

Report No. 48/2024

DOI: 10.4171/OWR/2024/48

## Deep Learning for PDE-based Inverse Problems

Organized by  
Simon Arridge, London  
Peter Maaß, Bremen  
Carola-Bibiane Schönlieb, Cambridge UK

27 October – 1 November 2024

**ABSTRACT.** Analysing learned concepts for PDE-based parameter identification problems requires input from different research areas such as inverse problems, partial differential equations, statistics and mathematical foundations of deep learning. This workshop brought together a critical mass of experts in the various field. A thorough mathematical theory for PDE-based inverse problems using learned concepts is within reach in the coming few years and the inspiration of this Oberwolfach meeting will substantially influence this development.

*Mathematics Subject Classification (2020):* 35R30, 65L09, 65Rxx, 44-XX.

*License:* Unless otherwise noted, the content of this report is licensed under CC BY SA 4.0.

### Introduction by the Organizers

This workshop was centered at the intersection of three research areas (inverse problems, partial differential equations, mathematical foundations of deep learning) with the aim of developing a solid mathematical theory for deep learning concepts for PDE-based inverse problems.

Such parameter identification problems for partial differential equation are commonly regarded as the toughest challenges in inverse problems. To be more precise, we may for example consider second order partial differential equations, which depend on a parameter function  $\lambda$ :

$$(1) \quad \mathcal{N}(u, \nabla u, \Delta u, u_t; \lambda) = 0$$

$$(2) \quad \lambda : \Omega \rightarrow \mathbb{R}$$

In this general notation  $\mathcal{N}$  encodes the differential equation as well as boundary conditions.

We always assume, that the parameter-to-state operator  $F$ , which maps a given parameter  $\lambda$  to the solution of the PDE, is well-posed. I.e. we assume a function space setting such that the solution  $u$  of the PDE is unique and depends continuously on the parameter  $\lambda$ .

We will consider the following hierarchy of problems:

- Forward problem solving a single PDE: given  $\lambda$ , compute  $u = F(\lambda)$
- Parametric studies: given many parameters  $\lambda \in X_N$ , compute corresponding  $u$ 's, i.e. evaluation of the parameter-to-state operator  $F$
- Parameter identification (inverse problem): given a measured  $u^\delta$  or its values  $Pu^\delta$  under a measurement operator  $P$ , determine corresponding  $\lambda$ , e.g. solve  $F(\lambda) \sim u^\delta$ , i.e. the inverse problem posed by the parameter-to-state operator.

The underlying PDE models, e.g. for climate modeling, molecular dynamics or large scale engineering models for digital twins, are getting more and more demanding requiring ever more computer power for their numerical simulation and solution.

This is already true for solving single PDE systems, but it is even more obvious for parametric studies, where several parameter settings need to be evaluated.

Classical numerical schemes such as finite elements are well established and efficient - but not to the extent needed for large scale parametric studies of complex PDEs. Hence, these methods definitely need to be reconsidered when attempting parameter identification problems, which typically are solved by generalized gradient descent iteration schemes requiring even more forward evaluations of the PDE.

Learned concepts are a natural choice for efficient computations in this contexts and they have been investigated intensively over the past few years. They predominantly either focus on deep learning for general inverse problems or on deep learning for fast PDE solvers and related parametric studies.

However, there is much less mathematical theory addressing deep learning concepts for PDE-based parameter identification problems (inverse problems). One reason is certainly the complexity of the problem: even if the PDE itself is linear, the related parameter identification problems and parameter-to-state operators are non-linear. More importantly, the inversion process is unstable or ill-posed and needs specific measures for regularization. Hence, well studied deep learning concepts for PDE forward solvers or other stable operator settings fail when applied to inverse problems. Such problems can only be solved in a combination of model driven expert knowledge of the underlying PDE in combination with specific network architectures.

**Contributions of this workshop.** The talks covered a broad spectrum of different aspects concerning learned methods for PDE-based parameter identification problems.

There were substantial contributions on the theoretical side, e.g. operator learning without the adjoint (Nicolas Boullé), convergence proofs for neural network solvers including discretization (Bangti Jin), linear algebra networks for linear inverse problems (Otmar Scherzer), novel theorems for operator approximation (Janek Gödeke), learning of regularizers (Giovanni Alberti), bilevel optimization (Juan Carlos De los Reyes), Lipschitz duality (Yury Korolev) or general approximation results for specific architectures (Chris Budd, Nick Heilenkötter, Sebastian Neumayer). In particular, estimates using Wasserstein metrics were discussed in at least two talks (Gabriele Steidl, Lisa Kreusser). There were several talks connecting mathematical foundations of deep learning with applications, such as analyzing the approximation properties, directly learning the inverse (Maarten de Hoop) of learning priors for Bayesian inverse problems in imaging (Thomas Pock) or using feature vectors (Felix Dietrich). In addition several talks focussed on novel concepts (Christina Runkel, Alexander Denker, Johannes Hertrich, Jianfeng Ning, Eldad Haber, Davide Murari) as well as challenging applications such as modeling turbulence, travel-time tomography or photo-acoustic tomography (Silvia Barbeiro, Bastian Harrach, Yolanne Yi Ran Lee, Hanno Gottschalk, Xiaoqun Zhang, Andreas Hauptmann, Tatiana Bubba).

Overall, this was a most stimulating workshop with plenty of ongoing discussions during the extended lunch breaks as well as long evenings using the most convenient facilities in Oberwolfach. In particular, this workshop has strengthened the paradigm, that truly novel developments in this field need a combination of statistical data analysis with classical analytical knowledge and motivation by real life applications.

Hence, we believe that a thorough mathematical theory for PDE-based inverse problems using learned concepts will emerge in the coming few years and that the inspiration of the Oberwolfach meeting will substantially influence future research in the field.

*Acknowledgement:* The MFO and the workshop organizers would like to thank the National Science Foundation for supporting the participation of junior researchers in the workshop by the grant DMS-2230648, “US Junior Oberwolfach Fellows”. Moreover, the MFO and the workshop organizers would like to thank the Oberwolfach Foundation for supporting the participation of junior researchers.



## Workshop: Deep Learning for PDE-based Inverse Problems

### Table of Contents

Rima Alaifari (joint with Francesca Bartolucci, Emmanuel de Bézenac, Bogdan Raonić, Roberto Molinaro, Siddhartha Mishra) <i>Representation equivalence for alias-free operator learning</i> .....	2813
Giovanni S. Alberti <i>Learning (simple) regularizers for inverse problems</i> .....	2815
Silvia Barbeiro <i>Mathematical models in elastography</i> .....	2817
Nicolas Boullé (joint with Diana Halikias, Samuel Otto, Alex Townsend) <i>Operator learning without the adjoint</i> .....	2820
Tatiana A. Bubba (joint with Subhadip Mukherjee, Luca Ratti, Andrea Sebastiani) <i>Learned proximal operators meets unrolling in imaging inverse problems</i>	2822
Chris Budd (joint with Simon Arridge, Teo Deveney, Lisa Kreusser) <i>Adaptivity, and expressivity in FKS and neural network approximations</i>	2825
Maarten V. de Hoop (joint with Nikola B. Kovachi, Matti Lassas, Nicholas H. Nelsen, Andrew M. Stuart) <i>Approximating the nonlinear inverse by neural operators</i> .....	2826
Juan Carlos De los Reyes <i>Bilevel learning for PDE inverse problems</i> .....	2827
Alexander Denker (joint with Željko Kereta, Simon Arridge) <i>Extending neural operator with Laplacian eigenfunctions for arbitrary domains</i> .....	2828
Felix Dietrich <i>Solving PDE-related inverse problems with random feature models</i> .....	2831
Janeke Gödeke (joint with Peter Maaß) <i>New universal operator approximation result for deep operator networks</i>	2835
Hanno Gottschalk (joint with Claudia Drygala, Francesca di Mare, Edmund Ross) <i>Generative modeling for turbulent flows</i> .....	2837
Eldad Haber <i>Multiscale methods for convolution neural networks</i> .....	2840

Bastian Harrach (joint with Andrej Brojatsch, Johannes Wagner) <i>Autoencoder-based global concave optimization for electrical impedance tomography</i> .....	2847
Andreas Hauptmann (joint with Simon Arridge, Anssi Manninen, Ozan Öktem, Carola-Bibiane Schönlieb) <i>Learned iterative reconstructions in photoacoustic tomography for the acoustic and optical problem</i> .....	2850
Nick Heilenkötter (joint with Clemens Arndt, Alexander Denker, Sören Dittmer, Meira Iske, Tobias Kluth, Peter Maaß, Judith Nickel) <i>Solving ill-posed inverse problems using iResNets</i> .....	2853
Johannes Hertrich (joint with Tim Jahn, Michael Quellmalz) <i>Fast kernel summation via slicing and Fourier transforms</i> .....	2855
Bangti Jin (joint with Zhi Zhou) <i>Conductivity imaging using deep neural networks</i> .....	2858
Yury Korolev (joint with Francesca Bartolucci, Marcello Carioni, José Iglesias, Emanuele Naldi, Stefano Vigogna) <i>Lipschitz duality in shallow neural networks</i> .....	2862
Lisa Kreusser <i>Unveiling the role of the Wasserstein distance in generative modelling</i> ..	2865
Yolanne Yi Ran Lee <i>Efficient diffusion models for MRI reconstruction</i> .....	2867
Davide Murari (joint with Priscilla Canizares, Carola-Bibiane Schönlieb, Ferdia Sherry, Zakhar Shumaylov) <i>Dynamical systems-based structured networks</i> .....	2869
Sebastian Neumayer <i>Spatially adaptive Ridge regularizers: Stability and performance</i> .....	2872
Jianfeng Ning (joint with Fuqun Han, Jun Zou) <i>Deep learning meets direct sampling methods for inverse scattering problems</i> .....	2874
Thomas Pock (joint with Muhamed Kuric) <i>Learning priors for Bayesian inverse problems in imaging</i> .....	2877
Christina Runkel (joint with Kanchana Vaishnavi Gandikota, Jonas Geiping, Carola-Bibiane Schönlieb, Michael Moeller) <i>Training data reconstruction</i> .....	2879
Otmar Scherzer (joint with Andrea Aspri) <i>Numerical linear algebra networks for solving linear inverse problems</i> ..	2881
Zakhar Shumaylov (joint with P. Zaika, J. Rowbottom, F. Sherry, M. Weber, C. B. Schönlieb) <i>Lie Algebra Canonicalization: Equivariant Neural Operators under arbitrary Lie Groups</i> .....	2884

---

Gabriele Steidl (joint with Jannis Chemseddine, Christian Wald, Richard Duong)	
<i>Neural sampling from Boltzmann densities: Fisher-Rao curves in the</i>	
<i>Wasserstein geometry</i> .....	2887
Xiaoqun Zhang (joint with Xiang Cao)	
<i>Subspace diffusion posterior sampling for travel-time tomography</i> .....	2890





## Abstracts

### Representation equivalence for alias-free operator learning

RIMA ALAIFARI

(joint work with Francesca Bartolucci, Emmanuel de Bézenac, Bogdan Raonić, Roberto Molinaro, Siddhartha Mishra)

In recent years, neural operators have become a popular tool aimed at learning a map between *infinite-dimensional* spaces. They consist of layers of linear integral operators followed by non-linear activation functions. Since the actual learning uses data, it is carried out in finite dimensions and requires a link to the infinite-dimensional setting. One way to establish this link is through *discretization invariance* [3] as in Fourier Neural Operators for example. It ensures that the discretized maps tend to an operator mapping between infinite dimensions as the discretization becomes finer.

In this work, we observe that discretization invariance is an *asymptotic property* and provide numerical evidence of significant discrepancies (through *aliasing errors*) when such architectures are tested at resolutions that differ from the training resolution. We develop a novel framework of *Representation equivalent Neural Operators (ReNOs)* [1], that rather guarantees a *continuous-discrete equivalence*. Our construction is based on utilizing *frame theory* [2] and generalizing aliasing errors to operators. This way, unique and stable reconstruction of the infinite-dimensional operator from its discretizations can be guaranteed. By connecting *all* discretizations to the same operator between infinite dimensions, any two discretizations are also tied together uniquely and stably. Numerical experiments further highlight that there is no trade-off between the expressivity of the network and its ReNO property. As a novel ReNO architecture we propose *Convolutional Neural Operators* [4], that achieve state-of-the-art results on a large set of benchmark PDEs.

**Alias-Free Framework for Operator Learning.** Let  $U$  be an operator between separable Hilbert spaces  $\mathcal{H}$  and  $\mathcal{K}$ , and let  $\Psi = \{\psi_i\}_{i \in I}$  and  $\Phi = \{\varphi_j\}_{j \in J}$  be *frames* [2] for  $\mathcal{M}_\Psi := \text{span}\{\psi_i : i \in I\} \subseteq \mathcal{H}$  and  $\mathcal{M}_\Phi := \text{span}\{\varphi_j : j \in J\} \subseteq \mathcal{K}$ , respectively. The synthesis operators  $T_\Psi$  and  $T_\Phi$  map sequences of frame coefficients back to functions in the respective Hilbert space, i.e.  $T_\Psi(\{c_i : i \in I\}) = \sum_{i \in I} c_i \psi_i$  and  $T_\Phi(\{c_j : j \in J\}) = \sum_{j \in J} c_j \varphi_j$ . Hence, out of a discrete realization  $u : \ell^2(I) \rightarrow \ell^2(J)$  of the operator  $U$  one can construct an operator mapping from  $\mathcal{H}$  to  $\mathcal{K}$  through  $T_\Phi \circ u \circ T_\Psi^\dagger : \mathcal{H} \rightarrow \mathcal{K}$ , where  $T_\Psi^\dagger$  denotes the pseudoinverse of  $T_\Psi$ . We can now introduce the *operator aliasing error*  $\epsilon(U, u, \Psi, \Phi)$

$$\epsilon(U, u, \Psi, \Phi) := U - T_\Phi \circ u \circ T_\Psi^\dagger.$$

It measures the discrepancy between  $U$  and its discrete implementation  $u$ , with the corresponding scalar error given by the operator norm  $\|\epsilon(U, u, \Psi, \Phi)\|$ . If this

error is zero, the operator  $U$  can be perfectly represented by the discrete map  $u$ , ensuring continuous-discrete equivalence (CDE).

The concept of *Representation equivalent Neural Operators (ReNOs)* requires an extension of the above. On the level of discretizations, we will consider  $u$  more broadly as a map that takes in any pair of frames  $(\Psi, \Phi)$  for any subspaces  $\mathcal{M}_\Psi \subseteq \mathcal{H}, \mathcal{M}_\Phi \subseteq \mathcal{K}$  and outputs a mapping from  $\ell^2(I)$  to  $\ell^2(J)$ :

$$u(\Psi, \Phi) : \ell^2(I) \rightarrow \ell^2(J).$$

In other words,  $u$  *must change* with the choice of discrete representations  $\Psi$  and  $\Phi$  in order to eliminate aliasing errors. We then say that  $(U, u)$  is a ReNO if, for any pair  $(\Psi, \Phi)$  with

$$(1) \quad \text{Dom } U \subseteq \mathcal{M}_\Psi \text{ and } \text{Ran } U \subseteq \mathcal{M}_\Phi$$

there is no aliasing error, i.e.  $\epsilon(U, u, \Psi, \Phi) = 0$ . Condition (1) simply ensures that only those frame sequences are considered, that can actually represent the domain and range of  $U$ , respectively. This definition can be straightforwardly extended to a layerwise instantiation, by ensuring zero aliasing error at every layer. As a consequence of the ReNO condition, any two discrete representations of the operator are equivalent (cf. Fig. 1), preserving the underlying structure in function spaces. We note that one can also consider  $\epsilon$ -ReNOs, where a small, controlled amount of aliasing is permissible, i.e.  $\|\epsilon(U, u, \Psi, \Phi)\| \leq \epsilon$ , for all  $\Psi$  and  $\Phi$  satisfying (1).

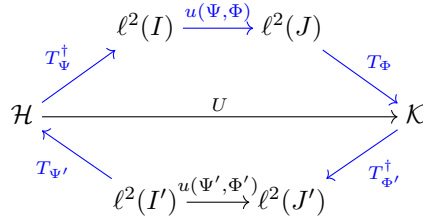


FIGURE 1. Property of ReNOs: discrete representations for different discretizations are equivalent.

Classical Convolutional Neural Networks (CNNs) and Fourier Neural Operators (FNOs) do not satisfy the ReNO condition due to inconsistencies across different discretizations. The discrete convolution operation in CNNs does not preserve the continuous-discrete equivalence. For FNO, the issue lies in the nonlinear activation function: the considered function space is that of bandlimited functions throughout the network. When a pointwise nonlinearity  $\sigma$ , such as the ReLU or GeLU is applied to a function  $f$ , then in general,  $\sigma(f)$  is no longer bandlimited and has an effective bandwidth significantly larger than that of  $f$ .

In contrast, Convolutional Neural Operators (CNOs) take this into account by upsampling the function before applying a nonlinear activation function (followed by a downsampling operation to control the dimensionality throughout the

network). CNOs take the form of a U-Net architecture and convolutions are implemented in the original space without passing to the Fourier domain (see [4] for details and tests on a wide range of benchmark PDEs).

**Outlook.** An interesting future direction is to develop and analyze quantitative measures of error in ReNOs. One source of error lies in extending the concept to  $\varepsilon$ -ReNOs, providing a nuanced approach to operator learning with controlled aliasing. Another error source is given by the leap from discrete to finite representations. If  $\text{Dom } U$  and  $\text{Ran } U$  are infinite-dimensional, the index sets  $I$  and  $J$  will be infinite as well. By using finite approximations in practice, an *approximation error* is introduced. Controlled error bounds are an important aspect moving forward.

Another relevant question is that of identifying combinations of suitable function spaces and activation functions so that the space is closed under the action of the activation function. More generally, identifying pairs of function spaces  $B_1, B_2$  and activation functions  $\sigma$  for which  $\sigma(B_1) \subseteq B_2$  could be helpful in designing new architectures that satisfy the ReNO property.

#### REFERENCES

- [1] Bartolucci F., de Bézenac E., Raonić B., Molinaro R., Mishra S., & Alaifari R. *Representation equivalent neural operators: a framework for alias-free operator learning*, Advances in Neural Information Processing Systems, 37, 2023.
- [2] Christensen O. *An Introduction to Frames and Riesz Bases*, 2003.
- [3] Kovachki N., Li Z., Liu B., Azizzadenesheli K., Bhattacharya K., Stuart A., & Anandkumar A. *Neural operator: Learning maps between function spaces with applications to PDEs*, Journal of Machine Learning Research, 24(89), 1-97, 2023.
- [4] Raonić B., Molinaro R., De Ryck T., Rohner T., Bartolucci F., Alaifari R., Mishra S., & de Bézenac E. *Convolutional neural operators for robust and accurate learning of PDEs*, Advances in Neural Information Processing Systems, 37, 2023.

### Learning (simple) regularizers for inverse problems

GIOVANNI S. ALBERTI

**Inverse problems.** We consider a linear inverse problem of the form

$$y = Ax + \varepsilon,$$

where  $A: X \rightarrow Y$  is a bounded linear map between the separable Hilbert spaces  $X$  and  $Y$ , so-called forward map,  $\varepsilon$  represents noise,  $y$  are the noisy measurements and  $x$  is the unknown to be recovered. In a well-established Bayesian setting, both  $x$  and  $\varepsilon$  are treated as random variables [5]. The ill-posedness of the problem comes from the fact that  $A^{-1}$  is typically an unbounded operator. Classical examples of inverse problems of this form are deconvolution, for which  $A$  is a convolution operator, and computed tomography, for which  $A$  is the Radon transform. These problems are intrinsically infinite-dimensional, which motivates the need to use function spaces for modeling.

We consider classical regularization techniques, in which the recovery of  $x$  is performed by minimizing a functional of the form

$$\min_{x \in X} d(Ax, y) + J(x),$$

where  $d(Ax, y)$  is the data fidelity term, and depends on the statistics of the noise. Here, the focus is on the regularization term  $J: X \rightarrow \mathbb{R}$ , which should encode prior information available on the solution  $x$ . We consider the problem of learning the optimal  $J$ , within a fixed family of regularizers, from training data [4].

**Learning the optimal Tikhonov regularizer.** In [1], the focus is on generalized Tikhonov regularization, namely,

$$J(x) = \|B^{-1}(x - h)\|_X^2,$$

where  $h \in X$  is a reference signal and  $B: X \rightarrow X$  is positive and bounded. Intuitively,  $B^{-1}$  can be regarded as a generalization of a differential operator. Under suitable assumptions, it is possible to prove that the optimal regularization parameters, namely those minimizing the mean squared error of the reconstruction, are given by  $h^* = \mu_x$  and  $B^* = (\Sigma_x)^{\frac{1}{2}}$ , where  $\mu_x$  and  $\Sigma_x$  are the mean and the covariance of the random variable  $x$ , respectively. Furthermore, generalization estimates are derived, for both a supervised learning approach based on empirical risk minimization, and for an unsupervised learning approach, based on the approximation of  $\mu_x$  and  $\Sigma_x$  by using the corresponding empirical counterparts.

**Learning the optimal  $\ell^1$  regularizer.** The work [2] extends the results of [1] by replacing the generalized Tikhonov term with a sparsity promoting regularizer of the form

$$J(x) = \|B^{-1}x\|_{\ell^1},$$

where  $B: \ell^2 \rightarrow X$  can be interpreted as a change of basis. The  $\ell^1$  norm is used to promote sparsity. In this case, it is not possible to give a closed form expression of the optimal regularizer. Nevertheless, by using the Hölder continuity of the minimizer as a function of  $B$ , a result of independent interest, we are able to prove generalization bounds under general assumptions on the class of operators  $B$ .

**Sparse regularization via Gaussian mixtures.** In the work [3], we propose a probabilistic sparsity prior formulated as a mixture of degenerate Gaussians, capable of modeling sparsity with respect to a generic basis. Under this premise, we design a neural network that can be interpreted as the Bayes estimator for linear inverse problems. Additionally, we propose both supervised and unsupervised training strategies to estimate the parameters of this network. To evaluate the effectiveness of our approach, we conduct a numerical comparison with commonly employed sparsity-promoting regularization techniques, namely LASSO, group LASSO, iterative hard thresholding, and sparse coding/dictionary learning. Notably, our reconstructions consistently exhibit lower mean square error values across all 1D datasets used in the comparisons, even in cases where the datasets significantly deviate from a Gaussian mixture model.

## REFERENCES

- [1] G. S. Alberti, E. De Vito, M. Lassas, L. Ratti & M. Santacesaria, *Learning the optimal Tikhonov regularizer for inverse problems*, Advances in Neural Information Processing Systems, 34, 25205–25216, 2021.
- [2] G. S. Alberti, E. De Vito, T. Helin, M. Lassas, L. Ratti & M. Santacesaria, *Learning sparsity-promoting regularizers for linear inverse problems*, 2024.
- [3] G. S. Alberti, L. Ratti, M. Santacesaria & S. Sciutto, *Learning a Gaussian Mixture for Sparsity Regularization in Inverse Problems*, arXiv:2401.16612, 2024.
- [4] S. Arridge, P. Maaß, O. Öktem & C. B. Schönlieb, *Solving inverse problems using data-driven models*, Acta Numerica 28, 1–174, 2019.
- [5] A. M. Stuart, *Inverse problems: a Bayesian perspective*, Acta Numerica 19, 451–559, 2010

**Mathematical models in elastography**

SILVIA BARBEIRO

## 1. INTRODUCTION

Assessing the mechanical properties of biological tissue provides valuable insights for disease diagnosis and monitoring, as tissue stiffness has long been recognized as a key biomarker for certain diseases.

Optical coherence elastography (OCE) is an innovative biomedical imaging technique that uses optical coherence tomography (OCT) to form pictures of biological tissue in order to assess its biomechanical properties. When a mechanical load is applied to the tissue, it causes a deformation response. There are various implementations of OCE, differing in the type of mechanical loading applied to the tissue, the OCT scanning protocol, and the method used to measure tissue deformation, while all approaches rely on detecting tissue deformation through sequential OCT scans.

Mathematical modelling in elastography comprises two main steps. The first step focuses on using OCT images for the reconstruction of the displacement field in order to estimate the mechanical response of tissue under stress. The second step utilizes this mechanical response estimation to reconstruct the tissue’s internal mechanical properties.

The main challenges in this line of research involve the development of efficient methods, capable to deal with noisy data, to recover the mechanical properties that define the elastic medium, that is to solve the elastography inverse problem. Nevertheless, the development of robust methods for computing tissue displacements in optical coherence elastography (OCE) is also crucial, as they significantly influence the accuracy of estimating tissue elastic properties.

## 2. COMPUTING THE DISPLACEMENT FIELD

There are two predominant techniques that can be used to obtain tissue displacements, which are described in [10]: speckle tracking and phase-resolved detection. Speckle-tracking methods rely on intensity information and, as a result, generally require large displacements. In contrast, phase-resolved measurements utilize

phase difference information, offering nanoscale displacement sensitivity. In phase-resolved measurements considering a laser source, the phase difference  $\Delta\varphi$  is used to estimate the tissue displacement  $u_z$  [10]:

$$u_z = \Delta\varphi \frac{\lambda_0}{4\pi n},$$

where  $\lambda_0$  is the center wavelength of the laser and  $n$  is the refractive index of the sample.

One can find various methods for estimating  $\Delta\varphi$  in the literature, each offering different levels of accuracy and sensitivity to noise. The recent work [5] proposes a methodology that combines different phase-difference estimates such that the error can be minimized.

### 3. INVERSE ELASTOGRAPHY PROBLEM

Now, we focus on the numerical solution of the inverse elastography problem, which consists of determining the set of parameters that characterize the mechanical properties of the medium utilizing the knowledge of the excitation field at the boundary and the displacement field at a grid of points within the domain.

Under the assumption of time-harmonic excitations in the linear elasticity model, the displacement field exhibits a time-harmonic form [4, 6]. In this case, the following Lamé equation governs the elastic displacement field  $\mathbf{u}$  in a domain  $\Omega$

$$\mu\Delta\mathbf{u} + (\lambda + \mu)\nabla\nabla\cdot\mathbf{u} + \omega^2\rho\mathbf{u} = 0 \text{ in } \Omega,$$

where  $\rho$  is the material density,  $\omega$  is the frequency and the Lamé constants,  $\mu$  and  $\lambda$ , are related to Young's Modulus,  $E$ , and Poisson's ratio,  $\nu$ , by

$$\mu = \frac{E}{2(1+\nu)} \text{ and } \lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}.$$

To reflect the mechanical loading over the boundary, we consider a traction boundary condition.

In a more general setting, the Navier-Cauchy equation that governs the dynamic response of an isotropic, homogenous, Kelvin-Voigt viscoelastic material with small displacement assumption can be considered [8]:

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = \mu\Delta\mathbf{u} + (\lambda + \mu)\nabla\nabla\cdot\mathbf{u} + \eta\Delta\frac{\partial \mathbf{u}}{\partial t} + \eta\nabla\nabla\cdot\frac{\partial \mathbf{u}}{\partial t}$$

in  $\Omega \times \mathbb{R}^+$ , where  $\eta$  is the shear viscosity. Heterogeneity and anisotropy can also be introduced in the above models, which enable the simulation of complex materials like biological tissues or composites.

Different methodologies can be used to obtain the numerical solution of the above models, namely the finite element method [3, 9]. In [4] the method of fundamental solutions was proposed for the numerical simulation of the mechanical waves propagation and induced displacements in the human retina, assuming time-harmonic excitations.

There are different approaches in the literature for the inverse elastography problem in what concerns the definition of the function to be optimized. Some

methods are based on minimizing the difference between the measured and simulated displacements [6, 11, 12]. In [7], the cost function represents not only measures of the error in the displacement but also includes measures of the error in the stress fields, the momentum balance, and the constitutive law. Differently, in [2], the parameters that characterize the mechanical properties of the medium are computed such that the given data solves the direct problem, with the advantage that the objective function is convex and consequently the optimal solution can be computed with a single iteration.

Deep learning methods for solving inverse problems have been intensively developed [1], demonstrating remarkable performance across various applications. For the inverse elastography problem, approaches using neural networks has shown promising results. We refer to [7], where physics-informed neural networks are proposed to infer the Lamé parameters  $\lambda$  and  $\mu$  of a linear elastostatic model. Recently, in [9], neural networks are designed to characterize the spatial varying mechanical properties of the medium in a elastodynamic model.

#### ACKNOWLEDGEMENTS

This work was partially supported by the Centre for Mathematics of the University of Coimbra (funded by the Portuguese Government through FCT/MCTES, DOI 10.54499/UIDB/00324/2020).

#### REFERENCES

- [1] Arridge S., Maaß, P., Öktem O., Schönlieb C-B. (2019). Solving inverse problems using data-driven models. *Acta Numerica.*, 28:1-174.
- [2] Barbeiro S., Henriques R., Santos J.L. (2024). A quadratic optimization program for the inverse elastography problem. *J.Math.Industry* 14, 18.
- [3] Barbeiro S, Henriques R., Santos J.L.. The derivative free trust-region method for the inverse elastography problem. To appear in *Proceedings of the 22nd ECMI Conference on Industrial and Applied Mathematics*, Springer, 2023.
- [4] Barbeiro S., Serranho P. (2020). The method of fundamental solutions for the direct elastography problem in the human retina. *Proceedings of the 9th conference on Trefftz methods and 5th conference on method of fundamental solutions*. Berlin: Springer.
- [5] Batista A., Serranho P., Santos M. J., Correia C., Domingues J. P., Loureiro C., Cardoso J., Barbeiro S., Morgado M., Bernardes R. (2023). Phase-Resolved Optical Coherence Elastography: An Insight into Tissue Displacement Estimation. *Sensors*, 23(8), 3974.
- [6] Doyley M. M. (2012). Model-based elastography: a survey of approaches to the inverse elasticity problem. *Physics in Medicine and Biology*, 57(3):R35-R73, 2012.
- [7] Haghighat E., Raissi M., Moure A., Gomez H., Juanes R. (2021). A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics, *Computer Methods in Applied Mechanics and Engineering*, 379, 113741.
- [8] Han Z., Li J., Singh M., Wu C., Liu C., Raghunathan R., Aglyamov S.R., Vantipalli S. , Twa M. D., Larin K.V. (2017). Optical coherence elastography assessment of corneal viscoelasticity with a modified Rayleigh-Lamb wave model, *Journal of the Mechanical Behavior of Biomedical Materials*, 66, 87-94.
- [9] Henriques R., Barbeiro S. (2024). Finite elements with neural networks for the inverse elastography problem. *DMUC Preprint* 24-43.
- [10] Kennedy B.F., Kennedy K.M., Sampson D.D. (2014). A Review of Optical Coherence Elastography: Fundamentals, Techniques and Prospects. *IEEE J. Sel. Top. Quantum Electron.*, 20, 272-288.

