Incorporating Higher Order Constraints for Training Surrogate Models to Solve Inverse Problems

Jihui Jin
Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332
jihui@gatech.edu

Nicholas Durofchalk
Physics
Naval Postgraduate School
Monterey, CA 93943

Richard Touret
Mechanical Engineering
Georgia Institute of Technology
Atlanta, GA 30332

Karim G. Sabra
Mechanical Engineering
Georgia Institute of Technology
Atlanta, GA 30332

Justin K. Romberg
Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332

Abstract

Inverse problems describe the task of recovering some underlying signal given some observables. Typically, the observables are related via some non-linear forward model applied to the underlying signal. Inverting the non-linear forward model can be computationally expensive, as it involves calculating the adjoint when computing a descent direction. Rather than inverting the non-linear model, we instead train a surrogate forward model and leverage modern auto-grad libraries to solve for SSPs within a classical optimization framework. Current methods to train surrogate models are done in a black box supervised machine learning fashion and don't take advantage of any existing knowledge of the forward model. In this article, we propose a simple regularization method to enforce constraints on the gradients of the surrogate model in addition to the output to improve overall accuracy. We demonstrate the efficacy on an ocean acoustic tomography (OAT) example that aims to recover ocean sound speed profile (SSP) variations from acoustic observations (e.g. eigenray arrival times) within simulation of ocean dynamics in the Gulf of Mexico.

1 Introduction

Inverse problems arise in many scientific applications where the goal is to reconstruct some unknown signal, image or volume of interest from indirect observations. The forward process, or the mapping from the data to observations, is typically well known usually through modeling the physical process. However, inverting the model is often ill-posed or even non-invertible. More formally, let us consider the task of recovering some signal $\boldsymbol{x}$ from observations $\boldsymbol{y}$ that are related by some potentially non-linear forward model F via

$$\boldsymbol{y} = F(\boldsymbol{x}) + \boldsymbol{\eta}, \tag{1}$$

where $\boldsymbol{\eta}$ encapsulates noise or other perturbations. Our forward model F represents a computational model of the underlying physics of the measurement process. Classical solutions involve modeling the

NeurIPS 2021 AI for Science Workshop.

forward process with extremely high accuracy and then attempting to invert a stabilized or linearized variant, which often requires heavy domain knowledge.

Another approach to handle the ill-posed nature of the inversion task is to cast the problem as an optimization task and incorporate regularization. A regularizer is a measure of how well the proposed solution fits some known, and often hand-crafted, prior. This term makes the inversion well-posed by biasing towards certain solutions. Often times, the task is too large of dimension to invert directly. In other cases, the selected regularizer does not allow for a closed form solution. In these instances, the solution is solved for an in iterative fashion.

For many physics based inverse problems, computing a forward pass often involves solving a set of partial differential equations (PDEs) and can be far too computationally expensive to employ iterative solvers that can require hundreds if not thousands of iterations until convergence. Furthermore, calculating the gradient of such a forward model becomes even more daunting of a task. An increasingly popular approach to bypass this issue takes advantage of machine learning. Given a paired dataset of example data $\boldsymbol{x}$ and observations vy, a **surrogate forward model** in the form of a deep neural network is trained to map $\boldsymbol{x}$ to $\boldsymbol{y}$ through supervised learning. Once trained, the surrogate model can be used in place of the physics based forward model in standard optimization schemes. By leveraging existing auto-grad libraries [17], the gradient with respect to the input can easily be calculated, making iterative solvers feasible for recovering a solution $\boldsymbol{x}$ given some observations $\boldsymbol{y}$.

While as the surrogate forward model offers promise for solving inverse problems, it is trained in a black box fashion and does not incorporate any knowledge of the physical forward model. It provides no guarantees that the gradients produced by such a model will be useful for recovering the minima of some optimization function. Sobolev Training [3] offers an interesting approach by training not only on the output of the surrogate model, but the Jacobian as well. However, calculating and storing ground truth Jacobians of the physics model for even moderate dimensions quickly grows infeasible. Instead, we propose a simple and efficient Jacobian regularization technique when training the surrogate model that does not require any additional calculation and/or storage of the ground truth. Training in such a fashion will both increase the accuracy of the surrogate model as well as provide more stability when using it for inversion. We will demonstrate the efficacy of our proposed regularization on an ocean acoustic example.

