and Pytorch-lightning [6].

6 Broader Impact

With the advent of low cost sensors and the IoT revolution, sparse-sensing is seeing a resurgence in interest for many scientific applications. In a chaotic system, the physics implies that we can only approximately estimate a system from a few sparse measurements. The number of sensors, and the relative positions of these sensors are of utmost importance. When we use too few sensors and/or place them in locations where its impossible to fully observe the system, as it often happens in practice, the problem becomes more challenging. In light of these fundamental limitations, we show that our approach not only outperforms current methods in literature, but also presents new capabilities for large, realistic datasets that were previously intractable. Numerous applications ranging from structural health monitoring of aerospace systems [2] and civil infrastructure [25], earth system monitoring [16] and weather forecasting [26] have complex governing physics beyond the scope of current PDE-based sparse-sensing techniques. These can be promising application areas for our method, since it is generic and not restricted to a specific problem, as shown in the diverse datasets used in this work.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A] The code will be released upon the acceptance of the work
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] They are shown in the results plot
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A] The model is very light-weight. It trains on a standard GPU.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

We define the multi-head attention mechanism as in [23]. We have a query Q, a key K, and a value V, which are vectors whose components are independent random variables with mean 0 and variance 1. Starting with the equation for scaled dot product attention,

$$\mathrm{Attention}(Q, K, V) = \mathrm{Softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V, \tag{1}$$

where d_k is the dimension of K and Q, which is used as a factor that scales the variance of the dot-product to 1. Multi-head attention applies a set of H separate linear transformations to Q, K and V and computes the attention mechanism on each linear transformation h. It then concatenates the outputs from each head and computes a final linear transformation to form the output.

$$\mathrm{MultiHead}(Q, K, V) = [O_1 \oplus \cdots \oplus O_H]\, W_M \tag{2}$$

$$\text{where } O_h = \mathrm{Attention}(QW_h^Q, KW_h^K, VW_h^V). \tag{3}$$

We use two specific flavors of the multi-head attention mechanism to build the encoder of the Senseiver. Once the sensor data s_i and corresponding positional encodings a_i are concatenated ($x^{(0)}$) and processed using a dense linear layer ($x^{(1)}$), the matrix $x^{(1)}$ is passed through a cross-attention layer in which Q is a learnable latent array denoted $Q_{\mathrm{in}} \in \mathbb{R}^{\mathrm{N_g} \times \mathrm{N_q}}$, and K, V are both the output of the previous layer $x^{(1)} \in \mathbb{R}^{N_s \times M_1}$. Therefore, the set of weight matrices for each head $h = 1, 2, \ldots, H$ in Eq. 2 have dimensions $W_h^Q \in \mathbb{R}^{N_q \times N_k}$, $W_h^K \in \mathbb{R}^{M_1 \times N_k}$, $W_h^V \in \mathbb{R}^{M_1 \times N_o}$. The dimension of $O_h \in \mathbb{R}^{N_g \times N_o}$, and therefore the dimension of $W_M \in \mathbb{R}^{H \cdot N_o \times N_M}$, leading to the output $x^{(2)} N_g \times N_M$.

After the initial multi-head cross attention, $x^{(2)}$ is fed into a multi-head self-attention layer, in which Q, K and V in Eq. 2 are all given by $x^{(2)}$. The weight matrices are W_h^Q and $W_h^K \in \mathbb{R}^{N_M \times \tilde{N}_k}$, $W_h^V \in \mathbb{R}^{N_M \times \tilde{N}_o}$. The dimension of $O_h \in \mathbb{R}^{N_g \times \tilde{N}_o}$, and therefore $W_M \in \mathbb{R}^{H \cdot \tilde{N}_o \times N_M}$, leading to the output $x^{(3)} \in \mathbb{R}^{N_g \times N_M}$. These two sequential operations (cross-attention and self-attention) constitute one encoding block. In our experiments, we used three of these blocks where weights are shared. $x^{(3)}$ is recursively fed back through these blocks to yield the encoded sensor data $z \in \mathbb{R}^{N_g \times N_M}$.