- [11] Park E., Maniatty A. M. (2006). Shear modulus reconstruction in dynamic elastography: time harmonic case. *Phys Med Biol.*, 51(15):3697-721.
- [12] Serranho P., Barbeiro S., Henriques R., Batista A., Santos M., Correia C., Domingues J., Loureiro C., Cardoso J., Bernardes R., Morgado M., (2022). On the numerical solution of the inverse elastography problem for time-harmonic excitation. In: *Proceedings of the 2nd international conference on image processing and vision engineering (IMPROVE 2022)*, 259-64.

## Operator learning without the adjoint

NICOLAS BOULLÉ

(joint work with Diana Halikias, Samuel Otto, Alex Townsend)

There is a mystery at the heart of operator learning: how can one recover a non-self-adjoint operator from data without probing its adjoint? Current practical approaches suggest that one can accurately recover an operator while only using data generated by the forward action of the operator without access to the adjoint [5]. However, naively, it seems essential to sample the action of the adjoint for learning solution operator of time-dependent partial differential equations (PDEs) [3]. This motivates a fundamental question in numerical linear algebra: can one approximate a non-symmetric low-rank matrix without sketching its adjoint?

In this work, we explore the limits of adjoint-free low-rank matrix recovery and propose an approach that could help analyze the behavior of structured matrix recovery algorithms. Then, we show that one can approximate a family of non-self-adjoint infinite-dimensional compact operators via projection onto a Fourier basis without querying the adjoint. We apply the result to recover Green's functions of elliptic partial differential operators and derive an adjoint-free sample complexity bound. While existing infinite-dimensional numerical linear algebra theory justifies low sample complexity in operator learning [2, 4], ours is the first adjoint-free analysis that attempts to close the gap between theory and practice [1].

**Limits of adjoint-free low-rank matrix recovery.** We start in the fundamental setting of recovering a low-rank matrix by querying the map  $x \mapsto Fx$  but without access to  $x \mapsto F^*x$ . We show that querying  $x \mapsto F^*x$  is essential for recovering  $F$  and prove rigorous guarantees on the quality of the reconstruction in terms of how close  $F$  is to a symmetric matrix. Thus, we conclude that without prior knowledge of the properties of the adjoint, one must have access to its action.

We assume that  $F$  is  $\delta$ -near-symmetric (i.e., its left and right singular subspaces are  $\delta$ -close), but we only have access to partial information regarding the symmetry of  $F$ , namely that  $F$  is  $\epsilon$ -near-symmetric for some  $\epsilon \geq \delta$ , and sketching constraint  $FX$ . To quantify the resulting uncertainty about  $F$ , we define the set of possible matrices one could recover given this prior knowledge as

$$\Omega_{F,X}^\epsilon = \{A \in M_n(\mathbb{C}) : \text{rank}(A) = k, AX = FX, \exists Q \in O(k), \|U_A^* V_A - Q\|_2 \leq \epsilon\},$$

where  $A = U_A S_A V_A^*$  is the singular value decomposition of  $A$ ,  $O(k)$  is the group of  $k \times k$  orthogonal matrices, and  $\|\cdot\|_2$  denotes the spectral norm. Hence, given some



tolerance  $\epsilon$ ,  $\Omega_{F,X}^\epsilon$  is the set of  $\epsilon$ -near-symmetric matrices that can be returned by any low-rank recovery algorithm when approximating  $F$ , such as the randomized SVD [6, 7] or the Nyström method [8].

The size of  $\Omega_{F,X}^\epsilon$  is measured by its diameter in the spectral norm and determines the maximum accuracy of any reasonable reconstruction. If the diameter is large, one cannot estimate  $F$  accurately, as one cannot distinguish between any candidate matrix in  $\Omega_{F,X}^\epsilon$ . This is because any matrix in  $\Omega_{F,X}^\epsilon$  satisfies the sketching constraint and is near-symmetric. On the other hand, a small diameter guarantees the fidelity of the reconstruction. We provide sharp upper and lower bounds on the size of  $\Omega_{F,X}^\epsilon$ , i.e., determine how far apart any two matrices in  $\Omega_{F,X}^\epsilon$  can be from each other, with respect to  $\epsilon$ , which measures our prior knowledge of  $F$ 's symmetry. The upper and lower bounds on the diameter of  $\Omega_{F,X}^\epsilon$  reveal that the uncertainty about  $F$  given queries of its action is directly related to the uncertainty about the symmetry of its left and right singular subspaces. For example, our ability to recover a symmetric rank- $k$  matrix using  $k \leq s < n$  queries is fundamentally limited by our prior knowledge about the proximity of  $\text{Range}(F)$  and  $\text{Range}(F^*)$  because there are many asymmetric matrices with the same rank that satisfy the same sketching constraints. This result is a fundamental limitation of adjoint-free low-rank matrix recovery in numerical linear algebra and has implications for operator learning.

**An adjoint-free operator learning approach.** To provide an operator learning approach that does not need access to the adjoint, we exploit regularity results from PDE theory to estimate the range of the adjoint of the solution operator. This allows us to prove the first guarantees on the accuracy of adjoint-free approximations. Our key insight is to leverage the favorable properties of a prior self-adjoint operator, such as the Laplace–Beltrami operator, to use as an operator preconditioner in the approximation problem. In particular, we query the action of the solution operator on the eigenfunctions of the prior self-adjoint operator, yielding an approximation with an error that decays at a rate determined by the eigenvalues of the prior. This is remarkable because common operator learning techniques always seem to plateau; yet, we construct a simple algorithm that provably converges.

**The effect of non-normality on sample complexity.** We derive a sample complexity bound for our algorithm when applied to second-order uniformly-elliptic PDEs that are perturbed away from self-adjointness by lower-order terms. We show that for small perturbations, our bound on the approximation error grows linearly with the size of the perturbation, and we conjecture that this linear growth continues for large perturbations as well. This aspect of the error growth is also present in common operator learning techniques, as our numerical experiments illustrate. With respect to our operator learning algorithm, this means that the number of samples required to achieve a fixed error tolerance grows algebraically with the perturbation size.

## REFERENCES

- [1] N. Boullé, D. Halikias, S. E. Otto, and A. Townsend, *Operator learning without the adjoint*, to appear in J. Mach. Learn. Res. (2024).
- [2] N. Boullé, D. Halikias, and A. Townsend, *Elliptic PDE learning is provably data-efficient*, Proc. Natl. Acad. Sci. USA, 120 (2023), p. e2303904120.
- [3] N. Boullé, S. Kim, T. Shi, and A. Townsend, *Learning Green's functions associated with time-dependent partial differential equations*, J. Mach. Learn. Res., 23 (2022), pp. 1–34.
- [4] N. Boullé and A. Townsend, *Learning elliptic partial differential equations with randomized linear algebra*, Found. Comput. Math., 23 (2023), pp. 709–739.
- [5] N. Boullé and A. Townsend, *A mathematical guide to operator learning*, in Handbook of Numerical Analysis, vol. 25, Elsevier, 2024, ch. 3, pp. 83–125.
- [6] N. Halko, P.-G. Martinsson, and J.-A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288.
- [7] P.-G. Martinsson and J.A. Tropp, *Randomized numerical linear algebra: Foundations and algorithms*, Acta Numer., 29 (2020), pp. 403–572.
- [8] E. J. Nyström, *Über die praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben*, Acta Math., 54 (1930), pp. 185–204.

## Learned proximal operators meets unrolling in imaging inverse problems

TATIANA A. BUBBA

(joint work with Subhadi Mukherjee, Luca Ratti, Andrea Sebastiani)

Consider the linear inverse problem:

$$(1) \quad m = Au^\dagger + \epsilon$$

where  $m \in \mathbb{R}^d$  are the measurements,  $A : \mathbb{R}^n \rightarrow \mathbb{R}^d$  is the forward operator,  $\epsilon \in \mathbb{R}^d$  such that  $\|\epsilon\| \leq \delta$  is the noise and  $u^\dagger \in \mathbb{R}^n$  is an image to be recovered (from, e.g., tomographic data [8]). The Maximum a Posteriori (MAP) formulation allows to find an approximate solution of (1) by seeking a solution of the minimization problem (derived under the assumption  $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ ):

$$(2) \quad \arg \min_u \left\{ \frac{\lambda}{2} \|Au - m\|_2^2 + \phi(u) \right\},$$

where  $\lambda > 0$ . The role of  $\phi$  is that of a regularizer, i.e., it should address ill-posedness of (1), enforcing a priori regularity of the solution, and should in general be linked to the underlying image probability distribution, as well as maintaining the mathematical tractability of the functional in (2) to aid the application of efficient numerical optimization schemes. A prominent example for the class of imaging inverse problems is that of sparsity-promoting regularization solved via the Iterative Soft-Thresholding Algorithm (ISTA) [5].

In fact, ISTA is a special instance of the Proximal Gradient Descent (PGD) method which, starting from the initial guess  $u^{(0)} \in \mathbb{R}^n$ , computes a solution of (2) by forming the sequence  $(u^{(k)})_{k \in \mathbb{N}}$  defined by:

$$(3) \quad u^{(k+1)} = \text{prox}_{\gamma\phi}(u^{(k)} - \gamma \nabla f(u^{(k)})),$$

where  $f(u) = \frac{\lambda}{2}\|Au - m\|_2^2$ ,  $\gamma \in \mathbb{R}$  is a step size and  $\text{prox}_{\psi}(\cdot)$  denotes the proximity operator associated with a (weakly) convex function  $\psi$ . A convergence result, along with a linear convergence rate, of the sequence in (3) can be found in, e.g., [2]. The convergence rate can be improved by considering an extrapolated version of PGD:

$$(4) \quad \begin{cases} u^{(k+1)} &= \text{prox}_{\gamma\phi}(z^{(k)} - \gamma\nabla f(z^{(k)})), \\ z^{(k+1)} &= u^{(k+1)} + \alpha_{k+1}(u^{(k+1)} - u^{(k)}), \end{cases}$$

where  $(\alpha_k)_{k \in \mathbb{N}}$  forms the sequence of extrapolation parameters, which is crucial to determine the (accelerated) convergence properties of the extrapolated scheme [2].

In recent years, model-based strategies for solving ill-posed inverse problems, such as (2)-(3) or (2)-(4), have been successfully integrated with data-driven approaches, providing satisfying numerical results and insights into major theoretical and practical questions [1, 3]. A notable example, among the self-supervised techniques, is that of Plug-and-Play (PnP) [10], which can be deployed by replacing the proximity operator in algorithms like (3) or (4) with a denoiser  $D_\sigma$ :

$$(5) \quad u^{(k+1)} = D_\sigma(u^{(k)} - \gamma\nabla f(u^{(k)})),$$

where  $D_\sigma$  is separately trained using pairs  $(u_i, u_i + \eta_i)$  of clear and noisy images, with  $\eta_i \sim \mathcal{N}(0, \sigma^2 I)$ . Under suitable assumptions on the denoiser (e.g., Gradient Step denoiser), it is possible to guarantee that the update in (5) implicitly defines a variational regularization strategy (that is,  $D_\sigma$  is the proximity operator of a weakly convex  $\phi_\sigma$ ), and ensure its (linear) convergence [7].

Another paradigm for integrating model and data-based techniques is provided by algorithm unrolling (or *deep unfolding*), a supervised learning approach consisting of unfolding the successive iterations of an iterative scheme, such as PGD or ISTA [6]. This idea has been widely explored in the context of inverse problems in imaging. For instance, in [4] the authors proposed a novel convolutional neural network, called  $\Psi$ DONet, designed starting from the unrolled iterations of ISTA by adding a learnable “perturbation” which acts as an operator correction:

$$(6) \quad u^{(k+1)} = \text{prox}_{\gamma_k\phi}(u^{(k)} - \gamma_k\nabla f(u^{(k)}) - \beta_k\Lambda_{\zeta_k}u^{(k)}),$$

where the  $\zeta_k$ ’s are the CNN filters components for each layer  $k$ , which are learned end-to-end together with the parameters  $\beta_k$  and  $\gamma_k$ . In [4],  $\Psi$ DONet was applied to the inverse problem of limited angle tomography showing that the CNN structure can be informed by the convolutional nature of the X-ray transform [9], thus inducing a “microlocal regularization” that smooths away the singularities characterizing this imaging problem. The convergence analysis of the sequence in (6) is also provided in [4], relying on classical results on the convergence theory of ISTA.

In this talk, some recent results on combining the Gradient-Step (GS) denoiser formulation of a Plug-and-Play scheme with the unrolling framework of  $\Psi$ DONet are presented. Particular effort is put into the efficient formulation of the algorithm, by introducing an extrapolation strategy in the unrolled scheme which allows to reduce the computational resources necessary to compute the reconstruction while preserving the theoretical guarantees. In detail, the first part of the talk

is devoted to the convergence analysis of the following extrapolated PnP-PGD update:

$$(7) \quad \begin{cases} u^{(k+1)} &= D_\sigma(z^{(k)} - \gamma \nabla f(z^{(k)})), \\ z^{(k+1)} &= u^{(k+1)} + \alpha_{k+1}(u^{(k+1)} - u^{(k)}). \end{cases}$$

Linear convergence of the sequence in (7) to a stationary point of  $F_\sigma(u) := f(u) + \phi_\sigma(u)$  is guaranteed under suitable assumptions on the denoiser: in particular, it requires  $D_\sigma = I - \nabla g_\sigma$  (with  $g_\sigma(u) = \frac{1}{2}\|u - N_\sigma(u)\|^2$  using a neural network  $N_\sigma$ ) to be a GS denoiser and to ensure that the Lipschitz constant  $L_{g_\sigma}$  of  $\nabla g_\sigma$  is bounded above by 1. These requirements are in line with the related literature on PnP extrapolated schemes (cf., e.g., [7]). Guaranteeing  $L_\sigma < 1$  is not simply a theoretical requirement but turns out to be essential especially in the context of imaging problems with incomplete data, such as limited data tomography. To strike a balance between the expressivity and efficiency for the denoiser  $D_\sigma$ , a shallow network (e.g., composed of two layers) is employed in the numerical tests. This allows a precise estimate of  $L_\sigma$ , which is controllable during training by weight normalization.

In the second part of the talk, the extrapolated PnP strategy in (7) meets the deep unfolded framework of  $\Psi$ DONet, by studying the convergence analysis of the following update:

$$(8) \quad \begin{cases} u^{(k+1)} &= D_{\sigma_k}(z^{(k)} - \gamma_k \nabla f(z^{(k)}) - \beta_k \Lambda_{\zeta_k} u^{(k)}), \\ z^{(k+1)} &= u^{(k+1)} + \alpha_{k+1}(u^{(k+1)} - u^{(k)}). \end{cases}$$

Effectively, the iteration above turns the (extrapolated) PnP algorithm into a deep unfolded architectures: in practice, the PnP iteration is truncated to a fixed number of iterations so that the denoiser  $D_{\sigma_k}$  and the CNN  $\Lambda_{\zeta_k}$  are jointly trained end-to-end in a supervised fashion. Linear convergence of the sequence in (8) to a stationary point of the same data-driven regularization functional  $F_\sigma = f(u) + \phi_\sigma(u)$  from the first part of the talk is guaranteed under similar assumptions on the denoiser  $D_\sigma$  (including  $L_\sigma < 1$ ) and with the additional requirement that the intensity of the  $\Psi$ DO correction  $\Lambda_{\zeta_k}$  decays sufficiently fast as  $k \rightarrow \infty$ . It is also possible to prove a stability result of the learned reconstruction method in (8), relevant to provide sample error estimates and study its generalization properties.

The advantages of the update (8) are demonstrated in the context of limited angle tomography. Even though the convergence analysis only ensures a linear convergence rate, in practice it can be observed that the extrapolation step in (7)-(8) allows to reach a more sensible solution by using a lower number of layers (i.e., unrolling a smaller number of iterations). In the context of limited angle tomography, enabling the operator correction in (8) encoded by  $\Psi$ DONet is essential to smooth away the singularities, effectively improving the reconstruction quality.

## REFERENCES

- [1] S. Arridge, P. Maaß, O. Öktem and C.-B. Schönlieb, *Solving inverse problems using data-driven models*, Acta Numer. **28** (2019), 1–174.

- [2] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci. **2**(1) (2009), 183–202.
- [3] T.A. Bubba (ed.), *Data-driven models in inverse problems*, De Gruyter (2024).
- [4] T.A. Bubba, M. Galinier, M. Lassas, M. Prato, L. Ratti and S. Siltanen, *Deep neural networks for inverse problems with pseudodifferential operators: an application to limited-angle tomography*, SIAM J. Imaging Sci. **14**(2) (2021), 470–505.
- [5] I. Daubechies, M. Defrise and C. De Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Commun. Pure Appl. Math. **57** (11) (2004), 1413–1457.
- [6] K. Gregor and Y. LeCun, *Learning fast approximations of sparse coding*, Proc. 27th International Conference on Machine Learning (2010), 399–406.
- [7] S. Hurault, A. Chambolle, A. Leclaire and N. Papadakis, *Convergent plug-and-play with proximal denoiser and unconstrained regularization parameter*, J. Math Imaging Vis. **66** (2024), 616–638.
- [8] F. Natterer, *The mathematics of computerized tomography*, SIAM (2001).
- [9] E.T. Quinto, *Singularities of the X-Ray transform and limited data tomography in  $\mathbb{R}^2$  and  $\mathbb{R}^3$* , SIAM J. Math. Anal. **24**(5) (1993), 1215–1225.
- [10] S. Venkatakrishnan, C.A. Bouman and B. Wohlberg, *Plug-and-play priors for model based reconstruction*, 2013 IEEE GlobalSIP, 945–948.

## Adaptivity, and expressivity in FKS and neural network approximations

CHRIS BUDD

(joint work with Simon Arridge, Teo Deveney, Lisa Kreusser)

In this talk consider the problem of univariate nonlinear function approximation using shallow neural networks (NN) with a rectified linear unit (ReLU) activation function. We show that the  $L_2$  based approximation problem is ill-conditioned and the behaviour of optimisation algorithms used in training these networks degrades rapidly as the width of the network increases. This can lead to significantly poorer approximation in practice than we would expect both from the theoretical expressivity of the ReLU architecture and with traditional methods such as univariate Free Knot Splines (FKS). This impacts on the accuracy of using these methods in the context of a PINN to solve a differential equation, leading to poor convergence in this case.

Univariate shallow ReLU, NNs and FKS span the same function space, and thus have the same theoretical expressivity. However, the FKS representation, remains well-conditioned as the number of knots increases. We leverage the theory of optimal piecewise linear interpolants to improve the training procedure for a ReLU, NN. Using the equidistribution principle, we propose a two-level procedure for training the FKS by first solving the nonlinear problem of finding the optimal knot locations of the interpolating (I)FKS, and then the nearly linear, well-conditioned, problem of finding the optimal weights and knots of the FKS.

The training of the FKS gives insights into how we can train a ReLU NN effectively to give an equally accurate approximation. To do this we combine the training of the ReLU NN with an equidistribution based loss to find the breakpoints of the ReLU functions, this is then combined with preconditioning the ReLU, NN

approximation (to take an FKS form) to find the scalings of the ReLU functions. This procedure leads to a fast, well-conditioned and reliable method of finding an accurate shallow ReLU,NN approximation to a univariate target function. We test this method on a series of regular, singular, and rapidly varying target functions and obtain good results, realising the expressivity of the shallow ReLU network in all cases. We then extend our results to deeper networks.

## **Approximating the nonlinear inverse by neural operators**

MAARTEN V. DE HOOP

(joint work with Nikola B. Kovachi, Matti Lassas, Nicholas H. Nelsen,  
Andrew M. Stuart)

We give a general procedure for approximating the solution operators of inverse problems with neural operators. We present an abstract framework with conditions but focus on studying Calderón’s inverse conductivity problem in dimension two, on a bounded domain,  $\Omega$ , with smooth boundary, as a concrete example. The main contribution of this work is an approximation theorem for the inversion operator that holds even outside of the range of the forward operator making use of the Benyamini-Lindenstrauss extension theorem. To obtain this result, we consider the kernel of the Neumann-to-Dirichlet (NtD) map, identified as a Hilbert-Schmidt operator, as the data. The inversion operator maps this kernel to the conductivity. The conductivity belongs to the set requiring that it has a finite BV norm. This set contains discontinuous conductivities; its image under the forward map is compact in  $L^2(\partial\Omega \times \partial\Omega)$ , which follows from the continuity of this map. By interpolating the operator norm using Hilbert-Schmidt truncations we get a forward continuity result for the kernels. We show that the extension of the inversion map preserves the (logarithmic) modulus of continuity.

In the framework of learning the inverse, the noise in the NtD kernel must be small (based on the logarithmic stability of the extension) enough. We then give a universal approximation theorem with certain neural operators with explicit expression rates: Our quantitative rates relate the dependencies of the depth, width and rank of the neural operations in terms of the radius of compact sets, the modulus of continuity and the desired approximation error. By assuming tanh activation functions, the neural operator takes values in the same Sobolev space as the nonlinear operator it approximates by Meyers-Serrin theorem. We introduce nonlocal, integral kernel operator contributions in the first and last layers only. Despite the typical inclusion of (infinite-rank) integral kernel operators in the hidden layers of neural operators, this is not necessary for the constructed class of neural operators to be universal approximators. Earlier work on out of distribution risk bounds upon introducing stochastic depth (J.A. Lara Benitez, T. Furuya, F. Faucher, A. Kratsios, X. Tricoche & M.V. de Hoop, 2024) can be adapted to these results.

## Bilevel learning for PDE inverse problems

JUAN CARLOS DE LOS REYES

In recent years, novel optimization ideas have been applied to several inverse problems in combination with data-driven approaches, to improve the inversion by optimally choosing different parameters and quantities of interest. A rigorous framework to deal with these types of problems is bilevel optimization. Within this, the model-based inverse problem is considered as lower-level instance, while on the upper level a data-driven loss functional is minimized, usually considering a training set to learn from.

If the lower-level instance corresponds to a PDE inverse problem and a single training pair  $(u_{\text{true}}, y_{\text{obs}})$  is considered, the bilevel model for optimizing the weight  $\alpha \in \mathbb{R}^n$  in front of the regularizer can be mathematically formulated in the following form:

$$\begin{aligned}
 (1a) \quad & \underset{\alpha \in [\alpha_a, \alpha_b] \subset \mathbb{R}^n}{\text{minimize}} && \ell(u, u_{\text{true}}) \\
 (1b) \quad & \text{subject to} \begin{cases} \underset{(y, u)}{\text{minimize}} & \frac{1}{2} \int_{\mathcal{S}} |y - y_{\text{obs}}|^2 dx + \alpha \cdot \mathcal{R}(u) \\ \text{subject to} & e(y, u) = 0, \end{cases}
 \end{aligned}$$

where  $u_{\text{true}}$  is a known parameter from which learning is possible, and  $\ell$  is a loss function; for instance, the square loss,

$$\ell(u, u_{\text{true}}) := \|u - u_{\text{true}}\|_{L^2}^2.$$

The vectors  $\alpha_a, \alpha_b$ , satisfying  $0 < \alpha_a \leq \alpha_b < \infty$  componentwise, establish lower and upper bounds for the parameters to be found.

Concerning the lower-level problem (1b),  $\mathcal{S}$  is a subset of the domain  $\Omega$  for which observations are available,  $\mathcal{R} : U \rightarrow \mathbb{R}^n$  is the regularizer that incorporates prior information about the solution of the problem and  $y_{\text{obs}}$  are observations of the PDE state  $y \in Y$ . The PDE is expressed through the equation  $e(y, u) = 0$ , where  $e : Y \times U \rightarrow W$  is assumed to be Fréchet differentiable, with  $Y, U, W$  being reflexive Banach spaces. Moreover, it is assumed that  $e_y(\bar{y}, \bar{u}) \in \mathcal{L}(Y, W)$  is a bijection, to guarantee existence of an adjoint state (see, e.g., [2]).

Differently from classical imaging tasks, like denoising (see, e.g., [3]), if the inverse problem is governed by a PDE, the bilevel optimization problem structure becomes quite involved to be analyzed, as the convexity of the lower-level problem gets lost in general [1, 5].

The existence of an optimal vector  $\alpha$  can be proven under certain assumptions about the PDE, namely, the uniqueness of its solution, the weak continuity of the solution mapping, and the boundedness of the states corresponding to bounded parameters. Additionally, the regularizer  $\mathcal{R}$  is required to be coercive and weakly lower semi-continuous.

Another significant consequence of the loss of convexity in the lower-level instance (1b) is that the bilevel problem cannot simply be reformulated as a single-level one, by replacing the lower-level problem by its first-order necessary optimality condition. Instead, alternative strategies must be considered, incorporating first- and second-order optimality conditions for the lower-level problem, which remains an ongoing area of research.

## REFERENCES

- [1] P. Castro and J. C. De los Reyes. *A bilevel learning approach for optimal observation placement in variational data assimilation*. Inverse Problems, **36** (2020), 035202020.
- [2] J. C. De los Reyes, *Numerical PDE-Constrained Optimization*, Springer (2015).
- [3] J.C. De los Reyes. *Bilevel Imaging Learning Problems as Mathematical Programs with Complementarity Constraints: Reformulation and Theory*. SIAM Journal on Imaging Sciences, **16** (2023), 1655-16863.
- [4] E. Haber and L. Tenorio. *Learning regularization functionals—a supervised training approach*. Inverse Problems **19** (2003), 611.
- [5] G. Holler, K. Kunisch and C. Barnard. *A bilevel approach for parameter learning in inverse problems*. Inverse Problems, **14** (2018), 115012.

## Extending neural operator with Laplacian eigenfunctions for arbitrary domains

ALEXANDER DENKER

(joint work with Željko Kereta, Simon Arridge)

Neural operators learn mappings between infinite-dimensional function spaces with applications for solving solution operators for differential equations [1]. In particular, neural operators are intended to address a major drawback of typical learning approaches for differential equations by being discretisation-invariance, i.e., they act on, and can adapt to, any discretisation of the input function. Fourier neural operator (FNO) [2] is a particularly effective method that leverages the fast Fourier transform (FFT) to learn transformations in the frequency domain. However, the efficient implementation with the FFT constrains the FNO to rectangular domains with a uniform discretisation, which limits possible applications.

Extensions of FNO methods to non-rectangular domains have been extensively studied in recent years. A common method is to embed the irregular domain into a larger rectangle [3]. Another line of work tries to first transform, or interpolate, the data onto a regular grid before applying the FNO, cf. Geo-FNO [4] or GINO [5]. However, this introduces the additional challenge of learning, or defining, the transformation between the irregular and the regular grid. DSE-FNO [6] replaces the FFT with a non-uniform fast Fourier transform [7]. As an alternative to Fourier features, in [9] the authors propose to use a decomposition into eigenfunctions of the Laplace-Beltrami operator to learn neural operators on manifolds.

In a typical learning setting we have access to a dataset  $\{(a^{(i)}, u^{(i)})\}_{i=1}^N$  arising from some, possibly unknown, parametric PDE

$$(\mathcal{L}_a u)(x) = f(x), \quad x \in \Omega,$$



with appropriate boundary conditions. The goal is to approximate the solution operator  $G_f : a \mapsto u$  with a neural operator  $\mathcal{G}_\theta$ . The basic building block of most neural operators  $\mathcal{G}_\theta$  are iterative updates of the form

$$v_{t+1}(x) = \sigma(Wv_t(x) + [\mathcal{T}v_t](x)),$$

where  $v_t : \Omega \rightarrow \mathbb{R}^{d_v}$  is the input function to the block,  $x \in \Omega$  is the query point,  $W \in \mathbb{R}^{d_v \times d_v}$  is a learnable weight matrix and  $\sigma$  is a non-linear activation function, applied point-wise [1]. The operator  $\mathcal{T}$  is chosen as a linear integral operator

$$(\mathcal{T}v_t)(x) := \int_{\Omega} \kappa(x, y)v_t(y)dy.$$

Different choices of the kernel  $\kappa$  result in different neural operator architectures. Prominently, FNO [2] imposes a stationary kernel  $\kappa(x-y)$ . In this case, the integral operator reduces to a convolution operator and can be expressed as  $(\mathcal{T}_{\text{FNO}}v_t)(x) := \mathcal{F}^{-1}(\mathcal{F}(\kappa) \cdot \mathcal{F}(v_t))(x)$ , where  $\mathcal{F}$  is the Fourier transform.

**Laplacian operator network.** In this work we follow a different approach. We assume that we have a symmetric and positive definite kernel  $\kappa \in L^2(\Omega \times \Omega)$ . Then, following Mercer's theorem [10], there exists an orthonormal basis  $\{u_i\}_i$  with non-negative scalars  $\{\lambda_i\}_i$ , which can respectively be identified as eigenfunctions and eigenvalues of the corresponding integral operator, such that the kernel can be represented as

$$\kappa(s, t) = \sum_{i=1}^{\infty} \lambda_i u_i(s)u_i(t) \approx \sum_{i=1}^M \lambda_i u_i(s)u_i(t).$$

Approximating the kernel with the first  $M$  terms, we can define building blocks as

$$[\mathcal{T}_{\lambda}v_t](x) := \int_{\Omega} \sum_{i=1}^M \lambda_i u_i(x)u_i(s)v_t(s)ds = \sum_{i=1}^M \lambda_i \langle u_i, v_t \rangle u_i(x),$$

where  $\lambda = (\lambda_1, \dots, \lambda_M)^\top \in \mathbb{R}_{\geq 0}^M$  encodes learnable parameters. Using the operators

$$\mathcal{E} : L^2(\Omega) \rightarrow \mathbb{R}^M, \quad \mathcal{E}^* : \mathbb{R}^M \rightarrow L^2(\Omega)$$

$$u \mapsto (\langle u, u_1 \rangle, \dots, \langle u, u_M \rangle)^\top \quad (c_1, \dots, c_M)^\top \mapsto \sum_{i=1}^M c_i u_i,$$

the building block can in compact notation be written as

$$(1) \quad [\mathcal{T}_{\lambda}v_t](x) := \mathcal{E}^*(\lambda \odot \mathcal{E}(v_t))(x).$$

Mappings  $\mathcal{E}$  can be generalised to frame analysis and synthesis operators, where  $\{u_1, \dots, u_M\}$  can be linearly dependent. Recently [9], propose using eigenfunctions of the Laplacian. We refer to this as the Laplacian Operator Network (LONE). The eigenfunctions are given as the solutions of the Laplacian eigenvalue problem:

$$(2) \quad \begin{aligned} -\Delta u &= \lambda u \text{ in } \Omega, \\ u &= 0 \text{ on } \partial\Omega, \end{aligned}$$

where  $u \in C^2(\Omega)$  is the *eigenfunction* and  $\lambda \in \mathbb{R}$  the corresponding *eigenvalue*. Note that in (2) we use Dirichlet boundary conditions, though other choices are possible. Eigenfunctions can be computed analytically on simple domains, e.g., on a rectangle the eigenfunctions are given as sines or cosines. In this case eigenfunction decomposition corresponds to the discrete sine/cosine transform for regularly spaced data. Further, eigenfunctions on the sphere are given by spherical Bessel functions, in which case our architecture directly recovers the spherical-FNO [8].