2 Background

2.1 Direct Inversion

The task of directly inverting some potentially non-linear forward model F is often non-trivial or mathematically impossible. Instead, a more stable alternative is to iteratively solve for $\boldsymbol{x}$ given some observations $\boldsymbol{y}$. This is done by formulating the reconstruction as solving a least squares problem. This would take the form of

$$\hat{x} = \min_{x} \frac{1}{2} \left\| F(\boldsymbol{x}) - \boldsymbol{y} \right\|^2 + R(\boldsymbol{x}), \tag{2}$$

where R is some regularizer chosen to stabilize the inversion process. For many problems of interest, F is ill-posed. The regularizer R plays the role of biasing Equation 2 to more favorable solutions. Common examples of regularization include ℓ_2, ℓ_1, and TV. Each regularizer encourages certain properties on the estimated signal $\hat{\boldsymbol{x}}$ (e.g. smoothness, sparsity, piece-wise constant, etc.) and is often chosen based on the context of the task. This can be solved for iteratively via

$$\hat{\boldsymbol{x}}^{k+1} = \hat{\boldsymbol{x}}^{k} - \gamma J_F(\boldsymbol{x}^{k})^{\top} (F(\hat{\boldsymbol{x}}^{k}) - \boldsymbol{y}) - \nabla R(\boldsymbol{x}), \tag{3}$$

where γ is a tuned step size and $J_F(\hat{\boldsymbol{x}}^k)$ is the Jacobian of F evaluated at $\hat{\boldsymbol{x}}^k$.

2.2 Neural Adjoint

Iterative solutions such as Equation (3) often rely on computing the gradient with respect to some loss. The adjoint of F arises in the gradient calculation and can be computationally expensive to solve for or apply, especially for non-linear F. In some instances, this can involve solving an entirely different set of partial-differential equations.

The neural adjoint method [18] was proposed to tackle inverse problems with more computationally intensive forward models. Instead of training a parameterized model G_θ to learn an inverse mapping, it is trained to emulate the forward model [2, 6] and eventually serves the role of a surrogate forward model. This done in a supervised learning fashion, often with a simple mean-squared error loss. Not only does this provide a cheaper/quicker alternative to the physics based forward model, if trained with existing auto-grad libraries such as Pytorch [17], this also allows for efficient computation of the gradient with respect to the input, bypassing the need to explicitly solve for the adjoint when calculating the gradient.

Once trained, the parameters are fixed and the model G_θ is substituted for F in Equation (2) or other similar optimization frameworks. The auto-grad libraries are then used to efficiently compute a gradient with respect to the input $\boldsymbol{x}$, making it possible to iteratively solve for the best estimate $\hat{\boldsymbol{x}}$ that fits some observations $\boldsymbol{y}$. Existing works primarily focused on lower dimensional examples where the test set heavily overlapped with the training set and thus a simple "boundary-loss" regularizer was sufficient to produce accurate results.

This approach is an iterative method, and thus is slower than alternative methods that aim to directly learn the inverse [16, 8, 21, 5, 13, 20]. However, unlike learned inversion methods that yield a single estimate and are prone to overfitting to the training set, neural adjoint allows for the incorporation of various regularizers such as ℓ_1, ℓ_2, etc, to guide the optimization towards specific solutions. In addition, one can also restrict the optimization to some pre-determined basis that better represents the data while reducing the dimensionality. This method, although heavily inspired by existing solutions for inverse problems, does not currently take advantage of the forward model. It is also heavily dependent on learning the forward model accurately and thus susceptible to overfitting to the training set and not generalizing well should the test set distribution differ.