Next, in the decoder section, this latent vector z is concatenated with the sine-cosine encoding of the query position, a_q. The vector a_q is stacked N_g times to match the number of rows in z, forming a matrix $v \in \mathbb{R}^{N_g \times N_M + 2 \cdot D \cdot N_f}$. We then perform multi-head cross-attention on v, and pass the output to a linear layer which yield an output $\hat{s} \in \mathbb{R}^{N_{\mathrm{out}}}$.

References

[1] Cristina Albaladejo et al. “A Low-Cost Sensor Buoy System for Monitoring Shallow Marine Environments”. In: *Sensors 2012, Vol. 12, Pages 9613-9634* 12.7 (July 2012), pp. 9613–9634. ISSN: 1424-8220. DOI: 10.3390/S120709613. URL: https://www.mdpi.com/1424-8220/12/7/9613/htm%20https://www.mdpi.com/1424-8220/12/7/9613.

[2] Steven L. Brunton et al. “Data-Driven Aerospace Engineering: Reframing the Industry with Machine Learning”. In: *AIAA Journal* 59.8 (July 2021), pp. 1–26. ISSN: 0001-1452. DOI: 10.2514/1.J060131. URL: https://arc.aiaa.org/doi/10.2514/1.J060131.

[3] Aakanksha Chowdhery et al. “PaLM: Scaling Language Modeling with Pathways”. In: (Apr. 2022). URL: http://arxiv.org/abs/2204.02311.

[4] Tim Colonius and Kunihiko Taira. “A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions”. In: *Computer Methods in Applied Mechanics and Engineering* 197.25-28 (Apr. 2008), pp. 2131–2146. ISSN: 0045-7825. DOI: 10.1016/J.CMA.2007.08.014.

[5] Ranjan Das. “A simulated annealing-based inverse computational fluid dynamics model for unknown parameter estimation in fluid flow problem”. In: *http://dx.doi.org/10.1080/10618562.2011.632375* 26.9-10 (Oct. 2012), pp. 499–513. ISSN: 10618562. DOI: 10.1080/10618562.2011.632375. URL: https://www.tandfonline.com/doi/abs/10.1080/10618562.2011.632375.

[6] William Falcon et al. "PyTorch Lightning". In: *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning* 3 (2019).

[7] Koji Fukagata, Nobuhide Kasagi, and Petros Koumoutsakos. "A theoretical prediction of friction drag reduction in turbulent flow by superhydrophobic surfaces". In: *Physics of Fluids* 18.5 (May 2006), p. 051703. ISSN: 1070-6631. DOI: `10.1063/1.2205307`. URL: `https://aip.scitation.org/doi/abs/10.1063/1.2205307`.

[8] Kai Fukami et al. "Global field reconstruction from sparse sensors with Voronoi tessellation-assisted deep learning". In: *Nature Machine Intelligence 2021 3:11* 3.11 (Oct. 2021), pp. 945–951. ISSN: 2522-5839. DOI: `10.1038/s42256-021-00402-2`. URL: `https://www.nature.com/articles/s42256-021-00402-2`.

[9] Marco Gherlone et al. "Shape sensing of 3D frame structures using an inverse Finite Element Method". In: *International Journal of Solids and Structures* 49.22 (Nov. 2012), pp. 3100–3112. ISSN: 0020-7683. DOI: `10.1016/J.IJSOLSTR.2012.06.009`.

[10] Yan Gu et al. "Application of the meshless generalized finite difference method to inverse heat source problems". In: *International Journal of Heat and Mass Transfer* 108 (May 2017), pp. 721–729. ISSN: 0017-9310. DOI: `10.1016/J.IJHEATMASSTRANSFER.2016.12.084`.

[11] Charles R. Harris et al. "Array programming with NumPy". In: *Nature 2020 585:7825* 585.7825 (Sept. 2020), pp. 357–362. ISSN: 1476-4687. DOI: `10.1038/s41586-020-2649-2`. URL: `https://www.nature.com/articles/s41586-020-2649-2`.

[12] J D Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: `10.1109/MCSE.2007.55`.

[13] Andrew Jaegle et al. "Perceiver: General Perception with Iterative Attention". In: (2021). URL: `http://arxiv.org/abs/2103.03206`.