**Numerics.** To highlight the ability of LONE to deal with arbitrary domains, we study the Darcy Flow on different domains  $\Omega \subset \mathbb{R}^2$  given by

$$\begin{aligned} -\nabla(a(x)\nabla u(x)) &= f(x), \quad x \in \Omega \\ u(x) &= 0 \quad \text{on } \Omega, \end{aligned}$$

with diffusion coefficient  $a \in L^\infty(\Omega)$  and forcing function  $f \in L^2(\Omega)$ . The goal is to learn the parameter-to-solution map  $G^\dagger : L^\infty(\Omega) \rightarrow H_0^1(\Omega)$  for a fixed forcing function  $f$ . We consider a similar setting to [2] with  $f \equiv 1$ .

We compare LONE against DSE-FNO [6] and Geo-FNO [4] on three different domains: rectangle, disk and L-shape. For all neural operators we use a similar number of coefficients and the same training setup. For both Geo-FNO and DSE-FNO we use  $12 \times 12$  Fourier modes, 32 channels and 4 layers. For LONE we use the first 128 eigenfunctions, 32 channels and 4 layers.

TABLE 1. **Darcy Flow:** Comparison for different domains  $\Omega$ .

	Rectangle		Disk		L-shape	
	AE	RE [%]	AE	RE [%]	AE	RE [%]
<b>Geo-FNO</b>	2.56e−4	4.49	2.01e−3	3.19	7.49e−5	3.20
<b>DSE-FNO</b>	1.44e−4	2.50	1.85e−3	2.92	5.02e−5	2.13
<b>LONE</b>	6.09e−5	1.05	1.26e−3	2.02	2.83e−5	1.20

The absolute and the relative  $L_1$ -error are presented in Table 1. For all domain types we see that LONE performs better than FNO approaches. The training of Geo-FNO was more unstable, as the transformation has to be learned in parallel, resulting in a worse performance across all domains. Even for the rectangle, there is a slight increase in the performance when using LONE. This can be explained by the fact that Laplace eigenfunctions directly encode the Dirichlet boundary conditions of the Darcy Flow. This highlights an interesting feature of LONE: eigenfunctions can be constructed for a specific PDE.

These initial results suggest that for applications involving diverse and complex geometries, LONE may offer a more robust solution. In future work, we want to extend this framework to time-dependent PDEs and evaluate other choices instead of the Laplacian eigenfunctions.

## REFERENCES

- [1] K. Azizzadenesheli et al., *Neural operators for accelerating scientific simulations and design*, Nature Reviews Physics (2024), 1–9.
- [2] Z. Li et al., *Fourier Neural Operator for Parametric Partial Differential Equations*, International Conference on Learning Representations (2021).
- [3] L. Lu et al., *A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data*, Computer Methods in Applied Mechanics and Engineering 393 (2022): 114778.
- [4] Z. Li et al., *Fourier neural operator with learned deformations for pdes on general geometries*, Journal of Machine Learning Research 24.388 (2023): 1–26.
- [5] Z. Li et al., *Geometry-informed neural operator for large-scale 3d pdes*, Advances in Neural Information Processing Systems 36 (2024).
- [6] L. Lingsch et al., *Beyond Regular Grids: Fourier-Based Neural Operators on Arbitrary Domains*, arXiv preprint arXiv:2305.19663 (2023).
- [7] L. Greengard and L. June-Yub, *Accelerating the nonuniform fast Fourier transform*, SIAM review 46.3 (2004): 443–454.
- [8] B. Bonev et al., *Spherical fourier neural operators: Learning stable dynamics on the sphere*, International conference on machine learning, PMLR, 2023.
- [9] G. Chen et al. *Learning neural operators on riemannian manifolds*, arXiv preprint arXiv:2302.08166 (2023).
- [10] J. Mercer *Functions of positive and negative type, and their connection the theory of integral equations*, Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character 209.441-458 (1909): 415–446.

**Solving PDE-related inverse problems with random feature models**

FELIX DIETRICH

We discuss a sampling scheme for a data-dependent probability distribution of the parameters of neural networks. Such sampled networks are provably dense in the continuous functions, and have a convergence rate in the number of neurons that is independent of the input dimension. Using sampled neurons as basis functions in an ansatz allow us to effectively solve partial differential equations. In computational experiments comparing training speed and accuracy, the sampling scheme outperforms iterative, gradient-based optimization by several orders of magnitude. For inverse problems involving parameter fields, sampled neurons offer expressive basis functions for the unknown field as well as the PDE solution.

## 1. INTRODUCTION

The setting we discuss concerns inverse problems that are related to partial differential equations (PDEs) with unknown parameter (fields). The forward problem (cf. (1), left) involves the solution of a parameterized, elliptic PDE, on which we have worked on before [2, 8]. The inverse problem (cf. (1), right) starts with a set of values obtained from a solution of this PDE where the parameter field is unknown and to be determined.

Inverse problems pose a challenge for iterative training methods because the loss landscape is typically more complicated to navigate. Owing to the problems associated with overfitting to noisy data and high sensitivity of accuracy

TABLE 1. Forward vs. inverse problem settings. In this extended abstract, we discuss the inverse problem case, learning parameter fields for parameterized PDEs.

<b>Solving equations</b>	<b>Learning equations</b>
Given PDE (parameters)	Given solution
Solve PDE	Solve inverse problem
To get solution	To get PDE (parameters)
(Forward problem)	(Inverse problem)

with respect to the number of neurons in PIELM, [6] propose a Bayesian physics-informed extreme learning machine (BPIELM) to solve both forward and inverse linear PDE problems with noisy data. [4] use distributed PINNs to solve inverse problems for predicting scalar coefficients of PDEs. [7] propose VC-PINN for PDE problems with variable coefficients. [5] discuss PINNs for full-waveform inversion problems. [3] use ELMs to solve inverse PDE problems involving predictions of scalar coefficients in PDEs and space-varying coefficients.

## 2. CONSTRUCTING THE NEURAL NETWORK ANSATZ WITHOUT GRADIENT DESCENT

We parameterize the approximation of a solution with a neural network with one hidden layer, activation function  $\sigma = \tanh$ ,  $M$  neurons, so that

$$(1) \quad \hat{u}() = C[\Phi(x), \mathbb{1}] = c\sigma(Wx^\top + b) + c_0.$$

Here,  $c \in \mathbb{R}^{1 \times M}$  and  $c_0 \in \mathbb{R}$  are real-valued parameters of the “last layer” of the network,  $W \in \mathbb{R}^{M \times d}$  and  $b \in \mathbb{R}^{M \times 1}$  are internal parameters of the single “hidden layer”, and  $C := [c, c_0] \in \mathbb{R}^{1 \times (M+1)}$ . The activation functions are stacked in  $\Phi = [\phi_1, \dots, \phi_M]$ , where  $\phi_m(x) = \sigma(w_m x^\top + b_m)$ .

To obtain the parameters of the hidden layer, we employ a recently developed sampling approach [1]. It is different from the extreme learning machine (ELM) framework, where the weight and bias space is the space  $\mathbb{R}^{M \times d} \times [-\eta, \eta]$ , where  $\eta$  is sufficiently large. Our approach, the sampling where it matters (SWIM) framework, follows our previous work [1] and restricts the weight space to  $\Omega \times \Omega$ . We construct each weight and bias pair  $w_m, b_m$  by taking two points  $x^{(1)}, x^{(2)} \in \Omega$  and construct the weight and bias as  $w_m = s_1 \frac{x^{(2)} - x^{(1)}}{\|x^{(2)} - x^{(1)}\|^2}$ ,  $b_m = -\langle w_m, x^{(1)} \rangle + s_2$ , where  $s_1, s_2$  are constants dependent on the activation function. In the unsupervised setting of solving PDEs, one can choose pairs of collocation points from a uniform distribution over all possible pairs of collocation points, which is the default setting in this paper, as we do not know the solution of the PDE beforehand.

The key benefits of randomly sampling basis functions include much shorter training times and improved accuracy compared to PINNs (both from one to five orders of magnitude, see [2, 8]), nearly matching the numerical state-of-the-art solvers. Moreover, the advantages compared to the classical numerical solvers such

as finite elements, finite differences, or finite volume approaches include spectral convergence (i.e., requiring much fewer basis functions) without requiring a mesh, making it much easier to implement on complex geometries.

The suitability of each of the proposed approaches depends on the true PDE solution’s gradient distribution. We empirically observe that ELM performs better in approximating solutions with shallow gradients, while SWIM (by sampling weights from close data points) performs better in approximating solutions with steep gradients.

### 3. COMPUTATIONAL EXPERIMENTS

We now demonstrate the efficacy of our approach in solving an inverse problem involving the estimation of an unknown coefficient field. Here, we repeat the experiment from [3] and set up an inverse problem involving the Helmholtz equation on  $\Omega = [0, 1.5]^2$  with Dirichlet (fixed-value) boundary conditions, and a space-dependent parameter field  $\gamma$ , so that

$$\begin{aligned}\Delta u(x) - \gamma(x)u(x) &= f(x) \quad \text{for } x \in \Omega, \\ u(x) &= g(x) \quad \text{for } x \in \partial\Omega.\end{aligned}$$

For  $x = [x_1, x_2]^T \in \Omega \subset \mathbb{R}^2$ , the true solution  $u$  and coefficient field  $\gamma$  involved in the PDE are defined by

$$(2a) \quad \gamma(x_1, x_2) = 100 \left( 1 + \frac{1}{4} \sin(2\pi x_1) + \frac{1}{4} \sin(2\pi x_2) \right),$$

$$(2b) \quad \begin{aligned}u(x_1, x_2) &= \left( \frac{5}{2} \sin\left(\pi x_1 - \frac{2\pi}{5}\right) + \frac{3}{2} \cos\left(2\pi x_1 + \frac{3\pi}{10}\right) \right) \\ &\quad \times \left( \frac{5}{2} \sin\left(\pi x_2 - \frac{2\pi}{5}\right) + \frac{3}{2} \cos\left(2\pi x_2 + \frac{3\pi}{10}\right) \right).\end{aligned}$$

The forcing term is constructed by substituting these functions in the PDE.

TABLE 2. Summary of results for the computational experiment.

Architecture	(2,400,1) for $u$ and $\gamma$
Relative $L^2$ error (PDE)	$1.85\text{e-}2 \pm 1.5\text{e-}2$
Rel. $L^2$ error ( $\gamma$ )	$1.23\text{e-}2 \pm 9.10\text{e-}3$
Training time (s)	$5.2\text{e-}2 \pm 1.6\text{e-}2$

We solve the Helmholtz equation involving the inverse space-dependent coefficient field on the domain  $\Omega = [0, 1.5]^2$ , where the true solution and coefficient field of a manufactured solution are described in (2). We assume access to values of  $u$  at  $N = 300$  measurement points, distributed uniformly at random inside  $\Omega$ . The goal is to construct an accurate approximation of the solution  $u(x_1, x_2)$  and space-dependent coefficient field  $\gamma(x_1, x_2)$  on  $\Omega$ .

We parametrize the solution of the Helmholtz equation and the coefficient field with two different SWIM networks with a single hidden layer with  $\tanh$  activation functions with widths  $m_u$  and  $m_\gamma$ , respectively. We write  $u(x_1, x_2) = \phi_u(X)c + c_0 = (\phi_u(X), 1)C$  and  $\gamma(x_1, x_2) = \phi_\gamma(X)d + d_0 = (\phi_\gamma(X), 1)D$ , where,  $C := (c^T, c_0) \in \mathbb{R}^{m_u+1}$ ,  $D := (d^T, d_0) \in \mathbb{R}^{m_\gamma+1}$  and outputs of hidden layers  $\phi_u(X) \in \mathbb{R}^{N \times m_u}$  and  $\phi_\gamma(X) \in \mathbb{R}^{N \times m_\gamma}$ . Since the last layer parameters of both these networks  $C$  and  $D$ , respectively, are unknown, the term  $\gamma \times u$  in the Helmholtz equation introduces a non-linearity in the coefficients, though the PDE is linear in the solution  $u$ . We use the alternating least squares algorithm to solve for the coefficients  $C$  and  $D$ . Our setting in the manuscript is much more challenging compared to [3], as we neither distribute additional measurement points on the boundary nor use domain decomposition and use multiple neural networks. So, we cannot compare our results directly. (1) shows the resulting approximation with a network of  $M = 400$  neurons in one hidden layer. The relative error is plotted over a grid of  $101 \times 101$  test points. (2) lists all results in this section.

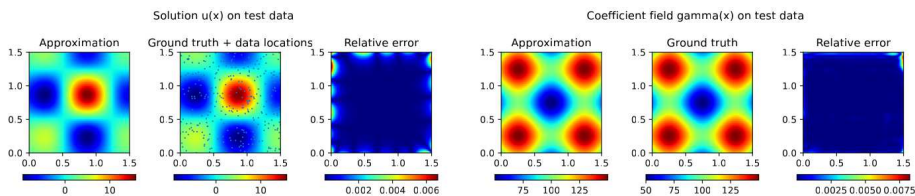


FIGURE 1. Left: Ground truth solution and approximation for the Helmholtz PDE. Right: Ground truth parameter field  $\gamma$  and our approximation.

## REFERENCES

- [1] Erik L Bolager, Iryna Burak, Chinmay Datar, Qing Sun, and Felix Dietrich. Sampling weights of deep neural networks. In *Advances in Neural Information Processing Systems*, volume 36, pages 63075–63116. Curran Associates, Inc., 2023.
- [2] Chinmay Datar, Taniya Kapoor, Abhishek Chandra, Qing Sun, Iryna Burak, Erik Lien Bolager, Anna Veselovska, Massimo Fornasier, and Felix Dietrich. Solving partial differential equations with sampled neural networks, May 2024.
- [3] Suchuan Dong and Yiran Wang. A method for computing inverse parametric pde problems with random-weight neural networks. *Journal of Computational Physics*, 489:112263, 2023.
- [4] Vikas Dwivedi, Nishant Parashar, and Balaji Srinivasan. Distributed learning machines for solving forward and inverse problems in partial differential equations. *Neurocomputing*, 420:299–316, 2021.
- [5] Leon Herrmann, Tim B urchner, Felix Dietrich, and Stefan Kollmannsberger. On the use of neural networks for full waveform inversion. *Computer Methods in Applied Mechanics and Engineering*, 415:116278, October 2023.
- [6] Xu Liu, Wen Yao, Wei Peng, and Weien Zhou. Bayesian physics-informed extreme learning machine for forward and inverse pde problems with noisy data. *Neurocomputing*, 549:126425, 2023.

- [7] Zhengwu Miao and Yong Chen. Vc-pinn: Variable coefficient physics-informed neural network for forward and inverse problems of pdes with variable coefficient. *Physica D: Nonlinear Phenomena*, 456:133945, 2023.
- [8] Atamert Rahma, Chinmay Datar, and Felix Dietrich. Training Hamiltonian neural networks without backpropagation, November 2024.

## New universal operator approximation result for deep operator networks

JANEK GÖDEKE

(joint work with Peter Maaß)

Learning operators between infinite-dimensional function spaces is highly desirable in fields like partial differential equations (PDEs). For instance, learning parameter-to-state maps that map the parameter function of a PDE to the corresponding solution. Over the last five years, several deep learning frameworks have emerged, such as Deep Operator Networks (DeepONets) or (Fourier) Neural Operators. On the theoretical side, universal approximation theorems have been derived, which state that networks of these types can approximate many operators arbitrarily well in different topologies. A frequently considered type of approximating continuous (non-linear) operators  $G : \mathcal{X} \rightarrow \mathcal{Y}$  between suitable (real) Banach spaces is the following: It has been shown that, for each compact set  $K \subset \mathcal{X}$ , there exists a sequence of networks  $G_{n,K} : \mathcal{X} \rightarrow \mathcal{Y}$  (e.g., DeepONets [2], [3]; (Fourier) Neural Operators [4], [6]) that converges uniformly to  $G$  on this compact set, i.e.

$$\sup_{f \in K} \|G(f) - G_{n,K}(f)\| \xrightarrow{n \rightarrow \infty} 0.$$

However, it has been an open question whether a universal sequence  $G_n$  can be chosen that approximates the operator as above, but on every compact set  $K$ .

We have found a proof for DeepONet-like architectures, which we state below in Theorem 1. One of the keys is the bounded approximation property of separable Banach spaces. For a definition, we follow [1].

**Definition 1.** A separable Banach space  $\mathcal{X}$  is said to have the bounded approximation property (BAP) if there exists a sequence of linear, bounded, finite rank operators  $T_n : \mathcal{X} \rightarrow \mathcal{X}$  such that for all compact  $K \subset \mathcal{X}$  it holds that

$$\sup_{f \in K} \|f - T_n f\| \xrightarrow{n \rightarrow \infty} 0.$$

**Theorem 1.** Let  $G : \mathcal{X} \rightarrow \mathcal{Y}$  be a continuous mapping between separable Banach spaces. Assume that  $\mathcal{Y}$  has a Schauder basis  $(b_k)_{k \in \mathbb{N}}$ , and  $\mathcal{X}$  has the BAP with mappings  $T_n$ . Consider any isomorphisms  $\psi_n : T_n(\mathcal{X}) \rightarrow \mathbb{R}^{d_n}$ , where  $d_n$  is the finite rank of  $T_n$ . Then there exists a sequence of neural networks  $\varphi_n : \mathbb{R}^{d_n} \rightarrow \mathbb{R}^n$  such that for every compact  $K \subset \mathcal{X}$  it is

$$\sup_{f \in K} \left\| G(f) - \sum_{k=1}^n \varphi_{n,k} \left( (\psi_n \circ T_n) f \right) b_k \right\| \xrightarrow{n \rightarrow \infty} 0,$$

where  $\varphi_{n,k}$  denotes the  $k$ th output of the neural network  $\varphi_n$ .

**Example 1.** Let  $\mathcal{X}$  be a Banach space having a Schauder basis  $(b_k)_{k \in \mathbb{N}}$ , i.e. for each  $f \in \mathcal{X}$  there are unique coefficients  $c_k(f)$  such that  $f = \sum_k c_k(f) b_k$ . Then  $\mathcal{X}$  has the BAP and suitable choices for  $T_n$  are the projections

$$T_n f = \sum_{k=1}^n c_k(f) b_k.$$

Further, one can choose  $\psi_n : T_n(\mathcal{X}) \rightarrow \mathbb{R}^n$  such that  $(\psi_n \circ T_n) f = (c_1(f), \dots, c_n(f))^T$  for all  $f \in \mathcal{X}$ . Hence, the operator  $G$  in Theorem 1 can be approximated by

$$G_n(f) = \sum_{k=1}^n \varphi_{n,k} \left( (c_1(f), \dots, c_n(f))^T \right) b_k,$$

which is a special case of the MIONet introduced in [5].

**Example 2.** Let  $\Omega$  be a compact subset of a Banach space  $Z$ . It is well-known that the space of continuous real-valued functions  $\mathcal{C}(\Omega, \mathbb{R})$ , equipped with the supremum norm, has a Schauder basis and hence the BAP. However, the projections in Example 1 are not the only possible choice for the mappings  $T_n$  in Definition 1. In fact, one can find  $T_n$  and  $\psi_n$  such that for all  $f \in \mathcal{C}(\Omega, \mathbb{R})$  it holds that

$$(\psi_n \circ T_n) f = (f(y_1), \dots, f(y_{d_n}))^T,$$

where the sampling points  $y_1, \dots, y_{d_n}$  are an  $\frac{1}{n}$ -net of  $\Omega$ . Hence,  $G : \mathcal{C}(\Omega, \mathbb{R}) \rightarrow \mathcal{Y}$  can be approximated by

$$G_n(f) = \sum_{k=1}^n \varphi_{n,k} \left( ((f(y_1), \dots, f(y_{d_n}))^T) b_k \right).$$

If, for example,  $\mathcal{Y} = \mathcal{C}(\Omega_2, \mathbb{R})$ , one can approximate all  $b_k$  by neural networks, which leads to the classical form of a DeepONet [2], [3].

To sum up, Theorem 1 provides a stronger universal approximation result compared to the literature, as the approximating sequence  $G_n$  does not depend on  $K$ . Furthermore, it covers both taking Schauder basis coefficients and sampling points as an input for the networks  $\varphi_n$ .

## REFERENCES

- [1] P.G. Casazza and N.J. Kalton, *Notes on approximation properties in separable Banach spaces*, Geometry of Banach Spaces: Proceedings of the Conference Held in Strobl, Austria 1989, P.F.X. Müller and W. Schachermayer (eds.), Cambridge University Press, Cambridge (1991), pp. 49–64.
- [2] T. Chen, H. Chen, *Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems*, IEEE Transactions on Neural Networks (1995), 6(4), pp. 911–917.



- [3] L. Lu, P. Jin, Z. Zhang, G. Karniadakis, *Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators*, Nature Machine Intelligence (2021), 3, pp. 218–229.
- [4] N. Kovachki, S. Landthaler, S. Mishra, *On Universal Approximation and Error Bounds for Fourier Neural Operators*, Journal of Machine Learning Research (2021), 22, pp. 1–76.
- [5] P. Jin, S. Meng, L. Lu, *MIONet: Learning Multiple-Input Operators via Tensor Product*, SIAM Journal on Scientific Computing (2022), 44(6), pp. A3490–A3514.
- [6] N. Kovachki et al., *Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs*, Journal of Machine Learning Research (2023), 24, pp. 1–97.

## Generative modeling for turbulent flows

HANNO GOTTSCHALK

(joint work with Claudia Drygala, Francesca di Mare, Edmund Ross)

The rapid development of generative learning opened the way to envision new methods in numerical simulation. If looking at physical dynamical systems from a machine learning perspective, which is based on the statistical evaluation data, the most natural system to start with are chaotic systems and turbulent flows, in particular [3, 4, 5, 11, 12]. In fact, while statistical (machine) learning is based on the stabilization of averages of functions evaluated on identically and independently distributed random variables  $\frac{1}{n} \sum_{j=1}^n f(X_i) \rightarrow \mathbb{E}_{X \sim \mu}[f(X)]$  as  $n \rightarrow \infty$  with  $X, X_i \sim \mu$  where  $\mu$  is a measure on  $\Omega \subseteq \mathbb{R}^d$ , we have a similar kind of stabilization for the chaotic flow  $\varphi_t(x)$  of a dynamical system due to the Birkhoff ergodic theorem

$$\frac{1}{T} \int_0^T f(\varphi_t(x_0)) dt \rightarrow \mathbb{E}_{X \sim \mu}[f(X)], \text{ as } T \rightarrow \infty,$$

where  $\mu$  is the invariant measure of the flow  $\phi_t$ , i.e.  $\mu = \phi_{t,*}\mu = \mu \circ \phi_t^{-1}$ .

Many approaches in generative learning estimate a transport map  $\phi : \Omega \rightarrow \Omega$  from the data, which maps an easy to simulate noise distribution  $\nu$  on  $\Omega$  to the target distribution  $\mu$ , i.e.  $\phi_*\nu \approx \mu$ . From a mathematical standpoint, the analysis has to start with proving the existence of a sufficiently regular map  $\phi_0$  such that  $\phi_{0,*}\nu = \mu$ . Otherwise there would be little hope for consistent learning. This relates generative learning to the field of optimal transport. In fact, under suitable conditions on the regularity of densities  $f_\nu$  and  $f_\mu$  of the source and target measure  $\nu$  and  $\mu$ , respectively, the existence of  $\phi_0$  and  $C^{k,\alpha}$  Hölder regularity can be inferred from the Beckmann problem in optimal transport [14] in combination with classical elliptic regularity theory [1, 8].

Let thus  $\mathcal{H}^G$  be the hypothesis space of generators, consisting of maps  $\phi : \Omega \rightarrow \mathbb{R}^d$  which are capable to approximate Hölder uniformly bounded functions to precision, and let  $\mathcal{H}^D$  be another hypothesis space of discriminators, which consists of maps  $D : \Omega \rightarrow (0, 1)$  that assign a state  $x$  of the physical system the likelihood of being a snapshot from the invariant measure  $\mu$  vs being generated from  $\phi_*\nu$ . Usually, both spaces are represented by neural networks. Following the approach



FIGURE 1. Two snapshots of the turbulence strength in the Kármán vortex street: from Large eddy simulation [15] (left), and generated by DCGAN [5] (right).

of generative adversarial learning [10], we consider the empirical loss function

$$L(\phi, D, T, n) = \frac{1}{2T} \int_0^T \log(D(\varphi_t(x_0))) \, dt + \frac{1}{2n} \sum_{j=1}^n \log(1 - D(\phi(U_j)))$$

where  $U_j \sim \nu$  are i.i.d. noise random variables. In the limit  $T, n \rightarrow \infty$  this expression converges to the theoretical loss

$$L(\phi, D) = \frac{1}{2} \mathbb{E}_{X \sim \mu} [\log D(X)] + \frac{1}{2} \mathbb{E}_{X \sim \phi_* \nu} [\log(1 - D(X))],$$

which, when maximized over  $D : \Omega \rightarrow (0, 1)$  with  $D^* = \frac{f_\mu}{f_\mu + f_{\phi_* \nu}}$  equals the Jensen - Shannon divergence between  $\mu$  and  $\phi_* \nu$  up to a constant  $\log(2)$ . During learning, the JS-divergence is minimized through the two player game [10]

$$\min_{\phi \in \mathcal{G}^G} \max_{D \in \mathcal{H}^D} \hat{L}(\phi, D, T, n).$$

Let  $\hat{\phi}_{T,n}$  be the minimizer of the above problem, then using the usual decomposition of errors in generative adversarial learning one can show that  $d_{\text{JS}}(\mu, \hat{\phi}_{*} \nu) \rightarrow 0$  almost surely as  $T, n \rightarrow \infty$  provided one adaptively enlarges the hypothesis space to control the model error by universal approximation [2, 5].

High quality training data for turbulent flows is produced by a large eddy simulation [15]. To learn the vortex street, we employ the DCGAN architecture and for the low pressure turbine half stage under the influence of a wake generator the pix2pixHD architecture conditioned to the wake position. We show that GAN generated flow snapshots not only have a realistic appearance, see Fig. 1, but also learn physical quantities like average turbulence strength or variation of turbulence physically correctly, see [5] for a theoretical analysis. Note that the inference speed and compute resource requirements are considerably (often a factor 100 - 1000) below time and compute requirements of classical simulation. [7] provides a comparison of GAN learned turbulence with turbulence generation by further generative models like variational auto encoders or diffusion models.

Recently, video GAN architectures have produced compelling results in the generation of dynamic scenes. To leverage these capabilities for numerical simulation, we first have to understand the nature of VideoGAN architectures. Often, a tokenizer is employed to map the physical state into a sequence of coarse 'tokenized' states. From a mathematical standpoint, this can be understood as a low dimensional projection of the state space. The VideoGAN then learns the system's dynamics on sequences of such projected states and reconstructs the original state using a superresolution module. If the system theoretic requirement of *observability* is fulfilled, this dynamics and superresolution map can – in principle – be learned by encoding the tokenized autoregressive dynamics with a universal approximating family of functions, like neural networks. First experiments show highly realistic dynamics for GAN-generated time dynamics of the turbulent flow in the Kármán street including physically realistic time like correlations of the turbulence strength. We will revisit this point in a forthcoming paper.

## REFERENCES

- [1] S. Agmon, A. Douglis, and L. Nirenberg, Estimates near the boundary for solutions of elliptic partial differential equations satisfying general boundary conditions. i, Communications on pure and applied mathematics, 12 (1959), pp. 623–727.
- [2] Hayk Asatryan, Hanno Gottschalk, Marieke Lippert, and Matthias Rottmann. A convenient infinite dimensional framework for generative adversarial learning. arXiv preprint arXiv:2011.12087, 2020.
- [3] Mathis Bode, Michael Gauding, Zeyu Lian, Dominik Denker, Marco Davidovic, Konstantin Kleinheinz, Jenia Jitsev, and Heinz Pitsch. Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows. Proceedings of the Combustion Institute, 38(2):2617–2625, 2021.
- [4] Zhiwen Deng, Chuangxin He, Yingzheng Liu, and Kyung Chun Kim. Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework. Physics of Fluids, 31(12), 2019.
- [5] Claudia Drygala, Benjamin Winhart, Francesca di Mare, and Hanno Gottschalk. Generative modeling of turbulence. Physics of Fluids, 34(3), 2022.
- [6] Claudia Drygala, Francesca di Mare, and Hanno Gottschalk. Generalization capabilities of conditional gan for turbulent flow under changes of geometry. In Proceedings of the 15th International Conference on Evolutionary and Deterministic Methods For Design, Optimization and Control (EUROGEN 2023), 2023.
- [7] Claudia Drygala, Edmund Ross, Francesca di Mare and Hanno Gottschalk, Comparison of Generative Learning Methods for Turbulence Modeling, preprint 2024, arXiv:2411.16417
- [8] Emily C. Ehrhardt, Hanno Gottschalk and Tobias J. Riedlinger, Numerical and statistical analysis of NeuralODE with Rubge Kutta time integration, to appear
- [9] Uriel Frisch and Andrei Nikolaevich Kolmogorov. Turbulence: the legacy of AN Kolmogorov. Cambridge University Press, 1995.
- [10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative adversarial neural networks, Advances in neural information processing systems, 2014.
- [11] Junhyuk Kim and Changhoon Lee. Deep unsupervised learning of turbulence for inflow generation at various reynolds numbers. Journal of Computational Physics, 406:109216, 2020.
- [12] Hyojin Kim, Junhyuk Kim, Sungjin Won, and Changhoon Lee. Unsupervised deep learning for super-resolution reconstruction of turbulence. Journal of Fluid Mechanics, 910, 2021.