2.3 Ocean Acoustic Tomography

Although the methods described in our paper should be generalizable to any forward model, we demonstrate on an ocean acoustic tomography problem. Knowledge of the sound speed variations is essential for accurate predictions of sound propagation in the ocean and the various acoustic remote sensing applications that rely on these predictions [9, 11, 4]. Most commonly, ocean sound speed is estimated using empirical formulae based on the local values of standard oceanographic parameters (such as ocean's temperature, salinity and density)[9, 11]; thus ocean sound variability in a given ocean volume can be estimated provided a sufficiently dense spatial and temporal sampling of these oceanographic parameters is available throughout the whole volume of interest. An alternative approach that alleviates this dense sampling requirement is to directly leverage the fundamental relationship between acoustic observations (e.g. arrival times measurements) and the actual volumetric spatio-temporal variability of the ocean sound speed profiles (SSPs). Ocean acoustic tomography (OAT), as introduced by Munk and Wunsch in 1979 [14], refers to the process of using the changes in the acoustic measurements collected between multiple pairs of sources and receivers as inputs of an inverse problem aimed at reconstructing SSPs variations (with respect to a known reference environment) within the ocean volume for 3D applications (or more commonly, selected ocean slices for 2D applications) sampled by the propagating acoustic waves between these sensor pairs. One challenge to this "inverse problem"-based approach for estimating SSPs is that the relationship between acoustic observations (typically arrival time measurements) and actual SSPs is fundamentally nonlinear. However, fractional SSPs fluctuations are typically small compared to the baseline SSP values (typically $< 1\%$). Classical OAT methods simplify this approach by linearizing the relationship between the estimated SSPs perturbations and the measured variations in arrival times of stable rays propagating between the source and receivers pairs (those rays being identified by standard ray-tracing methods using the baseline reference environment). This provides a simpler and numerically tractable solution for the OAT inverse problem [14, 1, 15, 19].

3 Method

3.1 Learning the Forward Model

NA aims to learn an accurate surrogate forward model to replace existing physics simulators. More formally, assume that we are given a forward model $F : \boldsymbol{c} \rightarrow \boldsymbol{\tau}$ (e.g. Bellhop Ray Tracing or some PDE solver) that maps our SSP $\boldsymbol{c} \in \mathbb{R}^m$, where m denotes the size of the discretization of the ocean, to some arrival time observations $\boldsymbol{\tau} \in \mathbb{R}^n$ where n denotes the total number of arrivals . We then train a neural net G_θ to approximate this same mapping. The model is trained using the mean squared error loss on a dataset of paired example $(\boldsymbol{c}_i, \boldsymbol{\tau}_i)$

$$\sum_i \|G_\theta(\boldsymbol{c}_i) - \boldsymbol{\tau}_i\|^2 . \tag{4}$$

In addition to architecture hyperparameters, we also test two regularization methods for constraining the network. The first is spectral normalization [12] that constrains the spectral norm of individual layers. The second is Jacobian regularization in a similar manner to [7]. Both regularization methods are designed to encourage a more robust and stable learning of the forward model as well as a smoother gradient. The best method amongst no regularization, spectral norm only, Jacobian regularization, and both, are selected based on the performance on a validation set.

3.2 Enforcing Higher Order Constraints

Although the NA model is trained to emulate the physics forward model, no physics-based constraints are placed during the learning process; it is entirely data-driven. An open area of exploration will be focused on enforcing higher order gradient accuracy. Training on both the output and corresponding Jacobian is known as Sobolev training [3]. However, applying this to such a large-dimensional learning problem presents its own challenges. While as auto-grad allows us to cheaply compute the gradients with respect to a single scalar output, using this to compute a full Jacobian scales linearly with the dimension of the output. Furthermore, evaluating and storing the ground truth Jacobian at a wide, diverse sampling of data-points will have large memory requirements. It can be an inefficient use of resources, especially when the outputs only depend on a subset of the inputs leading to a majority of the Jacobian being either zero or very close to zero. In the case of OAT, this is intuitively because an arrival time observation only interacts with a fairly small subset of the ocean.

As an alternative, we can use a linear Taylor series approximation. Let us denote G as the trained emulator (parameters dropped for clarity) and F as the physics forward model. Then, a taylor series expansion around point $\boldsymbol{c}$ would give

$$F(\boldsymbol{z}) \approx F(\boldsymbol{c}) + J_F(\boldsymbol{c})(\boldsymbol{z} - \boldsymbol{c}), \tag{5}$$

where $J_F(\boldsymbol{c})$ denotes the Jacobian of F at point $\boldsymbol{c}$ and second order terms are omitted. If we rearrange this and substitute our emulator G for the function F in the Jacobian calculation, we get that

$$J_G(\boldsymbol{c})(\boldsymbol{z} - \boldsymbol{c}) \approx F(\boldsymbol{z}) - F(\boldsymbol{c}). \tag{6}$$

Note that the left hand side is a simple Jacobian-vector product, a quantity trivial for auto-grad to compute in a single pass. The right hand side is a difference between the output of the physics forward model evaluated at two locations, i.e. the observations that we already currently store and use. Thus we can construct an approximate Jacobian enforcing loss using Equation (6) by selecting two points $\boldsymbol{c}, \boldsymbol{z}$ that are close to each other (to avoid violating the assumptions of the expansion) and evaluating a MSE on the difference between the two sides of the equation.