[14] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature 2021 596:7873* 596.7873 (July 2021), pp. 583–589. ISSN: 1476-4687. DOI: `10.1038/s41586-021-03819-2`. URL: `https://www.nature.com/articles/s41586-021-03819-2`.

[15] Jean Christophe Loiseau, Bernd R. Noack, and Steven L. Brunton. "Sparse reduced-order modelling: sensor-based dynamics to full-state estimation". In: *Journal of Fluid Mechanics* 844 (June 2018), pp. 459–490. ISSN: 0022-1120. DOI: `10.1017/JFM.2018.147`. URL: `https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/sparse-reducedorder-modelling-sensorbased-dynamics-to-fullstate-estimation/DD8216958838BBD935E024C01B41515A`.

[16] Manuel Lopez-Radcenco et al. "Locally-adapted convolution-based super-resolution of irregularly-sampled ocean remote sensing data". In: *Proceedings - International Conference on Image Processing, ICIP* 2017-September (Feb. 2018), pp. 4307–4311. ISSN: 15224880. DOI: `10.1109/ICIP.2017.8297095`.

[17] Marco Musy et al. *Vedo*. Mar. 2021. DOI: `10.5281/zenodo.4609336`. URL: `https://zenodo.org/record/4609336`.

[18] National Oceanic and Atmospheric Administration. *NOAA Physical Sciences Laboratory*. URL: `https://psl.noaa.gov/`.

[19] Alessandro Paoli et al. "Sensor Architectures and Technologies for Upper Limb 3D Surface Reconstruction: A Review". In: *Sensors 2020, Vol. 20, Page 6584* 20.22 (Nov. 2020), p. 6584. ISSN: 1424-8220. DOI: `10.3390/S20226584`. URL: `https://www.mdpi.com/1424-8220/20/22/6584/htm%20https://www.mdpi.com/1424-8220/20/22/6584`.

[20] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: NeurIPS (2019). URL: `http://arxiv.org/abs/1912.01703`.

[21] N. P. Ramskill et al. "Fast imaging of laboratory core floods using 3D compressed sensing RARE MRI". In: *Journal of Magnetic Resonance* 270 (Sept. 2016), pp. 187–197. ISSN: 1090-7807. DOI: `10.1016/J.JMR.2016.07.017`.

[22] Bertrand Rouet-Leduc, Claudia Hulbert, and Paul A. Johnson. "Continuous chatter of the Cascadia subduction zone revealed by machine learning". In: *Nature Geoscience 2018 12:1* 12.1 (Dec. 2018), pp. 75–79. ISSN: 1752-0908. DOI: `10.1038/s41561-018-0274-6`. URL: `https://www.nature.com/articles/s41561-018-0274-6`.

[23] Ashish Vaswani et al. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems* 2017-Decem.Nips (June 2017), pp. 5999–6009. ISSN: 10495258. URL: `http://arxiv.org/abs/1706.03762`.

[24] Jiahui Yu et al. "CoCa: Contrastive Captioners are Image-Text Foundation Models". In: (May 2022). URL: http://arxiv.org/abs/2205.01917.

[25] Fuh-Gwo Yuan et al. "Machine learning for structural health monitoring: challenges and opportunities". In: *https://doi.org/10.1117/12.2561610* 11379.23 (Apr. 2020), p. 1137903. ISSN: 1996756X. DOI: 10.1117/12.2561610. URL: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11379/1137903/Machine-learning-for-structural-health-monitoring-challenges-and-opportunities/10.1117/12.2561610.full%20https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11379/1137903/Machine-learning-for-structural-health-monitoring-challenges-and-opportunities/10.1117/12.2561610.short.

[26] Haoxuan Yuan, Qiangyu Zeng, and Jianxin He. "Adaptive Regularized Sparse Representation for Weather Radar Echo Super-Resolution Reconstruction". In: *2021 International Conference on Electronic Information Engineering and Computer Science, EIECS 2021* (Sept. 2021), pp. 33–38. DOI: 10.1109/EIECS53707.2021.9587997.

[27] Hongming Zhou et al. "Compressed representation learning for fluid field reconstruction from sparse sensor observations". In: *Proceedings of the International Joint Conference on Neural Networks* 2015-September (Sept. 2015). DOI: 10.1109/IJCNN.2015.7280519.