- [13] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun, editors, 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, 2016.
- [14] F. Santambrogio, Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs and Modeling, vol. 87 of Progress in Nonlinear Differential Equations and Their Applications, Springer Nature, Cham, 1st ed., 2015.
- [15] B. Winhart, M. Sinkwitz, A. Schramm, P. Post, and F. di Mare, “Large eddy simulation of periodic wake impact on boundary layer transition mechanisms on a highly loaded low-pressure turbine blade,” in Proceedings of the Turbo Expo: Power for Land, Sea, and Air, Vol. 84102, p. V02ET41A013.

## Multiscale methods for convolution neural networks

ELDAD HABER

### ABSTRACT

Convolutional Neural Networks (CNNs) are the backbone of many deep learning methods, but optimizing them remains computationally expensive. To address this, we explore multiscale training frameworks and mathematically identify key challenges, particularly when dealing with noisy inputs. Our analysis reveals that in the presence of noise, the gradient of standard CNNs in multiscale training may fail to converge as the mesh-size approaches to 0, undermining the optimization process. This insight drives the development of Mesh-Free Convolutions (MFCs), which are independent of input scale and avoid the pitfalls of traditional convolution kernels. We demonstrate that MFCs, with their robust gradient behavior, ensure convergence even with noisy inputs, enabling more efficient neural network optimization in multiscale settings. To validate the generality and effectiveness of our multiscale training approach, we show that (i) MFCs can theoretically deliver substantial computational speedups without sacrificing performance in practice, and (ii) standard convolutions benefit from our multiscale training framework in practice.

### 1. INTRODUCTION

In this work, we consider the task of learning a functional  $y(\mathbf{x}) = \phi(u(\mathbf{x}))$ , where  $\mathbf{x}$  is a position (in 2D  $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$  and in 3D  $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ ),  $u(\mathbf{x}) \in \mathcal{U}$  and  $y(\mathbf{x}) \in \mathcal{Y}$  are families of functions. To this end, we assume to have discrete samples from  $\mathcal{U}$  and  $\mathcal{Y}$ ,  $(\mathbf{u}_i^h = u_i(\mathbf{x}_h), \mathbf{y}_i^h = \phi(u_i(\mathbf{x}_h)))$ ,  $i = 1, \dots, M$  associated with some resolution  $h$ . A common approach to learning the function is to parameterize the problem, typically by a deep network, and replace  $\phi$  with a function  $f(\cdot, \cdot)$  that accepts the vector  $\mathbf{u}^h$  and learnable parameters  $\boldsymbol{\theta}$  which leads to the problem of estimating  $\boldsymbol{\theta}$  such that

$$(1) \quad \mathbf{y}_i^h \approx f(\mathbf{u}_i^h, \boldsymbol{\theta}), \quad i = 1, \dots, M.$$

To evaluate  $\boldsymbol{\theta}$ , the following stochastic optimization problem is formed and solved:

$$(2) \quad \min_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{u}^h, \mathbf{y}^h} \ell(f(\mathbf{u}^h, \boldsymbol{\theta}), \mathbf{y}^h),$$

Where  $\ell(\cdot, \cdot)$  is a loss function, typically mean square error. Standard approaches use variants of stochastic gradient descent (SGD) to estimate the loss and its gradient for different samples of  $(\mathbf{u}^h, \mathbf{y}^h)$ . In deep learning with convolutional neural networks, the parameter  $\boldsymbol{\theta}$  (the convolutional weights) has identical dimensions, independent of the resolution. Although SGD is widely used, its computational cost can become prohibitively high as the mesh-size  $h$  decreases, especially when evaluating the function  $f$  on a fine mesh for many samples  $\mathbf{u}_i^h$ . This challenge is worsened if the initial guess  $\boldsymbol{\theta}$  is far from optimal, requiring many costly iterations, for large data sizes  $M$ . One way to avoid large meshes is to use small crops of the data where large images are avoided, however, this can degrade performance, especially when a large receptive field is required for learning [3]

**Background and related work.** Computational cost reduction can be achieved by leveraging different resolutions, a concept foundational to multigrid and multiscale methods. These methods have a long history of solving partial differential equations and optimization problems [27, 5, 22]. Techniques like multigrid [27] and algorithms such as MGopt [22, 4] and Multilevel Monte Carlo [16, 15, 28] are widely used for optimization and differential equations.

In deep learning, multiscale or pyramidal approaches have been used in image processing tasks such as object detection, segmentation, and recognition, where analyzing multiple resolutions is key [24, 6], reviewed in [11]. Recent methods improve standard CNNs for multiscale computations by introducing specialized architectures and training methods. For instance, [18] uses multigrid methods in CNNs to boost efficiency and capture multiscale features, while [10] focuses on multiscale channel space learning, and [29] unifies both. [21] introduced the Fourier Neural Operator, enabling mesh-independent computations, and Wavelet-NNs were explored to capture multiscale features via wavelets [13, 12, 9].

While often overlooked, it is important to note that these approaches, can be divided into two families of approaches that leverage multiscale concepts. The *first* is to learn parameters for each scale, and a separate set of parameters that mix scales, as in UNet [23]. The *second*, called *multiscale training*, enables the approximation of fine-scale parameters using coarse-scale samples [17, 30, 8, 14]. The second approach aims to gain computational efficiency, as it approximates fine mesh parameters using coarse meshes, and it can be coupled with the first approach, and in particular with UNets.

**Our approach.** This work falls into the second category of multiscale training. We study multiscale algorithms that use coarse meshes to approximate high-resolution quantities, particularly the gradients of network parameters. Computing gradients on coarse grids is significantly cheaper than on fine grids, as noted in [26]. However, for efficient multiscale training, parameters on coarse and fine meshes must have "similar meaning," implying that both the loss and gradient on coarse meshes should approximate those on fine meshes. Specifically, the loss and

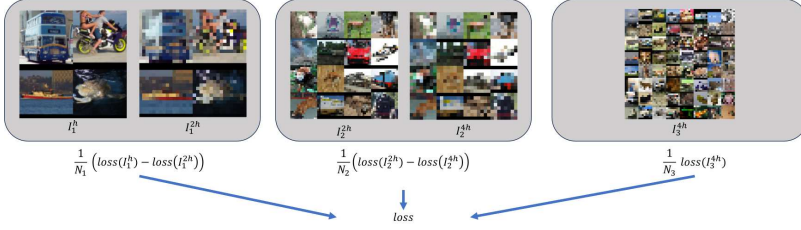


FIGURE 1. Illustration of our Multiscale-SGD, introduced in (2).

gradient with respect to parameters should converge to a finite value as  $h \rightarrow 0$ . In this work, we show that standard CNN gradients may not converge as the mesh size  $h$  approaches 0, suggesting CNNs under-utilize multiscale training. This motivates our development of mesh-free convolution kernels, whose values and gradients converge as  $h \rightarrow 0$ . Our approach builds on Differential-Operator theory [31] to create a family of learnable, mesh-independent convolutions for multiscale learning, resembling Fourier Neural Operators (FNO) [20] but with further expressibility.

Our main contributions are:

- (1) Propose a new multiscale training algorithm, Multiscale SGD.
- (2) Analyze the limitations of standard CNNs within a multiscale framework.
- (3) Introduce a family of mesh-independent CNNs inspired by differential operators.
- (4) Validate our approach on benchmark tasks, showcasing enhanced efficiency and scalability.

## 2. MULTISCALE STOCHASTIC GRADIENT DESCENT

We now present the standard approach of training CNNs, identify its major computational bottleneck, and propose a novel solution called Multiscale Stochastic Gradient Descent (Multiscale-SGD).

**Standard training of neural networks.** Suppose that we use a gradient descent-based method to train a CNN, with input resolution  $h^1$  with trainable parameters  $\theta$ . The  $k$ -th iteration reads:

$$(3) \quad \theta_{k+1} = \theta_k - \mu_k \mathbb{E} [\mathbf{g}(\mathbf{u}^h, \mathbf{y}^h, \theta_k)],$$

where  $\ell$  is some loss function (e.g., the mean-squared-error function), and the gradient of the loss with respect to the parameters is  $\mathbf{g}(\mathbf{u}^h, \mathbf{y}^h, \theta) = \nabla \ell(f(\mathbf{u}^h, \theta_k), \mathbf{y}^h)$ . The expectation  $\mathbb{E}$  is taken with respect to the input-label pairs  $(\mathbf{u}^h, \mathbf{y}^h)$ . Evaluating the expected value of the gradient can be highly expensive. To understand

---

<sup>1</sup>In this paper, we define resolution  $h$  as the pixel size on a 2D uniform meshgrid, where smaller  $h$  indicates higher resolution. For simplicity, we assume the same  $h$  across all dimensions, though different resolutions can be assigned per dimension.

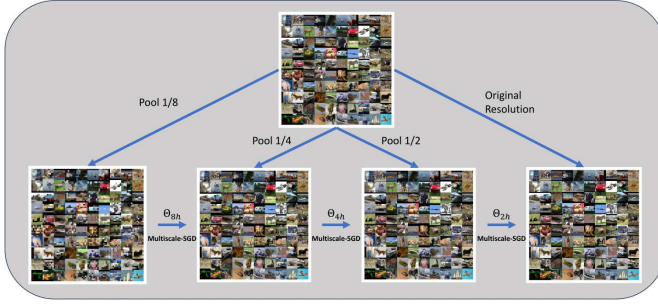


FIGURE 2. Illustration of the Full Multiscale Training.

why, consider the estimation of the gradient obtained via the average over  $\mathbf{g}$  with a batch of  $N$  samples:

$$(4) \quad \mathbb{E}_{\mathbf{u}^h, \mathbf{y}^h} [\mathbf{g}(\mathbf{u}^h, \mathbf{y}^h, \boldsymbol{\theta})] \approx \frac{1}{N} \sum_i \mathbf{g}(\mathbf{u}_i^h, \mathbf{y}_i^h, \boldsymbol{\theta}_k).$$

This approximation results in an error in the gradient. Under some mild assumptions on the sampling of the gradient value (see [19]), the error can be bounded by:

$$(5) \quad \left\| \mathbb{E} [\mathbf{g}(\mathbf{u}^h, \mathbf{y}^h, \boldsymbol{\theta}_k)] - \frac{1}{N} \sum_i \mathbf{g}(\mathbf{u}_i^h, \mathbf{y}_i^h, \boldsymbol{\theta}_k) \right\|_2 \leq \frac{C}{\sqrt{N}},$$

where  $C$  is some constant. Clearly, obtaining an accurate evaluation of the gradient (that is, with low variance) requires sampling  $\mathbf{g}$  across many data points with sufficiently high resolution  $h$ . This tradeoff between the sample size  $N$  and the accuracy of the gradient estimation, is the costly part of training a deep network on high-resolution data. To alleviate the problem, it is common to use large batches, effectively enlarging the sample size. It is also possible to use variance reduction techniques [2, 7, 1]. Nonetheless, for high-resolution images, or 3D inputs, the large memory requirement limits the size of the batch. However, a small batch size can result in noisy, highly inaccurate gradients, and slow convergence [25].

**2.1. Efficient training with multiscale stochastic gradient descent.** To reduce the cost of the computation of the gradient, we use a classical trick proposed in the context of Multilevel Monte Carlo methods [16]. To this end, let  $h = h_1 < h_2 < \dots < h_L$  be a sequence of mesh step sizes, in which the functions  $u$  and  $y$  are discretized on. We can easily sample (or coarsen)  $u$  and  $y$  to some mesh  $h_j, 1 \leq j \leq L$ . We consider the following identity, based on the telescopic sum and the linearity of the expectation:

$$(6) \quad \mathbb{E} [\mathbf{g}^{h_1}(\boldsymbol{\theta})] = \mathbb{E} [\mathbf{g}^{h_L}(\boldsymbol{\theta})] + \mathbb{E} [\mathbf{g}^{h_{L-1}}(\boldsymbol{\theta}) - \mathbf{g}^{h_L}(\boldsymbol{\theta})] + \dots + \mathbb{E} [\mathbf{g}^{h_1}(\boldsymbol{\theta}) - \mathbf{g}^{h_2}(\boldsymbol{\theta})],$$

where for shorthand we define the gradient of  $\boldsymbol{\theta}$  with resolution  $h_j$  by  $\mathbf{g}^{h_j}(\boldsymbol{\theta}) = \mathbf{g}(\mathbf{u}^{h_j}, \mathbf{y}^{h_j}, \boldsymbol{\theta})$ .

The core idea of our Multiscale Stochastic Gradient Descent (Multiscale-SGD) approach, is that *the expected value of each term in the telescopic sum is approximated using a different batch of data with a different batch size*. This concept is demonstrated in Figure 1 and it can be summarized as:

$$(7) \quad \mathbb{E} [\mathbf{g}^{h_1}(\boldsymbol{\theta})] \approx \frac{1}{N_L} \sum_i \mathbf{g}_i^{h_L}(\boldsymbol{\theta}) + \frac{1}{N_{L-1}} \sum_i \left( \mathbf{g}_i^{h_{L-1}}(\boldsymbol{\theta}) - \mathbf{g}_i^{h_L}(\boldsymbol{\theta}) \right) + \dots \\ + \frac{1}{N_1} \sum_i \left( \mathbf{g}_i^{h_1}(\boldsymbol{\theta}) - \mathbf{g}_i^{h_2}(\boldsymbol{\theta}) \right).$$

To understand why this concept is beneficial, we analyze the error obtained by sampling each term in (7). Evaluating the first term in the sum requires evaluating the function  $\mathbf{g}$  on the coarsest mesh (i.e., lowest resolution) using downsampled inputs. Therefore, it can be efficiently computed, while utilizing a large batch size,  $N_L$ . Thus, following (5), the approximation error of the first term in the (7) can be bounded by:

$$(8) \quad \left\| \mathbb{E} [\mathbf{g}^{h_L}(\boldsymbol{\theta})] - \frac{1}{N_L} \sum_i \mathbf{g}_i^{h_L}(\boldsymbol{\theta}) \right\|^2 \leq \frac{C}{\sqrt{N_L}}.$$

Following this step, we need to evaluate the terms of the form

$$(9) \quad \mathbf{r}_j = \mathbb{E} [\mathbf{g}^{h_{j-1}}(\boldsymbol{\theta}) - \mathbf{g}^{h_j}(\boldsymbol{\theta})],$$

Similarly, this step can be computed by resampling, with a batch size  $N_j$ :

$$(10) \quad \hat{\mathbf{r}}_j = \frac{1}{N_j} \sum_i \left( \mathbf{g}_i^{h_{j-1}}(\boldsymbol{\theta}) - \mathbf{g}_i^{h_j}(\boldsymbol{\theta}) \right),$$

for  $j = 1, \dots, L-1$ . The key question is: what is the error in approximating  $\mathbf{r}_j$  by the finite sample estimate  $\hat{\mathbf{r}}_j$ ? Previously, we focused on error due to sample size. However, note that the exact term  $\mathbf{r}_j$  is computed by evaluating  $\mathbf{g}$  on two resolutions of the same samples and subtracting the results.

If the evaluation of  $\mathbf{g}$  on different resolutions yields similar results, then  $\mathbf{g}$  computed on mesh with step size  $h_j$  can be utilized to approximate the gradient  $\mathbf{g}$  on mesh a mesh with finer resolution  $h_{j-1}$ , making the approximation error  $\hat{\mathbf{r}}_j$  small. Furthermore, assume that

$$(11) \quad \|\mathbf{g}_i^{h_{j-1}}(\boldsymbol{\theta}) - \mathbf{g}_i^{h_j}(\boldsymbol{\theta})\| \leq B h_j^p \quad p > 0,$$

for some constant  $B > 0$  and  $p > 0$ , both independent of the pixel-size  $h_j$ . Then, we can bound the error of approximating  $\mathbf{r}_j$  by  $\hat{\mathbf{r}}_j$ , as follows:

$$(12) \quad \|\mathbf{r}_j - \hat{\mathbf{r}}_j\| \leq BC \frac{h_j^p}{\sqrt{N_j}}.$$

Note that, under the assumption that (11) holds, the gradient approximation error between different resolutions decreases as the resolution increases (i.e.,  $h \rightarrow 0$ ). Indeed, sum of the gradient approximation error between subsequent resolutions



**Algorithm 1** Multiscale Stochastic Gradient Descent (Multiscale-SGD)

---

Set batch size to  $N_L$  and sample,  $N_L$  samples of  $\mathbf{u}^{h_1}$  and  $\mathbf{y}^{h_1}$   
 Pool  $\mathbf{u}^{h_L} = \mathbf{R}_{h_L}^{h_1} \mathbf{u}^{h_1}$ ,  $\mathbf{y}^{h_L} = \mathbf{R}_{h_L}^{h_1} \mathbf{y}^{h_1}$   
 Set  $loss = \ell(\mathbf{u}^{h_L}, \mathbf{y}^{h_L}, \boldsymbol{\theta})$   
**for**  $j = 1, \dots, L$  (in parallel) **do**  
   Set batch size to  $N_j$  and sample,  $N_j$  samples of  $\mathbf{u}^{h_1}$  and  $\mathbf{y}^{h_1}$   
   Pool  $\mathbf{u}^{h_j} = \mathbf{R}_{h_j}^{h_1} \mathbf{u}^{h_1}$ ,  $\mathbf{y}^{h_j} = \mathbf{R}_{h_j}^{h_1} \mathbf{y}^{h_1}$  and  $\mathbf{u}^{h_{j-1}} = \mathbf{R}_{h_{j-1}}^{h_1} \mathbf{u}^{h_1}$ ,  $\mathbf{y}^{h_{j-1}} = \mathbf{R}_{h_{j-1}}^{h_1} \mathbf{y}^{h_1}$   
   Compute the losses  $\ell(\mathbf{u}^{h_j}, \mathbf{y}^{h_j}, \boldsymbol{\theta})$  and  $\ell(\mathbf{u}^{h_{j-1}}, \mathbf{y}^{h_{j-1}}, \boldsymbol{\theta})$   
    $loss \leftarrow loss - \ell(\mathbf{u}^{h_j}, \mathbf{y}^{h_j}, \boldsymbol{\theta}) + \ell(\mathbf{u}^{h_{j-1}}, \mathbf{y}^{h_{j-1}}, \boldsymbol{\theta})$   
**end for**

---

(where each is defined in (9)), where the approximation is obtained from the telescopic sum in (7), can be bounded by:

$$(13) \quad e = C \left( \frac{1}{\sqrt{N_L}} + B \sum_{j=1}^{L-1} \frac{h_j^p}{\sqrt{N_j}} \right).$$

Let us look at an exemplary case, where  $p = 1$  and  $h_j = 2h_{j-1}$ , i.e., the resolution on each dimension increases by a factor of 2 between input representations. In this case, the sampling error on the coarsest mesh contributes  $N_L^{-1/2}$ . It then follows that, it is also possible to have the same order of error by choosing  $N_{L-1} = N_L/4$ . That is, to obtain the same order of error at subsequent levels, only 1/4 of the samples are required at the coarser grid compared to the finer one.

Following our Multiscale-SGD approach in (7), the sample size needed on the finest mesh is reduced by a factor of  $4^L$  from the original  $N_L$  while maintaining the same error order, leading to significant computational savings.

Beyond these savings, Multiscale-SGD is easy to implement. It simply requires computing the loss at different input scales and batches, which can be done in parallel. Since gradients are linear, the loss gradient naturally yields Multiscale-SGD. The full algorithm is outlined in (1).

## REFERENCES

- [1] Guillaume Alain, Alex Lamb, Chinnadhurai Sankar, Aaron Courville, and Yoshua Bengio. Variance reduction in sgd by distributed importance sampling. *arXiv preprint arXiv:1511.06481*, 2015.
- [2] Oron Anschel, Nir Baram, and Nahum Shimkin. Averaged-DQN: Variance reduction and stabilization for deep reinforcement learning. In *International Conference on Machine Learning*, pp. 176–185. PMLR, 2017.
- [3] André Araujo, Wade Norris, and Jack Sim. Computing receptive fields of convolutional neural networks. *Distill*, 4(11):e21, 2019.
- [4] A. Borzi. On the convergence of the mg/opt method. Number 5, December 2005.
- [5] W. Briggs, V. Henson, and S. McCormick. *A Multigrid Tutorial, Second Edition*. Society for Industrial and Applied Mathematics, second edition, 2000. doi: 10.1137/1.9780898719505. URL <https://epubs.siam.org/doi/abs/10.1137/1.9780898719505>.

- [6] Neil K. Chada, Ajay Jasra, Kody J. H. Law, and Sumeetpal S. Singh. Multi-level Bayesian deep neural networks. *arXiv preprint arXiv:2203.12961*, 2022. URL <https://arxiv.org/abs/2203.12961>.
- [7] Jianfei Chen, Jun Zhu, and Le Song. Stochastic training of graph convolutional networks with variance reduction. *arXiv preprint arXiv:1710.10568*, 2017.
- [8] Lei Ding, Jing Zhang, and Lorenzo Bruzzone. Semantic segmentation of large-size VHR remote sensing images using a two-stage multiscale training architecture. *IEEE Transactions on Geoscience and Remote Sensing*, 58(8):5367–5376, 2020.
- [9] Moshe Eliasof, Benjamin J. Bodner, and Eran Treister. Haar wavelet feature compression for quantized graph convolutional networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2023a.
- [10] Moshe Eliasof, Jonathan Ephrath, Lars Ruthotto, and Eran Treister. MGIC: Multigrid-in-channels neural network architectures. *SIAM Journal on Scientific Computing*, 45(3):S307–S328, 2023b.
- [11] Elizar Elizar, Mohd Asyraf Zulkifley, Rusdha Muharar, Mohd Hairi Mohd Zaman, and Seri Mastura Mustaza. A review on multiscale-deep-learning applications. *Sensors*, 22(19):7384, 2022.
- [12] Shahaf E. Finder, Yair Zohav, Maor Ashkenazi, and Eran Treister. Wavelet feature maps compression for image-to-image CNNs. *Advances in Neural Information Processing Systems*, 35:20592–20606, 2022.
- [13] Shin Fujieda, Kohei Takayama, and Toshiya Hachisuka. Wavelet convolutional neural networks. *arXiv preprint arXiv:1805.08620*, 2018.
- [14] Velappa Ganapathy and Kok Leong Liew. Handwritten character recognition using multiscale neural network training technique. *International Journal of Computer and Information Engineering*, 2(3):638–643, 2008.
- [15] Michael B. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.
- [16] Michael B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 2015.
- [17] Eldad Haber, Lars Ruthotto, Elliot Holtham, and Seong-Hwan Jun. Learning across scales - A multiscale method for convolution neural networks. abs/1703.02009:1–8, 2017, URL <http://arxiv.org/abs/1703.02009>.
- [18] Juncai He and Jinchao Xu. Mgnet: A unified framework of multigrid and convolutional neural network. *Science China Mathematics*, 62:1331–1354, 2019.
- [19] Adam M. Johansen, Ludger Evers, and N. Whiteley. Monte Carlo methods. *International Encyclopedia of Education*, pp. 296–303, 2010.
- [20] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020a.
- [21] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020b.
- [22] S.G. Nash. A multigrid approach to discretized optimization problems. *Optimization Methods and Software*, 14:99–116, 2000.
- [23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-net: Convolutional networks for biomedical image segmentation*. *CoRR*, abs/1505.04597, 2015. URL <http://arxiv.org/abs/1505.04597>.
- [24] C. B. Scott and Eric Mjolsness. Multilevel artificial neural network training for spatially correlated learning. *SIAM Journal on Scientific Computing*, 41(5):S297–S320, 2019.
- [25] A. Shapiro, D. Dentcheva, and D. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, Philadelphia, 2009.
- [26] Yuyang Shi and Rob Cornish. On multilevel Monte Carlo unbiased gradient estimation for deep latent variable models. In *International Conference on Artificial Intelligence and Statistics*, pp. 3925–3933. PMLR, 2021.

- [27] U. Trottenberg, C. Oosterlee, and A. Schuller. *Multigrid*. Academic Press, 2001.
- [28] Andreas Van Barel and Stefan Vandewalle. Robust optimization of pdes with random coefficients using a multilevel Monte Carlo method. *SIAM/ASA Journal on Uncertainty Quantification*, 7(1):174–202, 2019.
- [29] Antonia van Betteray, Matthias Rottmann, and Karsten Kahl. Mgiad: Multigrid in all dimensions. Efficiency and robustness by weight sharing and coarsening in resolution and channel dimensions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1292–1301, 2023.
- [30] Yating Wang, Siu Wun Cheung, Eric T. Chung, Yalchin Efendiev, and Min Wang. Deep multiscale model learning. *Journal of Computational Physics*, 406:109071, 2020.
- [31] Man-Wah Wong. *Introduction to Pseudo-Differential Operators, An*, volume 6. World Scientific Publishing Company, 2014.

## Autoencoder-based global concave optimization for electrical impedance tomography

BASTIAN HARRACH

(joint work with Andrej Brojatsch, Johannes Wagner)

We report on some preliminary work-in-progress results that aim to derive globally convergent reconstruction algorithms for the inverse coefficient problem of Electrical Impedance Tomography (aka the famous Calderón problem) with finitely-many measurements.

**The Calderón problem with finitely-many measurements.** Let  $\Omega \subseteq \mathbb{R}^k$ ,  $k \geq 2$  be a bounded domain with smooth boundary  $\partial\Omega$ . Let  $\sigma \in L_+^\infty(\Omega)$  and let

$$\Lambda(\sigma) : L_\diamond^2(\partial\Omega) \rightarrow L_\diamond^2(\partial\Omega), \quad g \mapsto u_\sigma^{(g)}|_{\partial\Omega}$$

be the Neumann-to-Dirichlet-operator (aka current-to-voltage map) for the EIT equation, i.e.,  $u_\sigma^{(g)} \in H_\diamond^1(\Omega)$  solves

$$\nabla \cdot (\sigma \nabla u_\sigma^{(g)}) = 0 \quad \text{in } \Omega, \quad \sigma \partial_\nu u_\sigma^{(g)}|_{\partial\Omega} = g.$$

It is easily shown that  $\Lambda(\sigma) \in \mathcal{L}(L_\diamond^2(\partial\Omega))$  is a compact and selfadjoint operator.

The inverse problem

$$\text{reconstruct } \sigma \in L_+^\infty(\Omega) \quad \text{from } \Lambda(\sigma) \in \mathcal{L}(L_\diamond^2(\partial\Omega))$$

has become famous under the name *Calderón problem*. It is known to be a highly non-linear and ill-posed problem. To introduce its variant with finitely many measurements, we introduce a pixel partition

$$\overline{\Omega} = \bigcup_{j=1}^n \overline{P_j}$$

where  $P_1, \dots, P_n \subseteq \Omega$  are non-empty, pairwise disjoint subdomains with Lipschitz boundaries. We assume that the conductivity coefficient  $\sigma \in L_+^\infty(\Omega)$  is piecewise constant with respect to this partition, i.e.  $\sigma = \sum_{j=1}^n \sigma_j \chi_{P_j}$ , with  $\sigma_1, \dots, \sigma_n \in \mathbb{R}_+$ , and  $\chi_{P_j}$  denoting the characteristic function on the  $j$ -th pixel. With a slight abuse of notation, we identify a piecewise constant function  $\sigma \in L_+^\infty(\Omega)$  with the vector

$\sigma = (\sigma_1, \dots, \sigma_n)^T \in \mathbb{R}_+^n$ . As a model for finitely-many measurements, we assume that we can measure the symmetric matrix

$$F(\sigma) = \left( \int_{\partial\Omega} g_i \Lambda(\sigma) g_j ds \right)_{i,j=1,\dots,m} \in \mathbb{S}^m \subset \mathbb{R}^{m \times m}$$

for  $m$  given boundary currents  $g_1, \dots, g_m \in L_\diamond^2(\partial\Omega)$ . This corresponds to measuring the Galerkin projection of  $\Lambda(\sigma)$  to the span of  $g_1, \dots, g_m$ . The gap electrode model in EIT can be written in this form by choosing  $g_j$  to be the characteristic function of the  $j$ -th electrode, and more sophisticated electrode models such as the shunt model or the complete electrode model lead to similar properties of  $F$ . We can thus state the *Calderón problem with finitely many measurements*

$$\text{reconstruct } \sigma \in \mathbb{R}_+^n \quad \text{from} \quad F(\sigma) \in \mathbb{S}^m \subset \mathbb{R}^{m \times m}.$$

**Concave data fitting formulation for EIT.** Standard data-fitting formulations for EIT lead to non-convex minimization problems in high dimensions for which globally convergent algorithms may seem completely out-of-reach. However, the recent result [2] shows that it is possible to reformulate the Calderón problem (with sufficiently many measurements and known a-priori bounds on  $\sigma$ ) as a convex semidefinite optimization problem. The reformulation involves an unknown linear cost functional so that its practical implementation is not immediate. We herein use a different (and simpler) approach to formulate the problem as a concave minimization problem over a convex set.

**Lemma 1.** *The following holds:*

(a) *If  $\sigma \in \mathbb{R}_+^n$  fulfills  $F(\sigma) = Y$  then  $\sigma$  minimizes*

$$\text{trace}(Y - F(\sigma)) \rightarrow \min! \quad \text{s.t.} \quad F(\sigma) \preceq Y.$$

(b) *The functional  $\sigma \mapsto \text{trace}(Y - F(\sigma))$  is concave.*

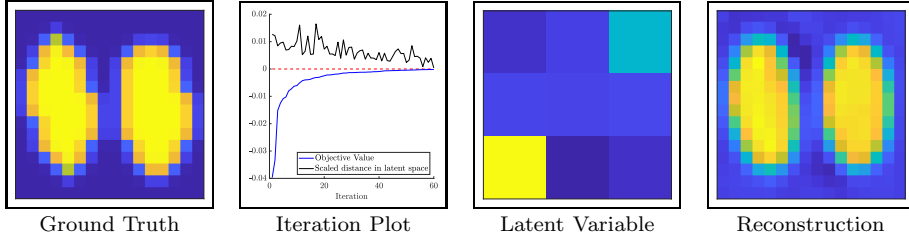
(c) *The constraint set  $\{\sigma \in \mathbb{R}_+^n : F(\sigma) \preceq Y\}$  is convex.*

*Proof.* This follows from the fact that  $F : \mathbb{R}_+^n \rightarrow \mathbb{S}^m$  is monotonically non-increasing and convex with respect to the componentwise ordering “ $\leq$ ” on  $\mathbb{R}^n$  and the Loewner ordering “ $\preceq$ ” on  $\mathbb{S}^m$ , cf. [2, Lemma 4.7].  $\square$

In practical applications one usually also knows a-priori upper and lower bounds of  $\sigma$  so that the constraint set becomes convex and bounded.