Before training, we can compute a neighborhood graph of viable pairs of points $\boldsymbol{c}$ and apply the above mentioned Jacobian loss at random during training. Our hope is that by training in such a fashion we encourage faster convergence (since the Jacobian will also have information) as well as more generalizability to unseen or underrepresented regions. Furthermore, certain studies [7] have shown that reguarlizing the Jacobian leads to more stability with respect to input perturbations. Although designed to be robust to adversarial attacks, the encouraged properties should also lead to more robust predictions during the inference phase of the neural adjoint method. The above loss is a minor tweak on the existing regularization technique already used, so should not lead to additional computational costs when training the surrogate model.

3.3 Performing Inference

In order to perform inference on a set of arrival times $\boldsymbol{\tau}$, we solve the following optimization problem that incorporates our trained network

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \|G_\theta(\boldsymbol{c}) - \boldsymbol{\tau}\| + R(\boldsymbol{c}). \tag{7}$$

We solve this iteratively by fixing the weights of the network and computing the gradient with respect to its input $\boldsymbol{c}$. Note that this same optimization problem can be set up with the original forward model F, but computing a gradient is often non-trivial and computationally expensive. By training a forward model approximation, we can leverage existing auto-grad libraries [17] to efficiently compute the gradient.

This can be solved with a variety of optimization algorithms to converge on some locally optimal $\boldsymbol{c}$. Since we are substituting the forward model with an approximation, we can account for any inaccuracies by introducing a tolerance level. Once the observation loss drops below a pre-determined level, the optimization terminates early. Additionally, this optimization framework can be used to explicitly incorporate some prior knowledge by restricting our estimates to be in the span of some basis $\boldsymbol{\Phi}$. In this case, the optimization is performed in a much lower dimensional subspace of the coefficients by replacing Equation (7) with

$$\hat{\boldsymbol{c}} = \boldsymbol{\Phi}(\arg\min_{\boldsymbol{\alpha}} \|G_\theta(\boldsymbol{\Phi}\boldsymbol{\alpha}) - \boldsymbol{\tau}\|). \tag{8}$$

Note that we incorporated a regularizer $R(\cdot)$ as an additional cost term. The regularizer encourages certain properties (e.g. smaller values) and helps guide the optimization towards particular SSPs. In our experiments, we test ℓ-2 regularization as well as a co-variance (as determined by the an SVD decomposition of the training set) weighted regularization.

Finally, it should be noted that the iterative nature of this method requires that we initialize our guess with some estimate. When optimizing from scratch, a reasonable candidate would be the average SSP from the training set. Alternatively, we can leverage the estimated SSPs from other OAT inverse methods by first initializing with that estimate and then refining it with the NA procedure. We note that there is an increase in computation time compared to directly learning the inverse due to the iterative nature of the algorithm, but the NA method offers certain trade offs outlined above that might be more beneficial in practice than the single estimate provided by learned direct inverse methods.

4 Preliminary Results

For demonstrating the results of our proposed regularization, we will focus on range dependent SSPs. Collecting experimental SSPs for the range dependent case is cost prohibitive, especially at the scale required for machine learning. Instead, we turn to a high fidelity simulation of the Gulf of Mexico [10]. The physics were simulated using Navier-Stokes over the course of a month at a variety of depth resolutions as shown in Figure 1.

We focus on 2D vertical slices within the 3D domain of the data available. For a fixed slice, the amount of data is restricted by the time resolution of the simulation. The acoustic properties are heavily dependent on the bathymetry, making it difficult to train models agnostic to these conditions. To address this, we propose to focus only on the upper half of the ocean and only rely on the direct and top-bounce acoustic paths. We will deploy 20 equally distributed vertical arrays of sources and an equivalent vertical array of receivers 5km away. Bellhop (ray-tracing based) simulations are currently being run to collect arrival time information. A majority of the fluctuation occurs closer to the surface, making it the region of interest for recovery.