**Globally convergent concave programming.** Concave optimization problems over convex bounded sets can be solved with globally convergent algorithms in moderately low dimensions, cf. [3], and [5, Chp. 7.2]. The key idea is that global minima of concave functionals on a polyhedra are attained in a corner. Thus, for a bounded convex set, one starts with a polyhedron containing the constraint set, finds the best corner, i.e. the global minimizer on this superset, and then cuts out the best corner with a hyperplane to shrink the polyhedron. This approach should also yields global convergence for our concave minimization problem in Lemma 1.

FIGURE 1. Concave minimization in the latent space.



**Concavity preserving autoencoder parametrization.** As concave global minimization is numerically feasible in moderately low dimensions, we now aim to describe our unknown conductivities by moderately many parameters. In the work [4], autoencoder techniques were used to find a 16-dimensional latent parametrization of lung images. The key idea is to train neural networks  $\Phi$  and  $\Psi$  so that

$$\Psi \circ \Phi \approx \text{id} \quad \text{on training set of lung images,}$$

where  $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^d$  encodes  $n$ -pixel images with  $d$  latent parameters, and  $\Psi : \mathbb{R}^d \rightarrow \mathbb{R}^n$  decodes  $n$ -pixel images from  $d$  latent parameters. To solve the inverse problem  $F(\sigma) = Y$  for a lung image  $\sigma$ , one then solves  $F(\Psi(p)) = Y$  for  $p \in \mathbb{R}^d$ , and obtains  $\sigma = \Psi(p)$ .

Combining this idea with our concave optimization approach in Lemma 1 we would thus minimize

$$\text{trace}(Y - F(\Psi(p))) \rightarrow \min! \quad \text{s.t.} \quad F(\Psi(p)) \preceq Y.$$

It is easily shown that this is a concave minimization problem on a convex set, if the decoder  $\Psi$  is concave. Since the decoder is a neural network that is a concatenation of linear functions and activator functions, it can be ensured to be convex by enforcing these linear functions and activator functions to be convex and non-decreasing. Training the autoencoder with shifted negated images one can thus construct a concave decoder  $\Psi$ .

Hence, we can reconstruct the conductivity by concave minimization in the low-dimensional latent parameter space. Figure 1 shows a preliminary numerical result for this approach using a 9-dimensional latent variable space and a FEM-implementation of the EIT forward problem with  $m = 31$  electrodes following [1]. The first image shows the true lung image, and the second image the error of the iterations (black line), and the (appropriately scaled) value of the objective functional (blue line). The third image shows the reconstructed latent variable  $p \in \mathbb{R}^9$  as a  $3 \times 3$ -image, and the last image shows the reconstructed lung image  $\sigma = \Psi(p)$ . Note that the objective functional converges monotonically to zero from below as the iterates are the global minimizers of the concave objective functional on a polygonal superset of the constraint set.

## REFERENCES

- [1] B. Harrach, *An introduction to finite element methods for inverse coefficient problems in elliptic PDEs*, Jahresber. Dtsch. Math. Ver. **123**(3), 183–210, 2021.
- [2] B. Harrach, *The Calderón problem with finitely many unknowns is equivalent to convex semidefinite optimization*, SIAM J. Math. Anal. **55**(5), 5666–5684, 2023.
- [3] K. L. Hoffman, *A method for globally minimizing concave functions over convex sets*, Math. Program. **20**, 22–32, 1981.
- [4] J. K. Seo, K. C. Kim, A. Jargal, K. Lee, B. Harrach, *A learning-based method for solving ill-posed nonlinear inverse problems: A simulation study of lung EIT*, SIAM J. Imaging Sci. **12**(3), 1275–1295, 2019.
- [5] H. Tuy, *Convex analysis and global optimization. 2nd edition.*, Springer Optim. Appl. **110**, Cham: Springer, 2016.

## Learned iterative reconstructions in photoacoustic tomography for the acoustic and optical problem

ANDREAS HAUPTMANN

(joint work with Simon Arridge, Anssi Manninen, Ozan Öktem,  
Carola-Bibiane Schönlieb)

### 1. LEARNED RECONSTRUCTIONS

We consider the general form of the underlying operator equation

$$(1) \quad A(f) = g,$$

for the inverse problem with  $f \in X$  and  $g \in Y$ . Here, the forward operator  $A: X \rightarrow Y$  can be either linear, for the acoustic problem, or the nonlinear for the optical problem. For the modelling of the forward problem in photoacoustic tomography we refer to [3].

In the following we concentrate on the inverse problem, which can be understood as formulating a reconstruction operator  $\mathcal{R}: Y \rightarrow X$ . Such a reconstruction operator should be ideally stable and provide a good estimate of the original signal  $f$  for given data  $g$ , i.e.,  $\mathcal{R}(g) \approx f$ . Classically, such a reconstruction operator would be handcrafted based on the analytical knowledge of the forward operator, or formulated in the variational framework as optimisation problem.

In recent years, the paradigm of data-driven reconstructions has gathered considerable attention, due to its success in improving reconstruction quality, but also computational speed-up. Nevertheless, the majority of such data-driven approaches still comes without a thorough mathematical understanding. While we can not solve this shortcoming, we will provide a conceptual overview of data-driven approaches in the following. For that, let first us define the concept of a learned reconstruction operator.

**Definition 1** (Learned reconstruction operator). *A family of mappings  $\mathcal{R}_\theta: Y \rightarrow X$  parametrised by  $\theta \in \Theta$  is called a learned reconstruction operator for the inverse problem in (1) if the parameters  $\theta$  are determined (learned) from example data (training data) that is generated in a way that is consistent with (1).*

With this general definition we can obtain a large class of different popular data-driven approaches. For instance, in its simplest form we can define *two-step* approaches by using a neural network  $\Lambda_\theta$  either in image or data space:

$$\begin{aligned}\mathcal{R}_\theta &= \Lambda_\theta \circ \mathcal{R}, \text{ with } \Lambda_\theta : X \rightarrow X && \text{(post-processing)} \\ \mathcal{R}_\theta &= \mathcal{R} \circ \Lambda_\theta, \text{ with } \Lambda_\theta : Y \rightarrow Y && \text{(pre-processing)}\end{aligned}$$

where  $\mathcal{R} : Y \rightarrow X$  is a hand-crafted reconstruction operator. Such two-step approaches are popular due to their simplicity and success in mildly ill-posed problems, but provide in its general form little methodological insights.

In contrast, a methodologically well motivated approach are *learned regulariser*. Here, the neural network  $\Lambda_\theta : X \rightarrow \mathbb{R}$  replaces a regulariser in the variational formulation. The learned reconstruction operator is then given as

$$\mathcal{R}_\theta(g) = \arg \min_{f \in X} \mathcal{D}(A(f), g) + \Lambda_\theta(f).$$

In this secondary case, one can impose various constraints on  $\Lambda_\theta$  to obtain convergence results, see [4] for a review.

## 2. LEARNED ITERATIVE RECONSTRUCTIONS

Let us now turn to the concept of learned iterative reconstructions, in which hand-crafted model and data-driven components are intertwined. More concretely, neural networks are sequentially combined with the evaluation of the forward operator, its adjoint in the linear case, or the Fréchet derivative for the nonlinear case.

Thus, in learned iterative reconstruction we aim to learn a reconstruction operator defined by the result of the iterative procedure, i.e.,  $\mathcal{R}_\theta(g) = f^N$ , where  $\theta = \{\theta_0, \dots, \theta_{N-1}\}$  and

$$(2) \quad f^{k+1} = \Lambda_{\theta_k}(f^k, h), \text{ with } k = 0, \dots, N-1,$$

with  $\Lambda_{\theta_k} : X \times X \rightarrow X$ . In the above formulation, the neural network combines a current iterative  $f^k \in X$  with a model-informed update direction  $h \in X$ . This update direction should contain model parameters and ideally information on “how to improve reconstructions”. For linear inverse problems this leads to the usual choice of  $h = A^*(Af - g) = \nabla_{\frac{1}{2}} \|Af - g\|_2^2$  and defines a *learned gradient scheme* [1].

**2.1. Learned Gauss-Newton.** Clearly the formulation in (2) is not limited to gradient type directions. Thus, we will now shortly discuss choices of the update directions  $h$  motivated by Newton type approaches which are often used for nonlinear inverse problems.

We remind that for the unregularised least-squares problem, the Newton updates with Hessian  $H_k$  and Jacobian  $J_k$  are given by

$$f^{k+1} = f^k - \beta_k H_k^{-1} (J_k^T (A(f^k) - g)),$$

this would be a straight-forward extension of the learned gradient descent before. Unfortunately, in inverse problems the Hessian may be ill-conditioned and on

top for high-dimensional problems expensive to compute. Thus, we consider the regularised version of the cost functional

$$\mathcal{E}(f) = \|A(f) - g\|_2^2 + \alpha \mathcal{R}(f).$$

Additionally, in practice a full Newton method is rarely used and rather Quasi-Newton or Newton-type approaches, such as the Gauss-Newton where we use an approximation of the Hessian as  $H \approx (J^T J)$ . This leads to the update directions  $h = (J_k^T J_k + \Gamma_{\mathcal{R}}^{-1})^{-1} \nabla \mathcal{E}(f^k)$ , with  $\Gamma_{\mathcal{R}}$  the covariance matrix of the regulariser, and defines a *Learned Gauss-Newton* [2]:

$$f^{k+1} = \Lambda_{\theta_k}(f^k, (J_k^T J_k + \Gamma_{\mathcal{R}})^{-1} \nabla \mathcal{E}(f^k)).$$

Additionally, we can explore the impact of different choices for the update direction. In the following we will explore the 3 choices for the optical problem:

- i.) Gradient descent:  $h = \nabla \mathcal{E}(f)$
- ii.) Gauss-Newton:  $h = (J^T J + \Gamma_{\mathcal{R}})^{-1} \nabla \mathcal{E}(f)$
- iii.) Quasi-Newton:  $h = B^T \nabla \mathcal{E}(f)$

For the Quasi-Newton update we choose the symmetric-rank 1 (SR1) updates.

**2.2. Training schemes.** Finally, a major computational challenge is the training of the learned reconstruction operator  $\mathcal{R}_{\theta}$ . Ideally, we would like to train end-to-end given supervised training data  $\{f_i, g_i\}$  by minimising

$$\theta^* = \arg \min_{\theta \in \Theta} \sum_i L(\mathcal{R}_{\theta}(g_i), f_i).$$

Nevertheless, this requires evaluation of the update directions  $h$ , including, the forward model  $A$  and computation of  $A^*$  or  $J$  in each forward as well as backward pass  $N$ -times for each training iterations, which will be well in ten thousands.

To avoid the computational demanding evaluation in each training iteration we can make use of a greedy training approach that has been earlier utilised for training in the linear acoustic problem [5], where the forward operator takes 12 seconds to evaluate in 3D. Instead of the above loss function, we now only compute iterate-wise optimality, thus the terminology greedy, by

$$\theta_k^* = \arg \min_{\theta \in \Theta} \sum_i L(\Lambda_{\theta}(f_i^k, h_i^k) - f_i),$$

where  $f_i^k = \Lambda_{\theta_{k-1}^*}(f_i^{k-1}, h_i^{k-1})$  and  $h_i^k$  are pre-computed. This allows decoupling of the network training from the evaluation of the model components and enables efficient training even for computationally demanding inverse problems.

### 3. PERFORMANCE COMPARISON FOR CHOICE OF UPDATE DIRECTIONS

A major question that remains is how the choice of update directions influences the performance of the learned reconstruction operator. Additionally, how greedy vs. end-to-end influences this performance. In ongoing work we explore this question in the context of the nonlinear optical problem, a preliminary result of the network performance can be seen in 1. This indicates that for the Gauss-Newton directions that there is only little difference between end-to-end and greedy training, while



for the less informative directions end-to-end training is necessary to obtain good performance.

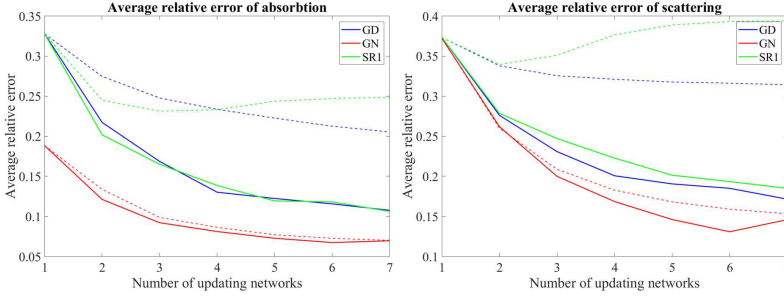


FIGURE 1. Performance comparison in reconstruction quality for absorption and scattering for different update directions and training schemes in the nonlinear optical problem.

## REFERENCES

- [1] J. Adler and O. Öktem, *Solving ill-posed inverse problems using iterative deep neural networks*, Inverse problems (2017), vol. 33(12), p. 124007.
- [2] M. Mozumder, A. Hauptmann, I. Nissilä, S. R. Arridge and T. Tarvainen, *A model-based iterative learning approach for diffuse optical tomography*, IEEE Transactions on Medical Imaging (2021), vol. 41(5), pp. 1289–1299.
- [3] A. Hauptmann and T. Tarvainen, *Model-based reconstructions for quantitative imaging in photoacoustic tomography*, Biomedical Photoacoustics: Technology and Applications (2024), pp. 133–153.
- [4] S. Mukherjee, A. Hauptmann, O. Öktem, M. Pereyra and C.-B. Schönlieb, *Learned reconstruction methods with convergence guarantees: A survey of concepts and applications*, IEEE Signal Processing Magazine (2023), vol. 40(1), pp. 164–182.
- [5] A. Hauptmann, F. Lucka, M. Betcke, N. Huynh, J. Adler, B. Cox, P. Beard and S. Ourselin, *Model-based learning for accelerated, limited-view 3-D photoacoustic tomography*, IEEE transactions on medical imaging (2018), vol. 37(6), pp. 1382–1393.

## Solving ill-posed inverse problems using iResNets

NICK HEILENKÖTTER

(joint work with Clemens Arndt, Alexander Denker, Sören Dittmer, Meira Iske, Tobias Kluth, Peter Maaß, Judith Nickel)

In the context of Bayesian inverse problems, the goal is to reconstruct an unknown source  $x \in X = \mathbb{R}^{d_x}$  from noisy measurements

$$y^\delta = \hat{A}x + \hat{\eta},$$

where both data and noise are assumed to follow (potentially unknown) probability distributions. Here, the measurement process is modeled by a linear operator

$\hat{A} : X \rightarrow Y$ ,  $Y = \mathbb{R}^{d_Y}$ . As usual in the Bayesian setting, we assume  $x \sim p_X$  and  $\hat{\eta} \sim p_H$ , where  $p_X, p_H : X \rightarrow \mathbb{R}_{\geq 0}$  are probability density functions with existing first and second moments. Furthermore,  $x$  and  $\eta$  are required to be stochastically independent, with zero-mean noise:  $\mathbb{E}(\hat{\eta}) = 0$ .

Since inverse problems are often ill-posed, incorporating data distributions into the reconstruction process is crucial. As a result, deep-learning-based approaches have shown promising results among many applications [3]. A method that ensures stability of the reconstructions is to employ invertible neural networks  $\varphi_\theta : X \rightarrow X$ . To this end, we modify the inverse problem and solve the normal equation

$$z^\delta = \hat{A}^* y^\delta = \hat{A}^*(\hat{A}x + \hat{\eta}) = Ax + \eta,$$

where we define  $A := \hat{A}^* \hat{A}$  and  $\eta = \hat{A}^* \hat{\eta}$ . Reconstruction is performed by first optimizing  $\varphi_\theta$  such that  $\varphi_\theta \approx A$ , followed by applying the inverse provided by the architecture of the neural network to obtain the estimate  $x^* = \varphi_\theta^{-1}(z^\delta)$ .

The optimization task for approximating the forward operator is given by

$$\varphi_\theta^* = \arg \min_{\varphi_\theta \in \mathcal{F}} \mathbb{E}_{(x, z^\delta)} (\ell[\varphi_\theta](x, z^\delta)).$$

This allows to choose a network architecture, determined by the inductive bias  $\mathcal{F}$ , as well as the loss function  $\ell$ . In recent work [1, 2], we have studied a variety of possible choices and their properties as regularization methods.

In [1], we have investigated *approximation training*, i.e. a MSE loss that forces the network to approximate the forward operator  $A$ :

$$\begin{aligned} \varphi_\theta^* &= \arg \min_{\varphi_\theta \in \mathcal{F}} \mathbb{E}_{x \sim p_X, \eta \sim p_H} (\|\varphi_\theta(x) - (Ax + \eta)\|^2) \\ &= \arg \min_{\varphi_\theta \in \mathcal{F}} \mathbb{E}_{x \sim p_X, \eta \sim p_H} (\|\varphi_\theta(x) - Ax\|^2). \end{aligned}$$

We notice that the training outcome is independent of the noise level and equals  $A$  whenever the network is able to approximate all linear operators. As we are interested in invertible architectures, we study a set of simple networks that follow the iResNet [4] approach in [1]. For restricted  $\mathcal{F}$ , the learned reconstruction scheme becomes data-dependent. However, in the case of sufficiently expressive architectures, the solution depends only on the first or at most second moments of the prior  $p_X$ .

A second approach that increases data-dependence was studied in [2]. By *reconstruction training*, we denote the objective

$$\begin{aligned} \varphi_\theta^* &= \arg \min_{\varphi_\theta \in \mathcal{F}} \mathbb{E}_{(x, z^\delta)} (\|\varphi_\theta^{-1}(z^\delta) - x\|^2) \\ &= \arg \min_{\varphi_\theta \in \mathcal{F}} \mathbb{E}_{(x, z^\delta)} (\|\varphi_\theta^{-1}(z^\delta) - \mathbb{E}_{x \sim p_X}(x|z^\delta)\|^2). \end{aligned}$$

Here, the optimal solution is the posterior expectation  $\mathbb{E}_{x \sim p_X}(x|z^\delta)$ , provided it can be approximated by the network. The resulting reconstruction method depends on the noise as well as the data distribution. These properties were also analyzed theoretically and numerically in [2].

A third and new approach is to study regularization terms of the approximation training to enhance data-dependency. More specifically, we assume Gaussian noise,

i.e.  $\eta \sim \mathcal{N}(0, \delta^2)$ , and study the (vector-field) divergence of the network as a regularization term:

$$\begin{aligned}\varphi_\theta^* &= \arg \min_{\varphi_\theta \in \mathcal{F}} \mathbb{E}_{x \sim p_X, \eta \sim p_H} (\|\varphi_\theta(x) - (Ax + \eta)\|^2 - \delta^2 \nabla \cdot \varphi_\theta(x)) \\ &= \arg \min_{\varphi_\theta \in \mathcal{F}} \mathbb{E}_{x \sim p_X, \eta \sim p_H} (\|\varphi_\theta(x) - (Ax - \delta^2 \nabla(\log \circ p_X)(x))\|^2).\end{aligned}$$

As a result, the network approximates the regularized normal equation that arises for the maximum-a-posteriori (MAP) estimator. If the MAP estimator is invertible and can be approximated, this implies that the inverse of the network learns the MAP, thus serving as a data-dependent regularization. Moreover, the noise level introduces an additional regularization parameter, while the training data noise does not influence the outcome.

## REFERENCES

- [1] C. Arndt, A. Denker, S. Dittmer, N. Heilenkötter, M. Iske, T. Kluth, P. Maaß, J. Nickel *Invertible residual networks in the context of regularization theory for linear inverse problems*, Inverse Problems 39 (2023), 125018
- [2] C. Arndt, S. Dittmer, N. Heilenkötter, M. Iske, T. Kluth, J. Nickel *Bayesian view on the training of invertible residual networks for solving linear inverse problems*, Inverse Problems 40 (2024), 045021
- [3] S. Arridge, P. Maaß, C.-B. Schönlieb *Deep Learning for Inverse Problems* Oberwolfach Reports (2022) vol. 18, p. 745–789
- [4] J. Behrmann, W. Grathwohl, R. T. Q. Chen, D. Duvenaud, J.-H. Jacobsen *Invertible Residual Networks* Proceedings of the 36th International Conference on Machine Learning (2019), vol 97, p. 573–582

## Fast kernel summation via slicing and Fourier transforms

JOHANNES HERTRICH

(joint work with Tim Jahn, Michael Quellmalz)

Let  $K: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$  be a radial kernel defined by  $K(x, y) = F(\|x - y\|)$  for some basis function  $f: \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ . For given points  $x_1, \dots, x_N \in \mathbb{R}^d$  and  $y_1, \dots, y_M \in \mathbb{R}^d$  and weights  $w_1, \dots, w_N \in \mathbb{R}$ , we are interested in computing the kernel sums

$$(1) \quad s_m = \sum_{n=1}^N w_n K(x_n, y_m), \quad m = 1, \dots, M.$$

Generally, the evaluation of  $s_m$  for all  $m = 1, \dots, M$  involves  $MN$  summands and has therefore computational complexity of  $\mathcal{O}(MN)$ . In the following, we want to approximate this sum to reduce the complexity to  $\mathcal{O}(M + N)$ . In dimension  $d \leq 3$ , this problem can be efficiently solved by fast Fourier summations [7] based on the non-equispaced fast Fourier transform [2] or by fast multipole methods [3]. For the negative distance kernel, we presented in [4, 6] an efficient sorting algorithm to compute the  $s_m$  with complexity  $\mathcal{O}((M + N) \log(M + N))$ . However, the runtime of these algorithms depends exponentially on the dimension, such that they are

computationally intractable for higher dimensions, which we addressed in [4, 5] as follows.

**Fast kernel summation via slicing.** In order to compute the  $s_m$ , from (1) efficiently, we proceed in three steps. First, we choose a random direction  $\xi$  from the uniform distribution  $\mathcal{U}_{\mathbb{S}^{d-1}}$  on the unit sphere  $\mathbb{S}^{d-1}$  and project the whole problem onto the subspace spanned by  $\xi$ . Second, we apply a fast method for the one-dimensional case to solve the projected problems. And third, we take the expectation over all possible directions  $\xi$ . From a mathematical viewpoint this corresponds to considering a kernel  $K$  of the form

$$(2) \quad K(x, y) = \mathbb{E}_{\xi \sim \mathcal{U}_{\mathbb{S}^{d-1}}} [\mathbf{k}(\langle \xi, x \rangle, \langle \xi, y \rangle)]$$

for some one-dimensional kernel  $\mathbf{k}: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ . Then, the kernel sum from (1) can be rewritten and approximated as

$$(3) \quad s_m = \mathbb{E}_{\xi \sim \mathcal{U}_{\mathbb{S}^{d-1}}} \left[ \sum_{n=1}^N w_n \mathbf{k}(\langle \xi, x \rangle, \langle \xi, y \rangle) \right] \approx \frac{1}{P} \sum_{p=1}^P \sum_{n=1}^N w_n \mathbf{k}(\langle \xi_p, x \rangle, \langle \xi_p, y \rangle),$$

where  $\xi_1, \dots, \xi_P$  are iid samples from  $\mathcal{U}_{\mathbb{S}^{d-1}}$ . Consequently, we can approximate  $s_m$  by computing  $P$  one-dimensional kernel sums instead of one  $d$ -dimensional. Using the fast algorithms for the one-dimensional kernel summation this has computational complexity  $\mathcal{O}(P(M + N))$ .

**Sliced kernels.** For applying the slicing algorithm for some given kernel  $K$ , we have to find a one-dimensional kernel  $\mathbf{k}$  such that (2) is fulfilled. For radial kernels  $K(x, y) = F(\|x - y\|)$  and  $\mathbf{k}(x, y) = f(|x - y|)$ , this can be rewritten in terms of the basis functions as

$$(4) \quad F(\|x\|) = \mathbb{E}_{\xi \sim \mathcal{U}_{\mathbb{S}^{d-1}}} [f(|x|)].$$

Then, we proved in [4], that a pair  $(F, f)$  fulfills the relation (4) if and only if  $F$  is the *generalized Riemann-Liouville fractional integral transform* of  $f$  given by

$$F(t) = \frac{2\Gamma(\frac{d}{2})}{\sqrt{\pi}\Gamma(\frac{d-1}{2})} \int_0^1 f(ts)(1-s^2)^{\frac{d-3}{2}} ds.$$

Considering that the functions  $f(x) = x^r$  are eigenfunctions of this integral transform, we obtain that for analytic basis functions of the form  $F(x) = \sum_{n=0}^{\infty} a_n x^n$  for  $x \geq 0$ , the function  $f(x) = \sum_{n=0}^{\infty} b_n x^n$  with  $b_n = \frac{\sqrt{\pi}\Gamma(\frac{n+d}{2})}{\Gamma(\frac{d}{2})\Gamma(\frac{n+1}{2})} a_n$  fulfills (4). In this way, the one-dimensional basis function  $f$  and corresponding kernel  $\mathbf{k}$  can be derived for many common kernels like Gauss, Laplace, Matérn and negative distance kernel.

An alternative way for deriving  $\mathbf{k}$  was outlined in [9] via a change of variables in the Fourier space. Assuming that all involved terms exist, we have that the functions  $G(x) = F(\|x\|)$  and  $g(x) = f(|x|)$  are related by  $g = c_d \mathcal{F}_1[x^{d-1} \mathcal{F}_d^{-1}[G]]$  for some dimension-dependent constant  $c_d$ , where  $\mathcal{F}_1$  and  $\mathcal{F}_d$  are the one- and  $d$ -dimensional Fourier transform. The existence assumptions are fulfilled, e.g.,

for continuous, positive definite kernels. This also links the slicing approach to random Fourier features [8], see [9] for details.

**Slicing error.** In (3), we approximated the expectation over all directions by a finite sum. We now want to bound the error which is introduced by this approximation. To this end, we note that by Bienaymé’s identity it holds that

$$\mathbb{E}_{\xi_1, \dots, \xi_P \sim \mathcal{U}_{\mathbb{S}^{d-1}}} \left[ \left( \frac{1}{P} \sum_{p=1}^P f(|\langle \xi_p, x \rangle|) - F(\|x\|) \right)^2 \right] = \frac{\mathbb{V}_d[f](x)}{P},$$

where  $\mathbb{V}_d[f](x) := \mathbb{E}_{\xi \sim \mathcal{U}_{\mathbb{S}^{d-1}}} \left[ (f(|\langle \xi, x \rangle|) - F(\|x\|))^2 \right]$  is the variance of  $f$ . Then, we proved several bounds on  $\mathbb{V}_d[f](x)$  in [4, 5].

- If  $K(x, y) = F(\|x - y\|)$  is continuous and positive definite with  $F(0) = 1$  and  $f$  is the corresponding one-dimensional basis function, then it holds that  $\mathbb{V}_d[f](x) \leq 1 - F(\|x\|)^2 \leq 1$ .
- For  $K(x, y) = -\|x - y\|$ , then  $\mathbb{V}_d[f](x) \leq \frac{\pi}{2}\|x\|$ .

In both cases, the mean squared error decays with a rate of  $\mathcal{O}(P^{-1})$ , i.e., the absolute error decays with rate  $\mathcal{O}(\frac{1}{\sqrt{P}})$ . It is always independent of  $d$ , but for the negative distance kernel it depends on  $\|x\|$ . For details, we refer to [5, Thm 1].

**QMC slicing.** So far, we assumed that the directions  $\xi_1, \dots, \xi_P$  are random iid samples from  $\mathcal{U}_{\mathbb{S}^{d-1}}$ . In order to improve the convergence error rate of  $\mathcal{O}(\frac{1}{\sqrt{P}})$ , we now drop this assumption and choose them by quasi-Monte Carlo designs. Following [1], we mainly consider the directions maximizing the pairwise distance  $\sum_{p,q=1}^P \|\xi_p - \xi_q\|$ . Under certain smoothness assumptions, the authors of [1] show that this gives a worst case error of  $\mathcal{O}(P^{\frac{-d}{2(d-1)}})$ . We verify these smoothness assumptions and consider other QMC designs in [5]. While we numerically observe better error rates, a formal proof of this remains an open question.

**Numerical comparison.** Finally, we compare our (QMC) slicing approach numerically with random Fourier features [8]. To this end, we use the MNIST dataset with  $N = M = 60000$  data points, which is reduced to  $d = 20$  dimensions by PCA. We choose the  $(x_n)_{n=1}^N$  and  $(y_m)_{m=1}^M$  both to be the whole dataset. As kernel, we use the Gauss, Laplace and negative distance kernel and set the weights  $w_n$  to one. Then, we compute the relative  $L^1$ -error of the corresponding kernel sums and visualize the results in Figure 1. The computation time is measured on a single CPU thread. We observe that for the Gaussian kernel, the slicing and RFF show a similar performance, while for the Laplace kernel, slicing shows a smaller error. This can be explained that for smooth kernels the Fourier transform decays faster such that very few features in RFF are sufficient, while for the non-smooth Laplace kernel more features are required to achieve a certain accuracy. For the negative distance kernel, RFFs are not applicable, since it is not positive definite. In all cases QMC slicing reduces the error significantly. Additional numerical comparisons are provided in [4, 5].

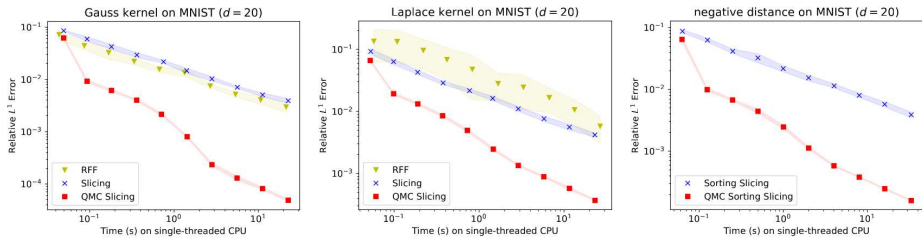


FIGURE 1. Comparison (error vs time) of RFF with (QMC) slicing with  $N = M = 60000$  and  $d = 20$ .

## REFERENCES

- [1] J. Brauchart, E. Saff, I. Sloan, and R. Womersley, *QMC designs: optimal order quasi Monte Carlo integration schemes on the sphere*, Mathematics of Computation, vol. 83(290), pp. 2821–2851, 2014.
- [2] A. Dutt and V. Rokhlin, *Fast Fourier transforms for nonequispaced data*, SIAM Journal on Scientific Computing, vol. 14(6), pp. 1368–1393, 1993.
- [3] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, Journal on Computational Physics, vol. 73(2), pp. 325–348, 1987.
- [4] J. Hertrich, *Fast Kernel Summation in High Dimensions via Slicing and Fourier Transforms*, SIAM Journal on Mathematics of Data Science, vol. 6(4), pp. 1109–1137, 2024.
- [5] J. Hertrich, T. Jahn and M. Quellmalz, *Fast Summation of Radial Kernels via QMC Slicing*, arXiv preprint arXiv:2410.01316, 2024.
- [6] J. Hertrich, C. Wald, F. Altekürger and P. Hagemann, *Generative Sliced MMD flows with Riesz kernels*, International Conference on Learning Representations, 2024.
- [7] D. Potts and G. Steidl, *Fast summation at nonequispaced knots by NFFT*, SIAM Journal on Scientific Computing, vol. 24(6), pp. 2013–2037, 2003.
- [8] A. Rahimi and B. Recht, *Random features for large-scale kernel machines*, Advances in Neural Information Processing Systems, vol. 20, 2007.
- [9] N. Rux, M. Quellmalz, G. Steidl *Slicing of radial functions: a dimension walk in the Fourier space*, arXiv preprint arXiv:2408.11612, 2024.

## Conductivity imaging using deep neural networks

BANGTI JIN

(joint work with Zhi Zhou)

Parameter identifications for differential equations represent a wide class of inverse problems. In terms of mathematical theory, many deep results on conditional stability have been established for concrete PDE parameter identification problems. Meanwhile, a large variety of inversion schemes have been proposed, often based on Tikhonov regularization. Thus, it is natural to ask whether one can use conditional stability results to analyze relevant numerical procedures. Conditional stability has been employed to derive convergence rates for Tikhonov regularization.

We aim to exploit “stability” results for deriving convergence rates for a discrete scheme for recovering a spatially dependent diffusion coefficient  $q$ . Let  $\Omega \subset \mathbb{R}^d$

( $d = 1, 2, 3$ ) be a convex polyhedral domain with a boundary  $\partial\Omega$ . Consider the following elliptic problem:

$$(1) \quad \begin{cases} -\nabla \cdot (q \nabla u) = f, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega, \end{cases}$$

where the function  $f$  denotes a given source term. The solution to problem (1) is denoted by  $u(q)$ . The inverse problem is to recover the exact diffusion coefficient  $q^\dagger$  from the pointwise observation  $z^\delta$ , with a noise level  $\delta$ ,  $\|z^\delta - u(q^\dagger)\|_{L^2(\Omega)} \leq \delta$ . The diffusion coefficient  $q$  is assumed to satisfy  $q \in \mathcal{A} := \{q : 0 < c_0 \leq q \leq c_1 \text{ in } \Omega\}$ . The goal is to develop robust discretization schemes for the inverse problems with provable error bounds.

### 1. FEM DISCRETIZATION

Now we describe one inversion scheme based on Tikhonov regularization and Galerkin FEM approximation. Let  $\mathcal{T}_h$  be a shape regular quasi-uniform triangulation of the domain  $\Omega$  into  $d$ -simplexes with a mesh size  $h$ . Over  $\mathcal{T}_h$ , we define two finite element spaces:  $X_h = \{v_h \in H_0^1(\Omega) : v_h|_K \in P_1(K) \ \forall K \in \mathcal{T}_h\}$  and  $V_h = \{v_h \in H^1(\Omega) : v_h|_K \in P_1(K) \ \forall K \in \mathcal{T}_h\}$ , which are used to approximate the state  $u$  and the diffusion coefficient  $q$ , respectively. Now the inversion scheme for problem (1) reads

$$(2) \quad \min_{q_h \in \mathcal{A}_h} J_{\gamma, h}(q_h) = \frac{1}{2} \|u_h(q_h) - z^\delta\|_{L^2(\Omega)}^2 + \frac{\gamma}{2} \|\nabla q_h\|_{L^2(\Omega)}^2,$$

with  $\mathcal{A}_h = \{q_h \in V_h : c_0 \leq q_h(x) \leq c_1 \text{ in } \Omega\}$  and  $u_h(q_h)$  satisfying

$$(3) \quad (q_h \nabla u_h(q_h), \nabla v_h) = (f, v_h), \quad \forall v_h \in X_h.$$

Then in the work [4], the following weighted error estimate was proved using an energy argument, with a novel test function.

**Theorem 1.** *Let the exact diffusion coefficient  $q^\dagger \in H^2(\Omega) \cap W^{1,\infty}(\Omega)$ ,  $u(q^\dagger)$  the solution to problem (1) with  $f \in L^\infty(\Omega)$ , and  $q_h^* \in \mathcal{A}_h$  a minimizer of problem (2)-(3). Then with  $\eta = h^2 + \delta + \gamma^{\frac{1}{2}}$ , there holds*

$$\int_{\Omega} (q^\dagger - q_h^*)^2 (q^\dagger |\nabla u(q^\dagger)|^2 + f u(q^\dagger)) \, dx \leq c(h\gamma^{-\frac{1}{2}}\eta + \min(h + h^{-1}\eta, 1))\gamma^{-\frac{1}{2}}\eta.$$

Theorem 1 allows deriving the standard  $L^2(\Omega)$  bound, under condition (4), which holds for certain problem data.

**Corollary 1.** *Let  $q^\dagger \in H^2(\Omega) \cap W^{1,\infty}(\Omega)$  and  $f \in L^\infty(\Omega)$ , and assume that there exists some  $\beta \geq 0$  such that*

$$(4) \quad (q^\dagger |\nabla u(q^\dagger)|^2 + f u(q^\dagger))(x) \geq c \, \text{dist}(x, \partial\Omega)^\beta \quad \text{a.e. in } \Omega.$$

*Then the approximation  $q_h^*$  satisfies*

$$\|q^\dagger - q_h^*\|_{L^2(\Omega)} \leq c((h\gamma^{-\frac{1}{2}}\eta + \min(h^{-1}\eta, 1))\gamma^{-\frac{1}{2}}\eta)^{\frac{1}{2(1+\beta)}}.$$

*For any  $\delta > 0$ , the choices  $\gamma \sim \delta^2$  and  $h \sim \sqrt{\delta}$  imply  $\|q^\dagger - q_h^*\|_{L^2(\Omega)} \leq c\delta^{\frac{1}{4(1+\beta)}}$ .*

The analysis combines the consistency errors, and the stability estimates for the inverse problem, and fixing the norm gap between different the two types of results using inverse estimates for the finite element space.

## 2. NEURAL NETWORK APPROXIMATION

Deep neural networks (DNNs) represent a powerful tool for approximating functions living in high-dimensional spaces, and have received much attention within the inverse problems community. However, the discretization requires different treatment due to a lack of inverse estimates. We resort to an alternative approach. Upon letting  $\sigma = q\nabla u$ , we recast problem (1) (with the Neumann boundary condition  $q\partial_n u = g$  on  $\partial\Omega$ ) into a first-order system

$$(5) \quad \begin{cases} \sigma = q\nabla u, & \text{in } \Omega, \\ -\nabla \cdot \sigma = f, & \text{in } \Omega, \\ \sigma \cdot \mathbf{n} = g, & \text{on } \partial\Omega. \end{cases}$$

To identify  $q$ , we employ a noisy observation  $z^\delta$  (of the exact data  $u(q^\dagger)$ ) in  $\Omega$ , with  $\delta := \|\nabla(u(q^\dagger) - z^\delta)\|_{L^2(\Omega)}$ . Then we employ a residual minimization scheme based on the first-order system (5), approximate both  $\sigma$  and  $q$  using DNNs, and substitute  $z^\delta$  for the scalar field  $u$ . For  $q$ , we use a DNN function class (of depth  $L_q$  and width  $W_q$ ) with the parametrization  $\mathfrak{P}_{p,\epsilon_q}$  (with  $p \geq 2$  and tolerance  $\epsilon_q$ ). Similarly, for  $\sigma$ , we employ  $d$  identical DNN function classes (of depth  $L_\sigma$  and width  $W_\sigma$ ) with the parametrizations  $\mathfrak{P}_{2,\epsilon_\sigma}$  (with tolerance  $\epsilon_\sigma$ ), and stack them into one DNN.  $\theta$  and  $\kappa$  denote the parameters of DNN approximations to  $q$  and  $\sigma$ , respectively. To enforce the box constraint on  $q$ , we employ a cutoff operator  $P_A$  defined by  $P_A(v) = \min(\max(c_0, v), c_1)$ . Using the least-squares formulation on the equality constraints, we obtain

$$(6) \quad \begin{aligned} J_\gamma(\theta, \kappa) = & \|\sigma_\kappa - P_A(q_\theta)\nabla z^\delta\|_{L^2(\Omega)}^2 + \gamma_\sigma \|\nabla \cdot \sigma_\kappa + f\|_{L^2(\Omega)}^2 \\ & + \gamma_b \|\sigma_\kappa \cdot \mathbf{n} - g\|_{L^2(\partial\Omega)}^2 + \gamma_q \|\nabla q_\theta\|_{L^2(\Omega)}^2. \end{aligned}$$

Then the DNN reconstruction problem reads

$$(7) \quad \min_{(\theta, \kappa) \in (\mathfrak{P}_{p,\epsilon_q}, \mathfrak{P}_{2,\epsilon_\sigma}^{\otimes d})} J_\gamma(\theta, \kappa),$$

where the superscript  $\otimes d$  denotes the  $d$ -fold direct product,  $\gamma_\sigma > 0$ ,  $\gamma_b > 0$  and  $\gamma_q > 0$  are penalty parameters, and  $\gamma = (\gamma_\sigma, \gamma_b, \gamma_q) \in \mathbb{R}_+^3$ .

Let  $X = \{X_j\}_{j=1}^{n_r}$  and  $Y = \{Y_j\}_{j=1}^{n_b}$  be independent and identically distributed (i.i.d.) samples drawn from the uniform distributions  $\mathcal{U}(\Omega)$  and  $\mathcal{U}(\partial\Omega)$ , respectively, where  $n_r$  and  $n_b$  are the numbers of sampling points in  $\Omega$  and on  $\partial\Omega$ ,



respectively. Then the empirical loss  $\widehat{J}_\gamma(\theta, \kappa)$  is given by

$$(8) \quad \widehat{J}_\gamma(\theta, \kappa) =: n_r^{-1} |\Omega| \sum_{j=1}^{n_r} \|\sigma_\kappa(X_j) - P_{\mathcal{A}}(q_\theta(X_j)) \nabla z^\delta(X_j)\|_{\ell^2}^2 \\ + \gamma_\sigma n_r^{-1} |\Omega| \sum_{j=1}^{n_r} (\nabla \cdot \sigma_\kappa(X_j) + f(X_j))^2 \\ + \gamma_b n_b^{-1} |\partial\Omega| \sum_{j=1}^{n_b} (\sigma_\kappa(Y_j) \cdot \mathbf{n} - g(Y_j))^2 + \gamma_q n_r^{-1} |\Omega| \sum_{j=1}^{n_r} \|\nabla q_\theta(X_j)\|_{\ell^2}^2.$$

Then we have the following (weighted)  $L^2(\Omega)$  error bounds on the DNN approximation  $q_\theta^*$  via the population loss  $J_\gamma(\theta, \kappa)$  [5, Theorem 3.3].

**Theorem 2.** *Let  $q^\dagger \in W^{2,p}(\Omega) \cap \mathcal{A}$ ,  $f \in H^1(\Omega) \cap L^\infty(\Omega)$  and  $g \in H^{\frac{3}{2}}(\partial\Omega) \cap L^\infty(\partial\Omega)$ , with  $p = \max(2, d + \nu)$  for small  $\nu > 0$ . For any  $\psi \in H^1(\Omega)$ , there exists a solution  $v_\psi$  of the equation  $\nabla u^\dagger \cdot \nabla v_\psi = \psi$  almost everywhere in  $\Omega$  and it satisfies  $\|v_\psi\|_{H^1(\Omega)} \leq c\|\psi\|_{H^1(\Omega)}$ . Fix small  $\epsilon_q, \epsilon_\sigma > 0$ , and let  $(q_\theta^*, \sigma_\kappa^*)$  be the DNN realization of a minimizer  $(\theta^*, \kappa^*) \in (\mathfrak{P}_{p, \epsilon_q}, \mathfrak{P}_{2, \epsilon_\sigma}^{\otimes d})$  of problem (7). Then with  $\eta := \epsilon_q + (\gamma_\sigma^{\frac{1}{2}} + \gamma_b^{\frac{1}{2}} + 1)\epsilon_\sigma + \delta + \gamma_q^{\frac{1}{2}}$ , there holds*

$$\|q^\dagger - P_{\mathcal{A}}(q_\theta^*)\|_{L^2(\Omega)} \leq c(1 + \gamma_\sigma^{-\frac{1}{4}} + \gamma_b^{-\frac{1}{4}})(1 + \gamma_q^{-\frac{1}{4}}\eta^{\frac{1}{2}})\eta^{\frac{1}{2}}.$$

Now we analyze the error of the approximation  $\widehat{q}_\theta^*$  via the empirical loss  $\widehat{J}_\gamma(\theta, \kappa)$  [5, Theorem 3.5]. The loss  $\widehat{J}_\gamma(\theta, \kappa)$  involves also quadrature errors arising from approximating the integrals via Monte Carlo methods. The key of the analysis is to bound the error  $\sup_{q_\theta \in \mathcal{N}_q, \sigma_\kappa \in \mathcal{N}_\sigma} |J_\gamma(q_\theta, \sigma_\kappa) - \widehat{J}_\gamma(q_\theta, \sigma_\kappa)|$  for suitable DNN function classes  $\mathcal{N}_q$  and  $\mathcal{N}_\sigma$  (corresponding to the sets  $\mathfrak{P}_{p, \epsilon_q}$  and  $\mathfrak{P}_{2, \epsilon_\sigma}^{\otimes d}$ , respectively), which are also known as statistical errors in statistical learning theory. The proof uses PAC-type generalization bounds, which boils down to Rademacher complexity of the DNN function classes  $\mathcal{N}_q$  and  $\mathcal{N}_\sigma$ .

**Theorem 3.** *Let the assumptions in Theorem 2 hold,  $f \in L^\infty(\Omega)$ , and  $z^\delta \in W^{1,\infty}(\Omega)$ . Fix small  $\epsilon_q, \epsilon_\sigma > 0$ , and let  $(\widehat{\theta}^*, \widehat{\kappa}^*) \in (\mathfrak{P}_{p, \epsilon_q}, \mathfrak{P}_{2, \epsilon_\sigma}^{\otimes d})$  be a minimizer of the empirical loss (8), and  $\widehat{q}_\theta^*$  and  $\widehat{\sigma}_\kappa^*$  their NN realizations. Let the bounds  $e_d, e_\sigma, e_b$  and  $e_q$  be quadrature errors for each term. Further define  $\eta$  by  $\eta := e_d + \gamma_\sigma e_\sigma + \gamma_b e_b + \gamma_q e_q + \epsilon_q^2 + (\gamma_\sigma + \gamma_b + 1)\epsilon_\sigma^2 + \delta^2 + \gamma_q$ . Then there holds*

$$\|q^\dagger - P_{\mathcal{A}}(\widehat{q}_\theta^*)\|_{L^2(\Omega)} \\ \leq c((e_d + \eta)^{\frac{1}{2}} + (e_\sigma + \gamma_\sigma^{-1}\eta)^{\frac{1}{2}} + (e_b + \gamma_b^{-1}\eta)^{\frac{1}{2}} + (e_q + \gamma_q^{-1}\eta)^{\frac{1}{2}}\delta)^{\frac{1}{2}}(e_q + \gamma_q^{-1}\eta)^{\frac{1}{4}}.$$

The numerical experiments in two and high-dimension show that the approach is very promising for noisy data, e.g., up to 10% noise in the data, and has recently been extended to anisotropic conductivity [1]. The idea of using conditional stability for error analysis has also been successfully employed for the hybrid DNN-FEM discretization scheme [2] and point source identification [3], and thus appear

more broadly applicable to nonlinear PDE inverse problems for which conditional stability results are available.

## REFERENCES

- [1] S. Cen, B. Jin, X. Li, Z. Zhou. *Imaging anisotropic conductivity from internal measurements with mixed least-squares deep neural networks*. Preprint, 2024.
- [2] S. Cen, B. Jin, Q. Quan, Z. Zhou. *Hybrid neural-network FEM approximation of diffusion coefficient in elliptic and parabolic problems*. IMA J. Numer. Anal. 2024;44(5): 3059–3093.
- [3] T. Hu, B. Jin, Z. Zhou. *Point source identification using singularity enriched neural networks*. Preprint, arXiv:2408.09143, 2024.
- [4] B. Jin, Z. Zhou, *Error analysis of finite element approximations of diffusion coefficient identification for elliptic and parabolic problems*. SIAM J. Numer. Anal. 2021;59(1):119–142.
- [5] B. Jin, X. Li, Q. Quan, Z. Zhou. *Conductivity imaging from internal measurements with mixed least-squares deep neural networks*. SIAM J. Imaging Sci. 2024;17(1): 147–187.

## Lipschitz duality in shallow neural networks

YURY KOROLEV

(joint work with Francesca Bartolucci, Marcello Carioni, José Iglesias,  
Emanuele Naldi, Stefano Vigogna)

Universal approximation theorems like [8, 14] state that any continuous function on a compact set  $K \subset \mathbb{R}^d$  can be approximated arbitrarily closely by a neural network with a single hidden layer and a non-polynomial activation function; in other words, universal approximation theorems are density results. This makes such networks a reasonable function class to approximate continuous functions.

However, universal approximation alone cannot explain empirical success of neural networks in approximating functions in very high dimensions ( $d \gg 1$ ). Many other function classes have their own versions of universal approximation, which may come, e.g., in the form of the Stone-Weierstrass theorem. For example, [15] gives the following, very general version.

**Theorem 1** ([15]). *Let  $K$  be compact and  $B$  be a linear subspace of  $C(K)$  that contains constant functions and separates points of  $K$ . If  $h \in C(\mathbb{R})$  is a non-affine function that operates on  $B$  (that is, if  $f \in B$  then  $h \circ f \in B$ ) then  $\overline{B} = C(K)$ .*

For  $h(t) = t^2$  we recover the “usual” Stone-Weierstrass theorem, for  $h(t) = |t|$  its “lattice version”.

While universal approximation doesn’t make neural networks stand out from other approximators, what does is that, for functions in certain subspaces of  $C(K)$ , they can achieve approximation rates that are independent of the dimension of  $K$ . The first result in this direction is due to Barron

**Theorem 2** ([4]). *Let  $f: [0, 1]^d \rightarrow \mathbb{R}$  be a function and  $\hat{f}(\xi)$  its Fourier transform. Suppose that  $\exists C > 0$  such that*

$$\int |\xi| \left| \hat{f}(\xi) \right| d\xi < C.$$

Suppose that  $\sigma: \mathbb{R} \rightarrow \mathbb{R}$  is sigmoidal. Then for any  $n \in \mathbb{N}$  there exists a neural network

$$(1) \quad f_n(x) := \sum_{i=1}^n a_i \sigma(\langle \omega_i, x \rangle + b_i), \quad \omega_i \in \mathbb{R}^d, a_i, b_i \in \mathbb{R},$$

such that

$$\|f - f_n\|_{L^2} \leq \frac{2C}{\sqrt{n}}.$$

The coefficients  $\{a_i\}_{i=1}^n$  can be chosen to satisfy  $\sum_{i=1}^n |a_i| \leq 2C$ .

While the above theorem gives a sufficient condition for dimension-independent approximation rates, there are larger spaces admitting such rates, notably the variation norm spaces [12, 13, 3, 2]. In particular, [2] proposed to parametrise a shallow neural network using a Radon measure over the parameter space rather than the parameter values themselves. The resulting construction, a so-called infinite-width shallow neural network, is given by

$$(2) \quad f_\mu(x) = \int_{\Theta} \sigma(\langle w, x \rangle + b) d\mu(w, b), \quad x \in \mathbb{R}^d,$$

where  $\mu \in \mathcal{M}(\Theta)$  is a signed Radon measure on the parameter space  $\Theta = \{(w, b) \in \mathbb{R}^{d+1} : w \in \mathbb{R}^d, b \in \mathbb{R}\}$ . By choosing the discrete measure  $\mu = \sum_{i=1}^N a_i \delta_{(\omega_i, b_i)}$  in (2), one recovers a shallow neural network with  $N$  neurons (1). Moreover, by identifying  $x$  with  $(x, 1)$  one can consider  $(w, b)$  as a single variable and write

$$(3) \quad f_\mu(x) = \int_{\Theta} \sigma(\langle \theta, x \rangle) d\mu(\theta), \quad x \in \mathbb{R}^d \times \{1\} \subset \mathbb{R}^{d+1}.$$

The space of infinite-width shallow neural networks was called the Barron space in [9]. The integral in (3) can be interpreted as a duality pairing

$$(4) \quad f_\mu(x) = \langle \mu, \sigma(\langle \cdot, x \rangle) \rangle,$$

where the meaning of the brackets depends on the parameter space  $\Theta$ . If this space is compact then the pairing can be understood as that between continuous functions  $C(\Theta)$  and finite Radon measures  $\mathcal{M}(\Theta)$ . If the space is only locally compact (as is often the case with neural networks), [6] proposed to use a measurable weighting function  $\beta: \Theta \rightarrow \mathbb{R}$  such that  $\sup_{\theta \in \Theta} |\rho(x, \theta) \beta(\theta)| = D_x < \infty$  for all  $x \in X$  and interpret the pairing (4) as that between continuous functions vanishing at infinity  $C_0(\Theta)$  and finite Radon measures  $\mathcal{M}(\Theta)$ :

$$f_\mu(x) = \mathcal{M}(\Theta) \langle \mu, \sigma(\langle \cdot, x \rangle) \rangle_{C_0(\Theta)} = \int_{\Theta} \sigma(\langle \theta, x \rangle) \beta(\theta) d\mu(\theta).$$

At this point, the weighting  $\beta$  seems to be artificial and a technical artefact.

We propose another interpretation of the pairing (4) in the case when the activation function  $\sigma$  is Lipschitz (which is true for all activation functions of practical interest), which is a version of the famous Kantorovich-Rubinstein duality. Let  $\text{Lip}(\Theta)$  denote the space of Lipschitz functions on  $\Theta$ . Choose a fixed base point  $e \in \Theta$  and set

$$(5) \quad \|f\|_{\text{Lip}} := \max \{|f(e)|, L(f)\},$$

where  $L(f)$  is the Lipschitz constant. Lipschitz spaces are always dual spaces and in many cases the predual is unique [16, Sect. 3.4]. This predual goes by several names, e.g. the Arens-Eells space [1] or the Lipschitz-free space [10]; it also has connections to (and roots in) optimal transport.

Denote by  $\mathcal{M}(\Theta)$  the space of finite Radon measures over  $\Theta$  and by  $\mathcal{M}^0(\Theta)$  the subspace of balanced measures, i.e. those with  $\mu(\Theta) = 0$ . Let  $\mathcal{M}_p(\Theta)$ , with  $p \in (0, +\infty)$ , be the subspace of measures in  $\mathcal{M}(\Theta)$  with finite  $p$ -moments. For  $p = 1$ , an alternative norm on  $\mathcal{M}_1(\Theta)$  can be defined as follows [7, Sect. 8.10(viii)]

$$(6) \quad \|\mu\|_{\text{KRu}} := |\mu(\Theta)| + \sup \left\{ \int_{\Theta} f \, d\mu(\theta) : f(e) = 0, L(f) \leq 1 \right\}.$$

For balanced measures the norm (6) is known as the Kantorovich-Rubinstein norm [11, Sec. VIII.4] denoted by  $\|\cdot\|_{\text{KR}}$ . In our notation  $\|\cdot\|_{\text{KRu}}$  we emphasize the fact that the measure can be unbalanced. The space  $\mathcal{M}_1(\Theta)$  is not complete under this norm, and its completion will be denoted by  $\text{KRu}(\Theta)$ . For balanced measures, the completion of the space of balanced measures with a finite first moment (which we denote by  $\mathcal{M}_1^0(\Theta)$ ) in the norm (6) is the more standard Kantorovich-Rubinstein space  $\text{KR}(\Theta)$ . It can be checked that  $\text{Lip}(\Theta)$  with the norm (5) is the dual of  $\text{KRu}(\Theta)$ .

Since many popular activation functions are Lipschitz, it seems reasonable to interpret the pairing in (4) as the dual pairing

$$(7) \quad f_{\mu}(x) = {}_{\text{KRu}(\Theta)} \langle \mu, \sigma(\langle \cdot, x \rangle) \rangle_{\text{Lip}(\Theta)}.$$

In the context of empirical risk minimisation, compactness plays an important role in ensuring the existence of minimisers. It is known that for a compact space  $\Theta$ , the unit ball with respect to the total variation norm is strongly compact in the Kantorovich-Rubinstein space [11, Thm. VIII.4.3]. Moreover, one can show that for a non-compact  $\Theta$  the result holds under certain moment conditions on the measures, which are also necessary to ensure such compactness.

**Theorem 3** ([5, Thm. 20]). *Suppose that  $p > 1$ . Then the following set is compact in  $\text{KRu}(\Theta)$*

$$B^p = \left\{ \mu \in \mathcal{M}(\Theta) : \int_{\Theta} (1 + d(\theta, e)^p) d|\mu|(\theta) \leq 1 \right\},$$

where  $d(\cdot, e)$  denotes the distance to the base point  $e \in \Theta$ .

With this compactness result at hand, we consider, for  $p > 1$ , the following regularised empirical risk minimisation problem

$$(8) \quad \inf_{\mu \in \text{KRu}(\Theta)} \frac{1}{N} \sum_{i=1}^N L(y_i, f_{\mu}(x_i)) + G_{\alpha, \beta}(\mu),$$

where  $\{x_i, y_i\}_{i=1}^N$  are the training data,  $\alpha \geq 0$  and  $\beta > 0$  are parameters and the regularizer  $G_{\alpha, \beta} : \text{KRu}(\Theta) \rightarrow [0, \infty]$  is defined by

$$(9) \quad G_{\alpha, \beta}(\mu) := \begin{cases} \alpha \|\mu\|_{\text{KRu}} + \beta \int_{\Theta} (1 + d(\theta, e)^p) d|\mu|(\theta) & \text{if } \mu \in \mathcal{M}_p(\Theta), \\ +\infty & \text{otherwise.} \end{cases}$$

In [5], we study this problem and prove a representer theorem, showing the existence of solutions of finite width. We study large data limits ( $N \rightarrow \infty$ ) and show strong KRo convergence of the solutions  $\mu_N$ , which implies uniform convergence of the corresponding functions  $f_{\mu_N}$  on bounded sets. We also show how the regulariser can be modified to describe network distillation and fusion. We generalize the framework further using the duality between continuous functions with controlled growth and weighted measures, which allows us to clarify the role of weights used in [6].

## REFERENCES

- [1] R. F. Arens and J. Eells, Jr., “On embedding uniform and topological spaces”, *Pacific J. Math.* 6 (1956), pp. 397–403.
- [2] F. Bach, “Breaking the curse of dimensionality with convex neural networks”, *Journal of Machine Learning Research* 18.19 (2017), pp. 1–53.
- [3] A. R. Barron, A. Cohen, W. Dahmen and R. A. DeVore, “Approximation and learning by greedy algorithms”, *The Annals of Statistics* 36.1 (2008), pp. 64–94.
- [4] A. Barron, “Universal approximation bounds for superpositions of a sigmoidal function”, *IEEE Transactions on Information Theory* 39.3 (1993), pp. 930–945.
- [5] F. Bartolucci, M. Carioni, J. Iglesias, Y. Korolev, E. Naldi and S. Vigogna, “A Lipschitz spaces view of infinitely wide shallow neural networks”, *arXiv:2410.14591* (2024).
- [6] F. Bartolucci, E. De Vito, L. Rosasco and S. Vigogna, “Understanding neural networks with reproducing kernel Banach spaces”, *Applied and Computational Harmonic Analysis* 62 (2023), pp. 194–236.
- [7] V. Bogachev, *Measure Theory*, Vol.2. Springer, 2007.
- [8] G. Cybenko, “Approximation by superpositions of a sigmoidal function”, *Mathematics of Control, Signals and Systems* 2.4 (1989), pp. 303–314.
- [9] W. E, C. Ma and L. Wu, “The Barron Space and the Flow-Induced Function Spaces for Neural Network Models”, *Constructive Approximation* 55.1 (2022), pp. 369–406.
- [10] G. Godefroy and N. J. Kalton, “Lipschitz-free Banach spaces”, *Studia Mathematica* 159 (2003), pp. 121–141.
- [11] L. V. Kantorovich and G. P. Akilov, *Functional analysis*, 2nd ed. Pergamon Press, 1982.
- [12] V. Kurkova and K. Sanguineti, “Bounds on rates of variable-basis and neural-network approximation”, *IEEE Transactions on Information Theory* 47.6 (2001), pp. 2659–2665.
- [13] V. Kurkova and M. Sanguineti, “Comparison of worst case errors in linear and neural network approximation”, *IEEE Transactions on Information Theory* 48.1 (2002), pp. 264–275.
- [14] M. Leshno, V. Ya. Lin, A. Pinkus, S. Schocken, “Multilayer feedforward networks with a nonpolynomial activation function can approximate any function”, *Neural Networks* 6.6 (1993), pp. 861–867.
- [15] Y. Sternfeld, Y. Weit, “An approximation theorem for vector valued functions”, *Geometric Aspects of Functional Analysis: Israel Seminar (GAFA)* 1987–88. Ed. by J. Lindenstrauss and V. D. Milman. Berlin, Heidelberg: Springer Berlin Heidelberg, 1989, pp. 126–137.
- [16] N. Weaver, *Lipschitz algebras*, 2nd ed. World Scientific Publishing, 2018.

## Unveiling the role of the Wasserstein distance in generative modelling

LISA KREUSSER

Generative models have gained remarkable popularity in the machine learning community over the last few years due to their ability to create realistic and complex data in a variety of applications. For instance, they are used to generate

photorealistic images in computer vision, synthesize natural-sounding speech in audio processing, and even produce coherent and contextually relevant text in natural language processing tasks. These models are typically categorized into three major classes: likelihood-based models, such as variational autoencoders (VAEs); implicit models, like generative adversarial networks (GANs); and score-based models, which leverage probabilistic approaches for data generation. In this talk, I shared insights into our recent research advancements focusing on two key areas: (i) Wasserstein generative adversarial networks (WGANs) and (ii) score-based diffusion models, both of which represent significant progress in the domain of generative modelling.

Wasserstein GANs were originally inspired by the idea of minimizing the Wasserstein distance between the real and generated data distributions, which was proposed as a theoretically robust alternative to traditional GAN loss functions. However, through theoretical analysis and empirical experiments, we demonstrate that the WGAN loss fails to provide a meaningful approximation of either the true distributional Wasserstein distance or the batch Wasserstein distance. We argue that the apparent success of WGANs arises precisely from this failure, which prevents the batch Wasserstein distance from being approximated effectively, resulting in unique learning dynamics that contribute to the model's empirical performance.