Some preliminary work has evaluated the ground truth SSP variation over the volume as seen in Figure 2. SSPs statistics are reduced over the entire spatial domain (minus depth) for any location deeper than 200m as well as time (all 30 days). These calculations are repeated across different depth resolutions, with more layers leading to more accurate physics simulation. Although the lower number of layers (30L, 70L) don't seem to quite capture the same variation as the higher numbers, they do confirm the trend that a majority of the variation occurs above 1000m, making it the region of interest for reconstruction. Note that these results do not use the arrival times from the above geometry yet and are purely for investigating properties of the SSP alone.

Acknowledgments and Disclosure of Funding

This project was supported by the Office of Naval Research Task Force Ocean under Grant No. N00014-19-1-2639.

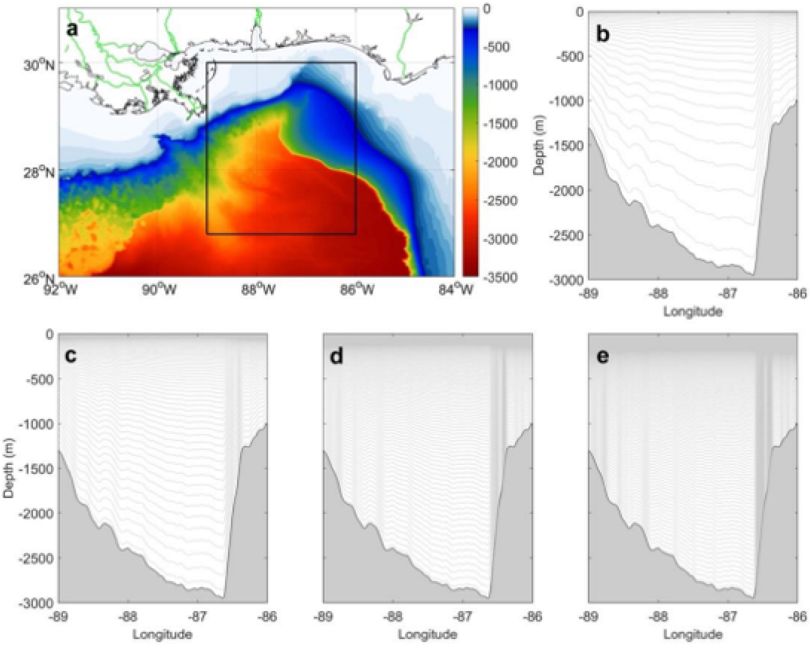

Figure 1: (a) Bathymetry in the northern Gulf of Mexico showing the modelled domain where CROCO is run at a 0.5 km horizontal resolution. (b-e) are the distribution of the 30, 70, 140, and 200 vertical sigma levels respectively along a east-west transect at 28°N across the nested domain. Provided by [10].

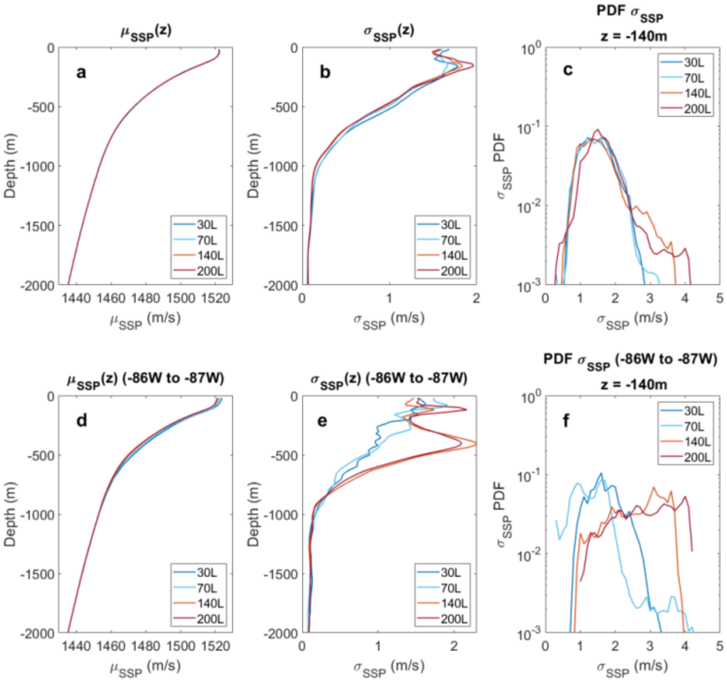

Figure 2: (top) The mean, standard deviation (STD) and probability density function of the STD calculated across the domain and time of sound speed profiles for the different depth resolution simulations calculated over all domains deeper than 200 m. (bottom) Same as top, but for the domain between 87°W and 86°W, where mesocale activity and density gradients are both elevated. Provided by [10]

References

[1] D Behringer, T Birdsall, M Brown, B Cornuelle, R Heinmiller, R Knox, K Metzger, W Munk, J Spiesberger, R Spindel, D Webb, P Worcester, and C Wunsch. A demonstration of ocean acoustic tomography. *Nature*, 299(5879):121–125, 1982.