On the other hand, score-based diffusion models have emerged as one of the most promising approaches in deep generative modelling, showcasing state-of-the-art performance across a wide range of generative tasks. These models are grounded in robust mathematical principles, employing stochastic differential equations (SDEs) and their deterministic counterparts, ordinary differential equations (ODEs), to model the data generation process. Our research systematically explores the relationship between the ODE and SDE dynamics in score-based diffusion models. By linking these dynamics to the Fokker–Planck equation, we derive a theoretical upper bound on the Wasserstein 2-distance between the distributions induced by the ODE and SDE formulations. This bound is expressed in terms of the residual error in the Fokker–Planck equation, providing valuable insights into the approximation quality of each approach.

Furthermore, we conduct numerical experiments to evaluate the practical implications of this theoretical analysis. By incorporating the Fokker–Planck residual as an additional regularization term during training, we observe that the gap between the ODE- and SDE-induced distributions can be significantly reduced. While this regularization improves the quality of distributions generated by the ODE, it comes with a trade-off: the sample quality of the SDE-based distributions may degrade.

Overall, our research provides new theoretical and practical insights into the functioning of WGANs and score-based diffusion models, shedding light on their respective strengths, limitations, and unique properties. These findings contribute to a deeper understanding of generative modelling and pave the way for further advancements in the design of robust and efficient generative frameworks.

## REFERENCES

- [1] T. Deveney, J. Stanczuk, L. M. Kreusser, C. Budd, and C.-B. Schönlieb, “Closing the ODE-SDE gap in score-based diffusion models through the Fokker-Planck equation,” arXiv: 2311.15996.
- [2] J. Stanczuk, C. Etmann, L. M. Kreusser, and C.-B. Schönlieb, “Wasserstein GANs Work Because They Fail (to Approximate the Wasserstein Distance),” arXiv: 2103.01678.

**Efficient diffusion models for MRI reconstruction**

YOLANNE YI RAN LEE

Lumbar spine disorders are a prevalent health issue, motivating research into targeted imaging for treatment planning and guided interventions. Magnetic Resonance Imaging (MRI) captures bone and soft tissue structures without radiation, albeit with prolonged acquisition times. To address the tradeoff between image quality and acquisition efficiency, thick-slice MRI has emerged as a practical solution, offering fast acquisition with high in-plane resolution but suffering from limited through-plane resolution and inconsistent contrast [1, 2]. Recent work in [3] proposes the Latent Space Diffusion Energy-Based Model (LSDEBM), which integrates an energy-based prior into a latent space diffusion process to refine reconstructions from thick-slice data. By leveraging the generative capabilities of diffusion models and employing a latent-space representation, LSDEBM overcomes the computational inefficiencies associated with traditional diffusion models, offering a computationally efficient and high-quality solution for 3D medical imaging.

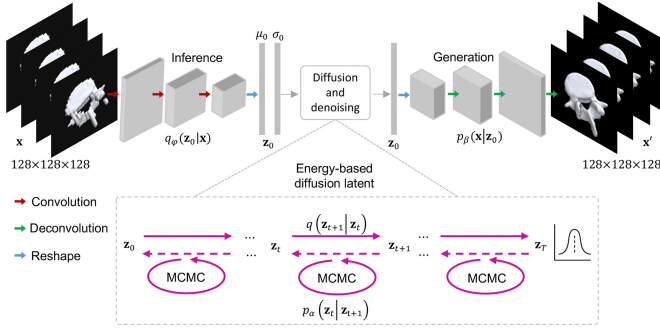


FIGURE 1. The schematic diagram of our network structure and proposed LEBM. The input is encoded into the latent space  $\mathbf{z}$ , where a forward diffusion process is constructed and a reverse process with a conditional energy-prior is learned.  $\mathbf{z}_0$  is then decoded back into the image dimensions.

The architecture of the LSDEBM is visualized in Fig. 1 [3]. An inference network  $\varphi$  uses convolutional layers to encode the image into a latent vector, which is then optimized by the energy-based diffusion and denoising processes. The generation network  $\beta$  then reconstructs the images via a series of deconvolutional layers with

higher resolution from the learned latent. Here, a conditional energy-based prior is applied in the denoising process instead of the approximation by a U-Net [4]. This latent-space formulation reduces computational overhead by constraining the diffusion process to a lower-dimensional representation, facilitating stable and efficient optimization. The reverse diffusion process is realized via Markov Chain Monte Carlo (MCMC) sampling using Langevin dynamics [5, 6]. The energy function, parameterized by a neural network, governs the denoising process, ensuring high fidelity in the reconstructed images.

LSDEBM outperforms VAEs and Latent Energy-Based Models (LEBMs) in reconstructing lumbar vertebrae from low-quality MRI, achieving higher Dice scores and improved performance across various metrics [3]. It matches VAE training times while being 50% faster than LEBMs, addressing the high computational demands of 3D medical imaging efficiently.

Beyond its application to vertebrae modeling, LSDEBM establishes a foundation for leveraging latent space diffusion processes in broader medical imaging tasks. Current work now explores other alternatives to further condition the latent space by leveraging similar concepts from [7]. We propose to use the pseudo-metric tensor,  $Jacobian^T Jacobian$ , to similarly induce sparsity and orthogonality in the latent space. This would encourage a meaningful latent space representation in a similar way that the energy-based model in [3] learns a ‘low energy’ representation with the added interpretation of a more efficient representation thanks to the sparse and orthogonal regularization.

## REFERENCES

- [1] M. P. Laaks, K. Juottonen, K. Partanen, P. Vainio, and H. Soininen, *MRI Volumetry of the Hippocampus: The Effect of Slice Thickness on Volume Formation*, Magnetic resonance imaging (1997), vol. 15(2), pp. 263–265.
- [2] Y. Sui, O. Afacan, C. Jaimes, A. Gholipour, and S. K. Warfield, *Scan-Specific Generative Neural Network for MRI Super-Resolution Reconstruction*, IEEE Transactions on Medical Imaging (2022), vol. 41(6), pp. 1383–1399.
- [3] Y. Wang, Y. Y. R. Lee, A. Dolfini, M. Reischl, E. Konukoglu, and K. Flouris, *Energy-Based Prior Latent Space Diffusion Model for Reconstruction of Lumbar Vertebrae from Thick Slice MRI*, MICCAI Workshop on Deep Generative Models (2024), pp. 22–32, Springer.
- [4] R. Gao, Y. Song, B. Poole, Y. N. Wu, and D. P. Kingma, *Learning Energy-Based Models by Diffusion Recovery Likelihood*, arXiv preprint arXiv:2012.08125 (2020).
- [5] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, *Deep Unsupervised Learning Using Nonequilibrium Thermodynamics*, International Conference on Machine Learning (2015), pp. 2256–2265, PMLR.
- [6] M. Welling, and Y. W. Teh, *Bayesian Learning via Stochastic Gradient Langevin Dynamics*, Proceedings of the 28th International Conference on Machine Learning (ICML-11) (2011), pp. 681–688.
- [7] K. Flouris, and E. Konukoglu, *Canonical Normalizing Flows for Manifold Learning*, Advances in Neural Information Processing Systems (2023), vol. 36, pp. 27294–27314.



## Dynamical systems-based structured networks

DAVIDE MURARI

(joint work with Priscilla Canizares, Carola-Bibiane Schönlieb, Ferdia Sherry, Zakhar Shumaylov)

Structured neural networks inspired by dynamical systems offer a powerful approach to solving problems where preserving qualitative properties, such as conservation laws, is crucial. A notable example, detailed in [1], are Lipschitz-constrained network denoisers, which ensure convergent fixed-point iterations when employed in Plug-and-Play algorithms. Another property to enforce over a network is symplecticity. As we will see in Section 1, this property can significantly improve the long-term stability of a network-based differential equation solver.

This extended abstract presents a procedure to enforce a specific structure over Residual Neural Networks (ResNets). ResNets have gained popularity because they provide a much more stable training process compared to feedforward neural networks. The fundamental layer of a ResNet takes the form

$$(1) \quad \mathbb{R}^d \ni x \mapsto x + \mathcal{F}(x; \theta_i) \in \mathbb{R}^d,$$

where  $\mathcal{F}(\cdot; \theta_i) : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is a vector field depending on some parameters collected in  $\theta_i$ . We can interpret the map in (1) as a single step of size 1 of the explicit Euler method applied to the parametric initial value problem

$$(2) \quad \begin{cases} y'(t) = \mathcal{F}(y(t); \theta_i) \\ y(0) = x. \end{cases}$$

We use the notation  $' = d/dt$  here and throughout this abstract. Based on this insight, we can generalise the standard ResNet architecture with layers as in (1) to encode additional structure. To do so, we can replace the explicit Euler method in (1) with any other one-step numerical method, and the differential equation in (2) can be designed as needed. For example, to get a network preserving the  $\ell^2$ -norm of the input, we could choose  $\mathcal{F}(x; \theta)$  to be a vector field tangent to a sphere and then use a norm-preserving integrator to replace the map in (1). To suitably choose how to replace these two building blocks and develop different ResNet-like architectures, we can rely on the well-developed fields of Dynamical Systems and Geometric Numerical Integration. We now present such an approach to design symplectic neural networks.

### 1. TIME-DEPENDENT SYMPLECTIC NEURAL NETWORKS

We now focus on time-dependent canonical Hamiltonian systems on  $\mathbb{R}^{2d}$ , which can be described with the differential equations

$$(3) \quad x'(t) = J \nabla_x H(x(t), t), \quad J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix},$$

where  $I, 0 \in \mathbb{R}^{d \times d}$  are the identity and zero matrices, respectively. In (3), the function  $H : \mathbb{R}^{2d} \times \mathbb{R} \rightarrow \mathbb{R}$  is the Hamiltonian energy, and the matrix  $J$  is the so-called canonical symplectic matrix. We denote with  $\phi_{H,t} : \mathbb{R}^{2d} \rightarrow \mathbb{R}^{2d}$  the time- $t$

flow map of (3) starting from time 0, i.e.,  $x(t) = \phi_{H,t}(x(0))$ . The map  $\phi_{H,t}$  is symplectic for every  $t \geq 0$ , meaning that

$$(\partial_{x_0} \phi_{H,t}(x_0))^\top J (\partial_{x_0} \phi_{H,t}(x_0)) = J$$

for every  $t \geq 0$  and  $x_0 \in \mathbb{R}^{2d}$ . By the chain rule, it is also easy to verify that composing symplectic maps leads to symplectic maps. These two properties allow us to build time-dependent symplectic neural networks by composing the exact flows of easily solvable parametric Hamiltonian systems.

We consider the Hamiltonian functions  $H_q$  and  $H_p$  defined as

$$(4) \quad H_q(q, p, t; \theta_q) = \partial_t U(q, t; \theta_q) \text{ and } H_p(q, p, t; \theta_p) = \partial_t K(p, t; \theta_p),$$

where the state variable  $x \in \mathbb{R}^{2d}$  is split into  $x = (q, p)$  with  $q, p \in \mathbb{R}^d$ , whereas  $\theta_q$  and  $\theta_p$  are the parameters determining the two energy functions. The equations of motion defined by  $H_q$  and  $H_p$  can be exactly integrated, leading to

$$(5) \quad \begin{aligned} \phi_{H_q,t}(q_0, p_0) &= \begin{pmatrix} q_0 \\ p_0 - (\nabla_q U(q_0, t; \theta_q) - \nabla_q U(q_0, 0; \theta_q)) \end{pmatrix}, \\ \phi_{H_p,t}(q_0, p_0) &= \begin{pmatrix} q_0 + (\nabla_p K(p_0, t; \theta_p) - \nabla_p K(p_0, 0; \theta_p)) \\ p_0 \end{pmatrix}. \end{aligned}$$

We then define a time-dependent symplectic network of  $2L$  layers as

$$(6) \quad \mathcal{N}(x, t; \theta) = \phi_{H_p^L,t} \circ \phi_{H_q^L,t} \circ \cdots \circ \phi_{H_p^1,t} \circ \phi_{H_q^1,t}(x).$$

The superscripts in  $H_q^i$  and  $H_p^i$  indicate that these are layer-specific functions, each with its own parameters. In practice, we define  $H_q^i(q, p, t; \theta_q^i) = \partial_t U(q, t; \theta_q^i)$  and  $H_p^i(q, p, t; \theta_p^i) = \partial_t K(p, t; \theta_p^i)$  where  $U$  and  $K$  are modelled by small neural networks depending respectively on the parameters  $\theta_q^i$  and  $\theta_p^i$ . The parameter  $\theta$ , on which  $\mathcal{N}$  depends, collects all layer-specific weights  $\theta_q^i$  and  $\theta_p^i$ . The network in (6) has the same structure described at the beginning of this abstract. In this specific case, we are in the “lucky” situation where no numerical method is required, as we can directly access the exact solutions of the differential equations defining the network layers. We provide more details on this network architecture, which we call SympFlow, in [2].

A one-step numerical method  $\psi^h$  with step size  $h > 0$  is called symplectic if it preserves the symplectic structure when applied to a Hamiltonian system, i.e., it satisfies  $(\partial_x \psi^h(x))^\top J (\partial_x \psi^h(x)) = J$ . A well-known result in geometric numerical integration (see [3, Section IX.8]) guarantees that a symplectic method approximating the solutions of a time-independent Hamiltonian system of the form  $x'(t) = J \nabla H(x(t))$ , will almost conserve the Hamiltonian energy  $H$  for exponentially long time intervals. Motivated by this result, we aim to design a symplectic neural network that approximates  $\phi_{H,t}$  while preserving the structure of the Hamiltonian system. We consider a time-dependent symplectic neural network as in (6) and train it to approximate  $\phi_{H,t}$  over a compact set  $\Omega \times [0, \Delta t] \subset \mathbb{R}^{2d} \times \mathbb{R}$ . We train the network by minimising the residual between the left and right-hand sides

of the target differential equation, i.e., the loss function

$$L(\theta) = \frac{1}{N} \sum_{i=1}^N \left\| \left. \frac{d}{dt} \mathcal{N}(x_0^i, t; \theta) \right|_{t=t^i} - J \nabla H(\mathcal{N}(x_0^i, t^i; \theta)) \right\|_2^2,$$

for  $N$  sampled initial conditions  $x_0^i \in \Omega$  and time instants  $t^i \in [0, \Delta t]$ ,  $i = 1, \dots, N$ .

We include in Figure 1 the results for the simple harmonic oscillator. In this case we set  $\Omega = [-1, 1]^2$  and  $\Delta t = 1$ . To make predictions up to time  $T = 500$ , we compose the network with itself sufficiently many times, as one would do with the exact flow map  $\phi_{H,t}$ . For example, to predict the state at  $t = \Delta t + \Delta t/2$ , we evaluate  $\mathcal{N}(\mathcal{N}(x, \Delta t; \theta), \Delta t/2; \theta)$ , recursively applying the network over smaller intervals. The figure compares the results obtained with our symplectic network, SympFlow, with a reference Runge–Kutta (5,4) solver and an unstructured ResNet. We see that, similarly to what occurs for numerical methods, the symplectic property improves the long-term energy behaviour. This demonstrates the practical advantage of incorporating symplecticity into the network design, particularly for systems requiring accurate long-term energy conservation. While these results are encouraging, further investigation is needed to extend backward error analysis, commonly used for numerical methods, to neural networks. This could provide deeper insight into the improved energy behaviour observed in our experiments.

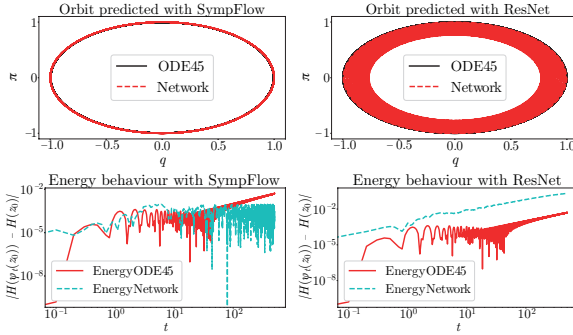


FIGURE 1. Comparison of the predicted orbit and the corresponding energy. The Hamiltonian of the target system is  $H(q, p) = (q^2 + p^2)/2$ , and we integrate up to the final time  $T = 500$ .

## REFERENCES

- [1] F. Sherry, E. Celledoni, M. J. Ehrhardt, D. Murari, B. Owren, and C. B. Schönlieb, (2024). Designing Stable Neural Networks using Convex Analysis and ODEs. *Physica D: Nonlinear Phenomena*, 463, 134159.
- [2] P. Canizares, D. Murari, C. B. Schönlieb, F. Sherry, and Z. Shumaylov, (2024). Hamiltonian Matching for Symplectic Neural Integrators. arXiv preprint arXiv:2410.18262.
- [3] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, Springer, 2006.

## Spatially adaptive Ridge regularizers: Stability and performance

SEBASTIAN NEUMAYER

Inverse problems in imaging amount to reconstructing an (unknown) image  $\mathbf{x} \in \mathbb{R}^d$  from a (noisy) observation  $\mathbf{y} \in \mathbb{R}^m$  determined by the linear relation

$$(1) \quad \mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n},$$

where  $\mathbf{H} \in \mathbb{R}^{m \times d}$  encodes the underlying data acquisition process and the noise  $\mathbf{n} \in \mathbb{R}^m$  accounts for imperfections in this description. As the matrix  $\mathbf{H}$  in equation (1) is often ill-conditioned or even non-invertible in practice, which implies that (1) is ill-posed in the sense of Hadamard, reconstructing  $\mathbf{x}$  from  $\mathbf{y}$  is usually a highly challenging task. In the past years, deep-learning-based approaches have become the state-of-the-art for this inversion task in many applications, see [2] for an overview. Although they are able to achieve impressive results, several concerns regarding their trustworthiness remain. In particular, we should keep in mind that, aside from reconstruction performance in terms of some quality metric, the following properties are usually also desirable:

- universality, i.e., that  $\mathbf{H}$  and the noise model are input parameters of the method;
- data consistency, namely that (1) is approximately satisfied;
- Lipschitz continuity of the data-to-reconstruction map  $\mathbf{y} \mapsto \mathbf{x}_{\text{rec}}$ ;
- interpretability, e.g., in the sense that there exists an explicit underlying model.

Recent works reveal the troublesome issues that potentially arise for deep-learning-based approaches if these principles are not met [1]. On the other hand, we know that these principles are (usually) met by classical variational solutions

$$(2) \quad \hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{R}^d} D(\mathbf{H}\mathbf{x}, \mathbf{y}) + \lambda R(\mathbf{x}),$$

where the data fidelity  $D: \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$  ensures data consistency and the regularizer  $R: \mathbb{R}^d \rightarrow \mathbb{R}$  incorporates prior information about the unknown  $\mathbf{x}$ . Unfortunately, hand-crafted regularizers such as the (convex) Tikhonov [9] or total-variation (TV) [8] ones cannot achieve the same reconstruction quality as data-driven approaches. Recent works attempt to close this performance gap by learning a regularizer  $R$  from data.

A pioneering approach is the *fields of experts* [7], where  $R$  is the sum of nonlinear 1D functions composed with convolutional filters

$$(3) \quad R: \mathbf{x} \mapsto \sum_{j=1}^{N_C} \langle \mathbf{1}_d, \psi_j(\mathbf{W}_j \mathbf{x}) \rangle.$$

In (3), the potentials  $\psi_j: \mathbb{R} \rightarrow \mathbb{R}^+$  are applied pixelwise, the  $\mathbf{W}_j \in \mathbb{R}^{d \times d}$  are convolution matrices, and  $j$  indexes along the  $N_C$  channels. Recently, we proposed an efficient scheme to learn both the filters and the 1D functions based on linear splines [3, 4]. Over the years, it has been observed that the use of a spatially varying regularization mask (as opposed to a constant weight  $\lambda \mathbf{1}_d$ ) can significantly boost

the performance for models of the form (2). More precisely, we modify (3) by using a spatially varying regularization mask  $\mathbf{\Lambda}: \mathbb{R}^m \rightarrow \mathbb{R}_+^d$  that depends on the data  $\mathbf{y}$  according to some heuristic, which leads to

$$(4) \quad R_{\mathbf{y}}: \mathbf{x} \mapsto \sum_{j=1}^{N_C} \langle \mathbf{\Lambda}_j(\mathbf{y}), \psi_j(\mathbf{W}_j \mathbf{x}) \rangle.$$

In Figure 1, we observe that indeed both the noise and the structure in the ground truth  $\mathbf{x}_{\text{gt}}$  lead to high cost. Consequently, to avoid oversmoothing, the values of  $\mathbf{\Lambda}$  should be smaller at structures. In [6], we proposed to compute  $\mathbf{\Lambda}$  based on an unconditional reconstruction  $\mathbf{x}_{\text{est}} \in \mathbb{R}^d$  associated to (2) with regularizer (3), which is then plugged into a mask-generating NN  $G: \mathbb{R}^d \rightarrow [0, 1]^{dN_C}$ . This significantly boosts the performance compared to the unconditional regularizer (3).

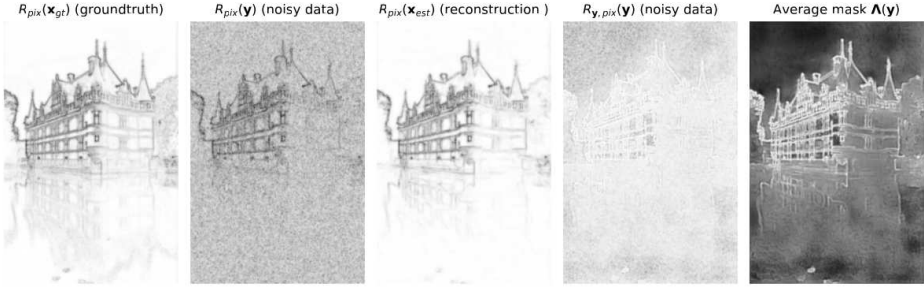


FIGURE 1. The first three images correspond to the pixelwise cost  $R_{\text{pix}}(\mathbf{x})$  for  $\mathbf{x} \in \{\mathbf{x}_{\text{gt}}, \mathbf{y}, \hat{\mathbf{x}}\}$ , respectively, where black corresponds to high values. The last two depict the adapted cost  $R_{\mathbf{y}, \text{pix}}(\mathbf{x}) = \mathbf{\Lambda}(\mathbf{y}) \odot R_{\text{pix}}(\mathbf{x})$ , and the mask.

Further, we studied three classic properties of the reconstruction model (2) and the associated reconstructions  $\hat{\mathbf{x}}$  in [5]:

- We show existence of minimizers based on geometric arguments that do not require the coercivity of  $R$ .
- We prove Lipschitz continuity of the (set-valued) data-to-reconstruction map  $\hat{\mathbf{x}}(\mathbf{y})$  for the convex and data-independent case. If  $\mathcal{R}_{\mathbf{y}}$  is data-dependent, we only obtain a weaker notion of Lipschitz continuity (Aubin property).
- We show that given a solution  $\hat{\mathbf{x}}$  with  $\mathbf{H}\hat{\mathbf{x}} = \mathbf{y}$  and noisy data  $\|\mathbf{y}_{\delta} - \mathbf{y}\| \leq \delta$ , we can find a parameter choice rule  $\lambda: \mathbb{R} \rightarrow \mathbb{R}$  with  $\lambda(\delta) \rightarrow 0$  as  $\delta \rightarrow 0$  such that

$$(5) \quad \lim_{\delta \rightarrow 0} \arg \min_{\mathbf{x} \in \mathbb{R}^d} D(\mathbf{H}\mathbf{x}, \mathbf{y}_{\delta}) + \lambda(\delta) R_{\mathbf{y}_{\delta}}(\mathbf{x}) = \hat{\mathbf{x}}.$$

In words, this means that we approach  $\hat{\mathbf{x}}$  in the low noise regime.

## REFERENCES

- [1] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen. On instabilities of deep learning in image reconstruction and the potential costs of AI. *Proc. Natl. Acad. Sci. U.S.A.*, 117(48):30088–30095, 2020.
- [2] S. Arridge, P. Maaß, O. Öktem, and C.-B. Schönlieb. Solving inverse problems using data-driven models. *Acta Numer.*, 28:1–174, 2019.
- [3] A. Goujon, S. Neumayer, P. Bohra, S. Ducotterd, and M. Unser. A neural-network-based convex regularizer for inverse problems. *IEEE Trans. Comput. Imaging*, 9:781–795, 2023.
- [4] A. Goujon, S. Neumayer, and M. Unser. Learning weakly convex regularizers for convergent image-reconstruction algorithms. *SIAM J. Imaging Sci.*, 17(1):91–115, 2024.
- [5] S. Neumayer and F. Altekürer. Stability of data-dependent ridge-regularization for inverse problems. *arXiv:2406.12289*, 2024.
- [6] S. Neumayer, M. Pourya, A. Goujon, and M. Unser. Boosting weakly convex ridge regularizers with spatial adaptivity. In *NeurIPS Workshop Deep Inverse*, 2023.
- [7] S. Roth and M. J. Black. Fields of experts. *Int. J. Comput. Vis.*, 82(2):205–229, 2009.
- [8] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, 1992.
- [9] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Sov. Math.*, 4:1035–1038, 1963.

## Deep learning meets direct sampling methods for inverse scattering problems

JIANFENG NING

(joint work with Fuqun Han, Jun Zou)

**Problem formulation.** Consider the inverse scattering problems governed by the Helmholtz equation:

$$(1) \quad \Delta u + k^2 n(x)u = 0 \quad \text{in } \mathbb{R}^d, \quad d = 2, 3.$$

where  $u = u^i + u^s$  is the total field.  $u^i$  and  $u^s$  are the incident and scattered fields, respectively. Denote  $u^\infty \in \mathbb{S}^{d-1}$  as the far-field pattern of  $u^s$ , then

$$(2) \quad u^s(x) = \frac{\exp(\mathbf{i}k|x|)}{|x|^{(N-1)/2}} \left\{ u^\infty(\hat{x}) + \mathcal{O}(1/|x|) \right\}, \quad |x| \rightarrow \infty.$$

The inverse scattering problem of our interest involves recovering unknown scatterers, i.e.,  $n(x)$ , from the measurement of  $u^s$  (equivalently  $u$ ) or  $u^\infty$ , corresponding to some incident fields.

**Direct sampling methods (DSM).** The DSMs can provide robust approximations for the shapes and locations of the unknown scatterer. They involve the computation of the indicator functions:

$$(3) \quad \mathcal{I}_{\text{DSM}}(z) := |\langle G(z, x), u^s(x) \rangle_{L^2(\Gamma)}| = \left| \int_{\Gamma} G(z, x) \overline{u^s(x)} ds(x) \right|,$$

$$(4) \quad \mathcal{I}_{\text{DSM}}^\infty(z) := |\langle G^\infty(z, x), u^\infty(x) \rangle_{L^2(\mathbb{S}^{N-1})}| = \left| \int_{\mathbb{S}^{N-1}} G^\infty(z, x) \overline{u^\infty(x)} ds(x) \right|,$$

for  $z \in \Omega$  and  $u^s, u^\infty$  data.

If the indicator functions take a relatively large value at a point  $z$ , the point  $z$  is likely to be within or near the scatterer, otherwise the point  $z$  is likely to be away from the scatterer.

Given phaseless total field data  $|u|$ , we proposed a DSM with the following indicator function[2]:

$$(5) \quad \mathcal{I}_{\text{DSM}}^{\text{phaseless}}(z) := \left| \int_{\Gamma_r} G(z, x_r) \frac{|u(x_r)|^2 - |u^i(x_r)|^2}{u^i(x_r)} ds(x_r) \right|, \quad \forall z \in \Omega,$$

and we have shown the following asymptotic relationships between the above three indicator functions:

$$(6) \quad \mathcal{I}_{\text{DSM}}^{\text{phaseless}}(z) = \mathcal{I}_{\text{DSM}}(z) + \mathcal{O}(R_r^{(1-N)/2}) = C_N \mathcal{I}_{\text{DSM}}^\infty(z) + \mathcal{O}(R_r^{(1-N)/2}).$$

**Learned post processing with DSM.** We proposed a deep learning approach integrating the DSM for inverse scattering problems[1, 2]. With  $N_{in}$  incident fields, we can get their corresponding  $N_{in}$  indicator functions  $\{\mathcal{I}_i\}_{i=1}^{N_{in}}$ . We employ the indicator functions  $\{\mathcal{I}_i\}_{i=1}^{N_{in}}$  as input to a network  $\mathcal{G}_\Theta$ , and the output is a reconstruction of the unknown medium  $n(x)$ . I.e.,

$$(7) \quad n(x) \approx \mathcal{G}_\Theta(\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_{N_{in}}).$$

A very effective network for this problem is the well-designed convolutional network: U-Net. With some properties of the indicator functions, the proposed learning approach has several attractive advantages.

- The computation of the indicator functions is very simple, cheap, and parallel.
- The DSM is very robust to noise, as the high-frequency noise is likely to be smoothed out in the DSM process, which makes the proposed learning scheme also very robust to noise.
- Denote  $\mathcal{I}_{D_1}$  and  $\mathcal{I}_{D_2}$  the indicator functions for scatterer  $D_1$  and  $D_2$ , respectively. Then we have  $\mathcal{I}_{D_1 \cup D_2} = \mathcal{I}_{D_1} + \mathcal{I}_{D_2} + \mathcal{O}(\text{dist}(D_1, D_2)^{(1-N)/2})$ . This means that the relation between the indicator function and the true image is mainly local, which is rather consistent with the local nature of CNNs.
- Based on the asymptotic properties (6), when the radius of the measurement surface is large enough, a network trained with  $u^s$  can also be applied to problems with data  $u^\infty$  or  $|u|$  available.
- The DSM does not require a prior knowledge of the type of the unknown scatterer. The proposed learning approach can solve problems when both penetrable and impenetrable scatterers coexist.