[2] Matthew Chantry, Sam Hatfield, Peter Dueben, Inna Polichtchouk, and Tim Palmer. Machine learning emulation of gravity wave drag in numerical weather forecasting. *Journal of Advances in Modeling Earth Systems*, 13(7):e2021MS002477, 2021.

[3] Wojciech M Czarnecki, Simon Osindero, Max Jaderberg, Grzegorz Swirszcz, and Razvan Pascanu. Sobolev training for neural networks. *Advances in Neural Information Processing Systems*, 30, 2017.

[4] David R Dowling and Karim G Sabra. Acoustic remote sensing. *Annual Review of Fluid Mechanics*, 47:221–243, 2015.

[5] Nicholas C Durofchalk, Jihui Jin, Heriberto J Vazquez, Kay L Gemba, Justin Romberg, and Karim G Sabra. Data driven source localization using a library of nearby shipping sources of opportunity. *JASA Express Letters*, 1(12):124802, 2021.

[6] Sam Hatfield, Matthew Chantry, Peter Dueben, Philippe Lopez, Alan Geer, and Tim Palmer. Building tangent-linear and adjoint models for data assimilation with neural networks. *Journal of Advances in Modeling Earth Systems*, 13(9):e2021MS002521, 2021.

[7] Judy Hoffman, Daniel A Roberts, and Sho Yaida. Robust learning with jacobian regularization. *arXiv preprint arXiv:1908.02729*, 2019.

[8] Zhaoqiong Huang, Ji Xu, Zaixiao Gong, Haibin Wang, and Yonghong Yan. Source localization using deep neural networks in a shallow water environment. *The journal of the acoustical society of America*, 143(5):2922–2932, 2018.

[9] Finn B Jensen, William A Kuperman, Michael B Porter, Henrik Schmidt, and Alexandra Tolstoy. *Computational Ocean Acoustics*, volume 794. Springer, 2011.

[10] Guangpen Liu, Richard Touret, Annalisa Bracco, and Karim Sabra. The role of vertical resolution in the representation of the ocean sound speed in mesoscale eddies. *Draft*, 2022.

[11] Herman Medwin and Clarence S Clay. *Fundamentals of Acoustical Oceanography*. Elsevier, 1998.

[12] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[13] Ali Mousavi and Richard G Baraniuk. Learning to invert: Signal recovery via deep convolutional networks. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2272–2276. IEEE, 2017.

[14] Walter Munk and Carl Wunsch. Ocean acoustic tomography: A scheme for large scale monitoring. *Deep Sea Research Part A. Oceanographic Research Papers*, 26(2):123–161, 1979.

[15] Walter H Munk and Peter F Worcester. Ocean acoustic tomography. *Oceanography*, 1(1):8–10, 1988.

[16] Haiqiang Niu, Emma Ozanich, and Peter Gerstoft. Ship localization in santa barbara channel using machine learning classifiers. *The journal of the acoustical society of America*, 142(5):EL455–EL460, 2017.

[17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[18] Simiao Ren, Willie Padilla, and Jordan Malof. Benchmarking deep inverse models over time, and the neural-adjoint method. *Advances in Neural Information Processing Systems*, 33:38–48, 2020.

[19] EK Skarsoulis, GA Athanassoulis, and Uwe Send. Ocean acoustic tomography based on peak arrivals. *The Journal of the Acoustical Society of America*, 100(2):797–813, 1996.

[20] Bo Wang and Huafeng Liu. Fbp-net for direct reconstruction of dynamic pet images. *Physics in Medicine & Biology*, 65(23):235008, 2020.

[21] Yun Wang and Hua Peng. Underwater acoustic source localization using generalized regression neural network. *The Journal of the Acoustical Society of America*, 143(4):2321–2331, 2018.