**Learning the probing functions for direct sampling methods for limited-aperture data.** Consider the problem with far-field data  $u^\infty$ , in (4) we employ  $G^\infty(z, \hat{x})$  as the probing function for full-aperture data. However, given limited-aperture data  $u^\infty|_\Gamma$  measured from partial area  $\Gamma$ , directly employing  $G^\infty(z, \hat{x})$  as the probing function can only provide low-resolution reconstruction[3]. Our aim is then to parameterize the desired probing function with a network  $\mathcal{NN}_\Theta(z, \hat{x})$

and train it without any training data by introducing a proper loss function. Specifically, we hope the network can meet the following approximation property for any far-field data  $u^\infty$ :

$$(8) \quad \langle \mathcal{NN}_\vartheta(z, \hat{x}), u^\infty(\hat{x}) \rangle_{L^2(\Gamma)} \approx \langle G^\infty(z, \hat{x}), u^\infty(\hat{x}) \rangle_{L^2(\mathbb{S}^1)}, \quad \forall z \in \Omega.$$

By the linearity of the inner product and the following important expression for  $u^\infty$ :

$$(9) \quad u^\infty(\hat{x}) = \int_{\mathbb{R}^N} G^\infty(y, \hat{x}) I(y) dy, \quad \text{where} \quad I(y) = (n(y) - 1)k^2 u(y),$$

it would be reasonable to require the following in order to meet the approximation property (9) for an appropriately selected integer  $N$ :

$$(10) \quad \langle \mathcal{NN}_\vartheta(z, \hat{x}), \sum_{n=1}^N c_n \exp(-ik\hat{x} \cdot y_n) \rangle_{L^2(\Gamma)} \approx \langle \exp(-ik\hat{x} \cdot z), \sum_{n=1}^N c_n \exp(-ik\hat{x} \cdot y_n) \rangle_{L^2(\mathbb{S}^1)}, \quad \forall y_n, z \in \Omega, \forall c_n \in \mathbb{C}.$$

Thus, we can introduce the following loss function[3]:

$$(11) \quad \text{Loss}(\vartheta) = \frac{1}{ML} \sum_{m=1}^M \sum_{l=1}^L \left| \frac{|\Gamma|}{Q} \sum_{q=1}^Q \mathcal{NN}_\vartheta(z_l, \hat{x}_q) \overline{v_m^\delta(\hat{x}_q)} - 2\pi \sum_{n=1}^N \overline{c_{nm}} J_0(k|z_l - y_{nm}|) \right|^2,$$

where  $v_m(\hat{x}) = \sum_{n=1}^N c_{nm} \exp(-ik\hat{x} \cdot y_{nm})$  and  $v_m^\delta(\hat{x})$  is obtained by polluting  $v_m(\hat{x})$  with noise. In each iteration  $\Re(c_{nm})$  and  $\Im(c_{nm})$  are randomly chosen from the normal distribution,  $\{y_{nm}\}$  are randomly chosen from the uniform distribution of  $\Omega$ .  $L$  is a fixed number and points  $\{z_l\}_{l=1}^L$  are randomly chosen from  $\Omega$  in each iteration.  $\{\hat{x}_q\}_{q=1}^Q$  are some uniformly distributed points on  $\Gamma$ .

It is easy to see the training process does not require any training data. In addition, given a new problem with the same measurement area, we do not need to retrain the network.

We also developed a finite space framework to construct probing functions for limited-aperture data[3]. However, in this non-learning approach, we must carefully choose a regularization parameter, while the proposed deep probing network can escape the need to choose a regularization parameter.

## REFERENCES

- [1] J. Ning, F. Han, and J. Zou, *A direct sampling-based deep learning approach for inverse medium scattering problems*, Inverse Problems (2023), vol. 40(1), pp. 015005.
- [2] J. Ning, H. Fuqun, and J. Zou, *A direct sampling method and its integration with deep learning for inverse scattering problems with phaseless data*, arXiv preprint arXiv:2403.02584 (2024).



- [3] J. Ning, and J. Zou, *Constructing probing functions for direct sampling methods for inverse scattering problems with limited-aperture data: finite space framework and deep probing network*, arXiv preprint arXiv:2410.05098 (2024).

## Learning priors for Bayesian inverse problems in imaging

THOMAS POCK

(joint work with Muhamed Kuric)

Inverse problems in imaging [1] aim to recover unknown signals from observed measurements, often described as  $y = A(x) + \eta$ , where  $y$  are the given measurements,  $A$  is a known forward operator,  $x$  the signal of interest, and  $\eta$  represents noise. The Bayesian framework expresses the posterior distribution of possible image reconstructions using Bayes's theorem as

$$p_{X|Y}(x|y) \propto p_{Y|X}(y|x)p_X(x),$$

where  $p_X(x)$  is the prior,  $p_{Y|X}(y|x)$  is the likelihood, and  $p_{X|Y}(x|y)$  is the posterior. In practice, the likelihood is usually defined through the physics of the imaging process and the noise distribution. For example in MRI, the likelihood is simply a multivariate Gaussian distribution of the form

$$p_{Y|X}(y|x) \propto \exp\left(-\frac{1}{2\sigma^2} \|SFx - y\|^2\right),$$

where  $F$  is the Fourier transform and  $S$  is a sampling operator [2]. However, the image prior  $p_X(x)$  is much harder to model by hand due to the structural complexity of images (natural or medical). Hence, machine learning approaches are needed in order to learn expressive image priors from data.

In this talk, we present a general method for learning priors from data based on marginal feature statistics and the principle of maximum entropy [3]. The idea is to learn a prior  $p_X(\cdot, \theta)$  that maximizes its entropy (and hence is least restrictive) but has the same marginal feature statistics as the training data. This amounts for solving the following constraint optimization problem:

$$\max_p \underbrace{\mathbb{E}_{x \sim p}[-\log p(x)]}_{H[p]}, \quad \mathbb{E}_{x \sim p}[1] = 1, \quad p(x) \geq 0, \quad \mathbb{E}_{x \sim p}[\phi_k(x)] = \mu_k, \quad k = 1, \dots, K,$$

where  $H[p]$  is the entropy,  $\phi_k(x)$  are some features (scalars, vectors or functions) and  $\mu_k$  are the corresponding feature expectations computed on the training data. It turns out that this solution to this problem yields Gibbs-like priors of the form

$$p_X(x; \theta) \propto \exp\left(-\sum_{k=1}^K \langle \theta_k, \phi_k(x) \rangle\right),$$

where  $\theta_k$  are the Lagrangian multiplier functions, which act as the potential functions in the energy of the Gibbs distribution. Interestingly, the Lagrangian dual problem of maximum entropy is equivalent to minimum negative log-likelihood learning.

In initial experiments, we used simple function-valued features  $\phi_k$ , which are obtained as an image histogram of the result of convolution of the image with small convolution kernels  $f_k$ . The kernels  $f_k$  were taken from the basis functions of a 2D discrete cosine transformation of size  $7 \times 7$ . Mathematically, the features  $\phi_k$  are given by

$$\phi_k(x, z) = \frac{1}{P} \sum_{p=1}^P \delta((f_k * x)_p - z), \quad z \in \mathbb{R}$$

where  $p = 1, \dots, P$  are the image pixels.

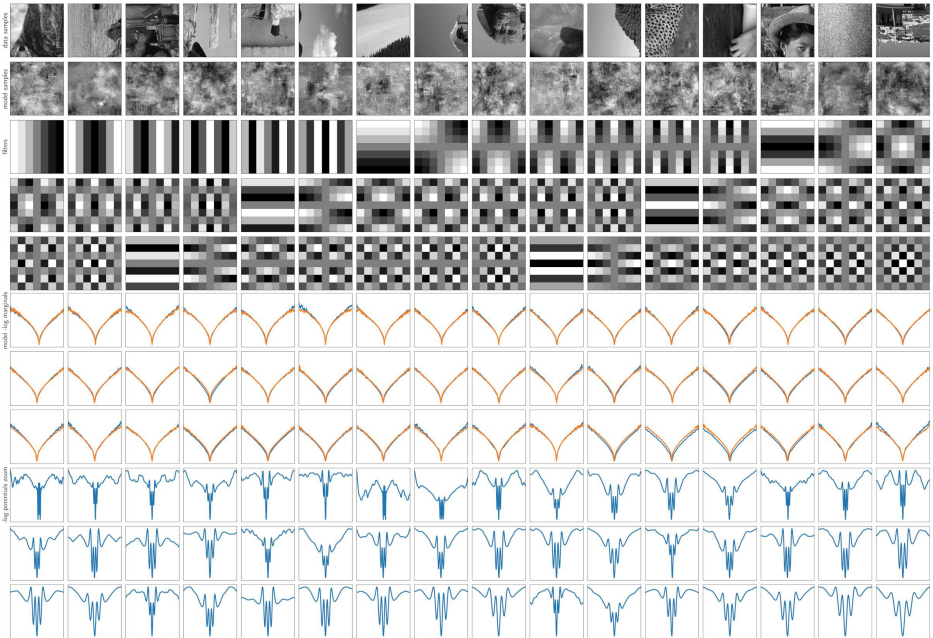


FIGURE 1. The first row shows sampled  $96 \times 96$  patches from a natural image database. The second row presents similar patches sampled from the learned model using a Gibbs-like sampler for approximating the partition function. The next three rows display  $7 \times 7$  convolutional filters. Following this, three rows compare feature histograms from the training data (blue) with those from the learned prior (orange). The final three rows depict the learned Lagrange multiplier functions, which define the energy potential functions in the prior.

Figure 1 shows preliminary results of learning a maximum entropy prior. The first row shows some sample patches of size  $96 \times 96$  extracted from a database of natural images. The second row shows sample patches of the same size extracted from the learned model using a Gibbs-like sampler. These samples are used during learning

to approximate the partition function. The next block of three rows shows the  $7 \times 7$  convolutional filters extracted from the discrete cosine transform. Note that the filters vary between low-frequency and high-frequency filters. The next block of three rows compares the feature histograms of the training data (in blue) with the corresponding histograms computed from the learned prior (in orange). It can be seen that the histograms match very closely, especially for the features with lower frequencies. The last block of three rows shows the shape of the learned Lagrangian multiplier functions that define the potential functions of the energy in the learned prior. It can be observed that the learned functions are much more oscillatory, especially around the zero point, in contrast to the common manually created potential functions.

Future directions will include applying the learned priors for solving inverse problems in imaging, extending this framework to non-linear features extracted from pre-trained neural networks and improving sampling algorithms for broader applicability.

#### REFERENCES

- [1] S. Arridge, P. Maaß, O. Öktem, and C. Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.
- [2] F. Knoll, K. Bredies, T. Pock, and R. Stollberger. Second order total generalized variation (TGV) for MRI. *Magnetic Resonance in Medicine*, 65(2):480–491, 2011.
- [3] S. C. Zhu, Y. N. Wu, and D. Mumford. Minimax entropy principle and its application to texture modeling. *Neural computation*, 9(8):1627–1660, 1997.

#### Training data reconstruction

CHRISTINA RUNKEL

(joint work with Kanchana Vaishnavi Gandikota, Jonas Geiping,  
Carola-Bibiane Schönlieb, Michael Moeller)

#### ABSTRACT

Being able to reconstruct training data from the parameters of a neural network is a major privacy concern. Previous works have shown that reconstructing training data, under certain circumstances, is possible. As further detailed in [1], we analyse such reconstructions empirically and propose a new formulation of the reconstruction as a solution to a bilevel optimisation problem. We demonstrate that our formulation as well as previous approaches highly depend on the initialisation of the training images  $x$  to reconstruct. In particular, we show that a random initialisation of  $x$  can lead to reconstructions that resemble valid training samples while not being part of the actual training dataset. Thus, our experiments on affine and one-hidden layer networks suggest that when reconstructing natural images, yet an adversary cannot identify whether reconstructed images have indeed been part of the set of training samples.

# PROBLEM FORMULATION

The empirical success of modern deep learning heavily relies on training on large datasets which may potentially contain private and sensitive data. The trained models can memorise [2] and replicate samples from training data [3]. Several recent works have focused on evaluating the vulnerabilities of deep networks in leaking information about training data using different attacks. In this work, we study the *training data reconstruction* problem, i.e., trying to reconstruct (parts of) the data a network was trained on without additional knowledge about network gradients or the initialisation.

Let us assume one has trained a neural network  $\Phi(x_i, \theta^*)$  with training examples  $(x_i, y_i) \in \mathbb{R}^K \times \mathbb{R}$ ,  $i \in \{1, \dots, m\}$ , by minimizing the expectation of a loss function  $\mathcal{L}$  over all training examples. Then we expect that  $\theta^* = \theta(x, y)$  for

$$(1) \quad \theta(x, y) \in \arg \min_{\theta} \frac{1}{m} \sum_{i=1}^m \mathcal{L}(\Phi(x_i; \theta), y_i).$$

Therefore, the natural way to recover the training data  $\{(x_i, y_i)\}$  a network  $\Phi$  is trained on if the final weights  $\theta^*$  are known, is to consider the bi-level optimization problem

$$(2) \quad \min_{x, y} l(\theta^*, \theta(x, y)) \quad \text{s. t.} \quad \theta(x, y) \text{ solves (1).}$$

Interestingly, previous works in this area, most prominently [4] and [5], did not consider (2). Instead, the authors used an analysis of Karush-Kuhn-Tucker (KKT) conditions on certain equivalent margin maximisation problems to propose an approach which - in our notation - could be phrased as

$$(3) \quad \min_x \|\nabla_{\theta} E(x, y; \theta^*)\|_2^2$$

$$(4) \quad \text{for } E(x, y; \theta) = \frac{1}{m} \sum_{i=1}^m \mathcal{L}(\Phi(x_i; \theta), y_i).$$

As shown in [6], the above can be interpreted as an approximation to the original cost function. More specifically, (3) becomes a majoriser of (2) (and can be used in iterative algorithms) if the costs  $E$  are strongly convex and the upper-level loss  $l$  is strongly smooth.

While [4, 5] demonstrated quite remarkable reconstructions of training data using (3) in image classification, we focus on the robustness and the faithfulness of such reconstruction with respect to different initialisations for both (2) and (3). Our numerical experiments demonstrate that while a carefully chosen initialisation allows recovering some examples  $x_i$  from the training data, there also exist cases where somewhat realistically looking data samples  $x_i$  are recovered that are *not* part of the training data: The energy landscapes of both (2) and (3), seem to have many minima, and initializing either approach with realistically looking images that have not been part of the training data quickly leads to a (numerical) convergence without resembling the training data at all. Thus, we argue that - despite the ability to reconstruct some training examples - some privacy remains

in the uncertainty of whether realistically looking images have actually been part of the training data.

## REFERENCES

- [1] C. Runkel, K. V. Gandikota, J. Geiping, C.-B. Schönlieb and M. Moeller, *Training Data Reconstruction: Privacy due to Uncertainty?*, Under Review (2024).
- [2] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer and C. Zhang, *Quantifying Memorization Across Neural Language Models*, The Eleventh International Conference on Learning Representations (2023), [https://openreview.net/forum?id=TatRHT\\_1cK](https://openreview.net/forum?id=TatRHT_1cK).
- [3] G. Somepalli, V. Singla, M. Goldblum, J. Geiping and T. Goldstein, *Diffusion art or digital forgery? investigating data replication in diffusion models*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023), pp. 6048–6058.
- [4] N. Haim, G. Vardi, G. Yehudai, O. Shamir and M. Irani, *Reconstructing training data from trained neural networks*, Advances in Neural Information Processing Systems (2022), vol. 35, pp. 2291–22924.
- [5] G. Buzaglo, N. Haim, G. Yehudai, G. Vardi, Y. Oz, Y. Nikankin and others, *Deconstructing Data Reconstruction: Multiclass, Weight Decay and General Losses*, Thirty-seventh Conference on Neural Information Processing Systems (2023).
- [6] J. Geiping and M. Moeller, *Parametric Majorization for Data-Driven Energy Minimization Methods*, Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019.

## Numerical linear algebra networks for solving linear inverse problems

OTMAR SCHERZER

(joint work with Andrea Aspri)

### 1. INTRODUCTION

We consider solving a potentially ill-conditioned **linear operator equation**:

$$(1) \quad F\mathbf{x} = \mathbf{y},$$

where  $\mathbf{x} \in \mathbb{R}^m$  and  $\mathbf{y} \in \mathbb{R}^{\overline{m}}$ . Following the terminology of [3, 1], we refer to  $\mathbb{R}^m$  as the *image space* and  $\mathbb{R}^{\overline{m}}$  as the *data space*. The main assumption of this work is that the operator  $F$  is **not** modeled by physical laws but indirectly via **training pairs**  $\mathcal{P} := \{(\mathbf{x}_i, \mathbf{y}_i) : i = 1, \dots, N\}$ , which satisfy

$$F\mathbf{x}_i = \mathbf{y}_i, \quad i = 1, \dots, N.$$

The integer  $N$  is called the *sampling size*. We study methods for **learning** the linear operator  $F$  and its inverse through *encoding* and *decoding*. After learning the linear operator  $F$ , we can solve Equation 1 for arbitrary data  $\mathbf{y} \in \mathbb{R}^{\overline{m}}$ . Operator learning is currently a very active field of research. Various methods have been developed, such as:

- **Black-box** strategies for linear operator learning (see, for instance, [11]).
- The use of **deep neural networks** for learning nonlinear operators (see [7, 9, 8, 10]).

We denote the spans of the training images and data by

$$X_N := \text{span} \{\mathbf{x}_i : i = 1, \dots, N\} \subseteq \mathbb{R}^m, \quad Y_N := \text{span} \{\mathbf{y}_i : i = 1, \dots, N\} \subseteq \mathbb{R}^{\overline{m}}.$$

We assume that the training images  $\mathbf{x}_i$  are linearly independent and that  $F$  has a trivial null space, so that the training data  $\mathbf{y}_i$  are also linearly independent.

## 2. CODERS FOR OPERATOR LEARNING

In **manifold learning**, a fundamental tool in machine learning, **coding** refers to representing data points that are assumed to lie on a **low-dimensional** manifold (see, for instance, [5, 4]). This setting, which does not involve an operator connecting data, is illustrated in Figure 1. Our approach for linear operator learning

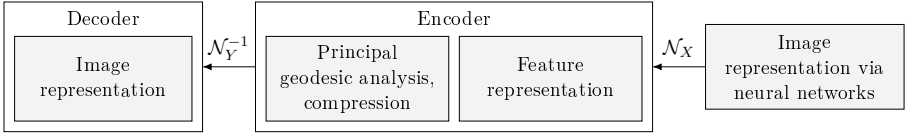


FIGURE 1. Variational encoding and decoding with neural networks: The image data are represented via a neural network, transformed into a **feature** space by the operator  $\mathcal{N}_X$ , compressed using principal geodesic analysis, and are then decoded with  $\mathcal{N}_Y^{-1}$ .

is conceptually similar but employs different techniques (compare Figures 1 and 2). Specifically, we base our strategy on **orthonormalization** of the training images and perform **principal component analysis** (PCA) on the data obtained by applying  $F$  to these orthonormalized images, as illustrated in Figure 2.

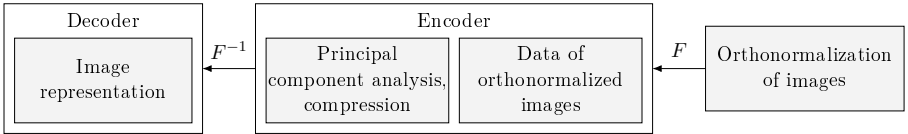


FIGURE 2. Proposed encoding and decoding scheme for **linear** operators: The image data are orthonormalized, and the corresponding data are computed by applying  $F$  - without utilizing any physical model for the forward operator. PCA is then applied to the resulting data to compress the data space. The decoder computes the inverse for given data in the compressed space.

## 3. RESULTS

The main results of our work, based on [2], are as follows:

- (1) We show that orthonormalization and coder strategies are interconnected concepts. In particular, we show that orthonormalization can be formulated as a **customized neural network**; see Figure 3.

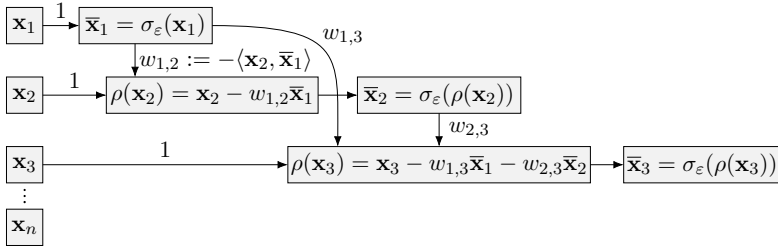


FIGURE 3. Neural network structure of the Gram-Schmidt orthonormalization. We use the function  $\sigma_\varepsilon(\mathbf{x})$  as an approximation of the projection operator  $\mathbf{x}/\|\mathbf{x}\|$  in Gram-Schmidt.

- (2) The double orthonormalization strategy, consisting of Gram-Schmidt orthonormalization of images and PCA of data, yields the singular value decomposition (SVD) of the linear operator  $F$ . Consequently, this strategy serves as a regularization method (see, for instance, [6]).

#### REFERENCES

- [1] A. Aspri, L. Frischauf, Y. Korolev, and O. Scherzer, *Data driven reconstruction using frames and riesz bases*, Deterministic and Stochastic Optimal Control and Inverse Problems (B. Jadamba, A. A. Khan, S. Migórski, and M. Sama, eds.), CRC Press (2021), pp. 303–318.
- [2] A. Aspri, L. Frischauf, and O. Scherzer, *Spectral function space learning and numerical linear algebra networks for solving linear inverse problems*, arXiv:2408.10690 (2024).
- [3] A. Aspri, Y. Korolev, and O. Scherzer, *Data driven regularization by projection*, IOP Publishing (2020), **36**, no. 12, 125009, Hybrid-OA.
- [4] J. Braunsmann, M. Rajkovic, M. Rumpf, and B. Wirth, *Learning low bending and low distortion manifold embeddings*, IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2021), vol. 85, pp. 4411–4419.
- [5] M. A. G. Duff, N. D. F. Campbell, and M. J. Ehrhardt, *Regularising inverse problems with generative machine learning models*, J Math Imaging Vis (2023), **66**, no. 1, pp. 37–56.
- [6] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of inverse problems*, Mathematics and its Applications, no. 375, Kluwer Academic Publishers Group, 1996.
- [7] Nikola Kovachki, Samuel Lanthaler, and Siddhartha Mishra, *On universal approximation and error bounds for fourier neural operators*, Journal of Machine Learning Research (2021), **22**, no. 290, pp. 1–76.
- [8] S. Lanthaler, Z. Li, and A. M. Stuart, *The nonlocal neural operator: Universal approximation*.
- [9] S. Lanthaler, S. Mishra, and G. E. Karniadakis, *Error estimates for deeponets: a deep learning framework in infinite dimensions*, arXiv:2102.09618 (2022) **6**, no. 1.
- [10] S. Lanthaler and A. M. Stuart, *The parametric complexity of operator learning*, arXiv:2306.15924 (2023).
- [11] A. Papoulis, *The fourier integral and its applications*, McGraw-Hill, 1962.

## Lie Algebra Canonicalization: Equivariant Neural Operators under arbitrary Lie Groups

ZAKHAR SHUMAYLOV

(joint work with P. Zaika, J. Rowbottom, F. Sherry, M. Weber, C. B. Schönlieb)

The quest for robust and generalizable machine learning models has driven recent interest in exploiting symmetries through equivariant neural networks. In the context of PDE solvers, recent works have shown that Lie point symmetries can be a useful inductive bias for Physics-Informed Neural Networks (PINNs) through data [1] and loss [2] augmentation. Despite this, directly enforcing equivariance within the model architecture for these problems remains elusive. This is because many PDEs admit non-compact symmetry groups, which do not act in representations and oftentimes are not studied beyond their infinitesimal generators, making them incompatible with most existing equivariant architectures. For example, the heat equation was only recently shown by [3] to admit a group of point symmetries  $SL(2, \mathbb{R}) \ltimes_{\phi} H(1, \mathbb{R})$ , which is not only non-compact, it also does not act in a representation on the underlying jet space. This renders most of the currently available equivariant architectures unusable.

In [4], we propose Lie aLgebrA Canonicalization (LieLAC), a novel approach that exploits only the action of infinitesimal generators of the symmetry group, circumventing the need for knowledge of the full group structure. Motivated by limitations of existing equivariant architectures, we turn to *energy-based canonicalization* as a means for inducing equivariance in existing models. This approach was first introduced in [5], but remains largely unexplored. Here, we revisit and extend this approach, and explore its utility in scientific machine learning applications.

Noting that many existing definitions and results in the prior literature do not extend readily to the general setting, we extend and refine existing definitions to

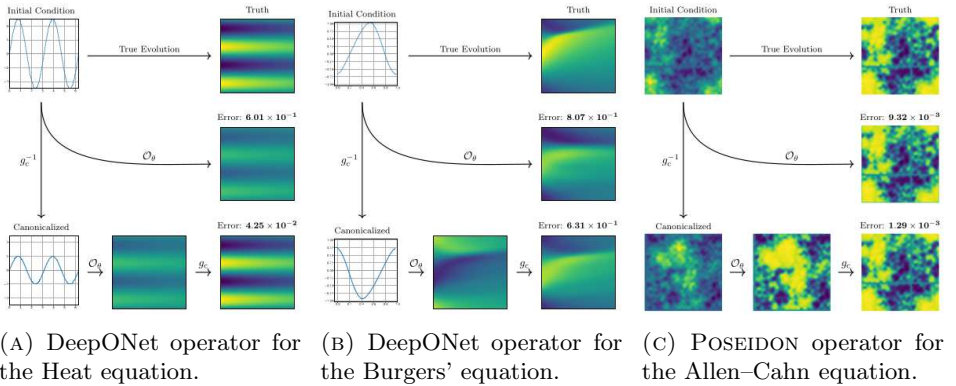


FIGURE 1. Canonicalization pipeline for numerical PDE evolution.



handle non-discrete and non-compact groups. The proposed energy-based canonicalization approach provides a unifying theoretical framework for constructing frames and canonicalizations. This new framework encapsulates prior work of [6] on weighted frames, and our energy-based canonicalizations live naturally in this framework. We showcase all the different notions of canonicalization and framing together, shown in 2.

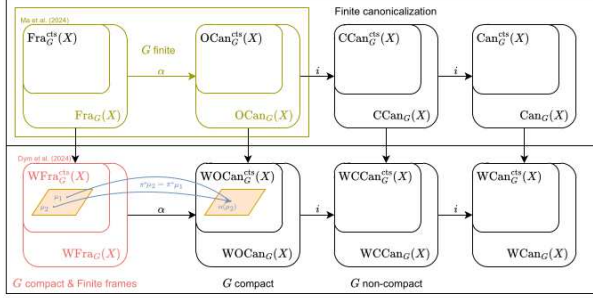


FIGURE 2. Connections between the notions of frames and canonicalizations introduced previously and in this work. Top row: finite frames and canonicalizations, mapping vertically into weighted versions via normalized counting measures. Inside each, contained a sequentially closed subspace of those that preserve continuity.

## 1. FRAMES AND CANONICALIZATIONS

The two main ways of imposing equivariance on arbitrary models: *frame averaging* and *canonicalization*, both arise from the study of the Reynolds operator. *Frames* are defined, for each element  $x \in X$ , as a subset of the group  $G$  for which to average over, with an equivariance condition detailed below to ensure equivariance of the resulting function. *Canonicalizations*<sup>1</sup> are the opposite perspective, for each element  $x \in X$ , directly providing a subset of  $X$  over which to directly average the function over with an invariance condition to ensure the resulting function is equivariant.

### Definition 1.

*Frames are functions  $\mathcal{F} : X \rightarrow 2^G \setminus \{\emptyset\}$  such that for any  $x \in X$  and  $g \in G$ :  $\mathcal{F}(gx) = g\mathcal{F}(x)$ . We call the space of frames  $\text{Fra}_G(X)$ .*

*Canonicalizations are functions  $\mathcal{C} : X \rightarrow 2^X \setminus \{\emptyset\}$  s.t.  $\forall x \in X, g \in G : \mathcal{C}(gx) = \mathcal{C}(x)$ . We call the space of canonicalizations  $\text{Can}_G(X)$ . Their action on functions*

<sup>1</sup>Original proposal of [5] relied on these being singletons, requiring authors to introduce relaxed equivariance. Treating these as sets turns out to be natural for both energy canonicalizations, and for providing (equivalence) isomorphism of (weighted) canonicalizations with (weighted) frames [7].

$\phi$  is:

$$\mathcal{F}(\phi)(x) = \frac{1}{|\mathcal{F}(x)|} \sum_{g \in \mathcal{F}(x)} \phi(g^{-1}x) \quad \mathcal{C}(\phi)(x) = \frac{1}{|\mathcal{C}(x)|} \sum_{y \in \mathcal{C}(x)} \phi(y)$$

As  $\mathcal{F}(x) = G_x \mathcal{F}(x)$  for any frame, frame averaging is limited to the case that the frames are finite at each point, i.e. that the set  $\mathcal{F}(x)$  is finite  $\forall x \in X$ . For frames this can only be the case when the group acts with finite stabilizers at all points. This leads to the following definition of [6], which we extend to canonicalizations.

**Definition 2.**

(Weakly)-equivariant weighted frames are functions  $\mu_{[\cdot]} : X \rightarrow P\text{Meas}(G)$  s.t.  $\forall x \in X, g \in G : \mu_{gx} = g_* \mu_x$  (respectively:  $\forall x \in X, g \in G. (\pi_{gx})_* \mu_{gx} = (\pi_{gx})_* g_* \mu_x$  where  $\pi_{gx} : G \rightarrow G/G_{gx}$  is the quotient map. We call the space of weakly-equivariant weighted frames  $WFra_G(X)$ .

Weighted canonicalizations are  $G$ -invariant functions  $\kappa_{[\cdot]} : X \rightarrow P\text{Meas}(X)$ . We call the space of weighted canonicalizations  $WCan_G(X)$ . Their action on  $\phi$  is:

$$\mu(\phi)(x) = \int_G \phi(g^{-1} \cdot x) d\mu_x(g) \quad \kappa(\phi)(x) = \int_X \phi d\kappa_x$$

As in the non-weighted case, weighted orbit canonicalizations and weakly-equivariant weighted frames can be shown to be equivalent [4].

## 2. ENERGY CANONICALIZATION

For constructing frames and canonicalization we turn to energy canonicalization. We start with an energy function  $E : X \rightarrow [0, +\infty]$  which we then minimize over the orbits of the group action  $Gx$ . First, in the case where  $G$  is finite, every orbit in  $X$  necessarily has a minimum of  $E$ , and we note that energy minimization naturally results in a frame:

$$(1) \quad \mathcal{F}_E(x) = \arg \min_{g \in G} E(g^{-1}x) \quad \mathcal{C}_E(x) = \arg \min_{y \in Gx} E(y)$$

In the non-compact cases we wish to consider, the classical definitions aren't suitable for two main reasons: the set of minima may not be finite or there may be no minima. Moving to their weighted versions turns out to be beneficial, however further care needs to be taken due to the non-compactness. We refer the reader to [4] for a full discussion. In turn, by performing the minimization in (1), canonicalizations can efficiently be found. We showcase the effectiveness of our method on image classification tasks under homography and affine group invariance in. We further discuss turning neural operators Lie point symmetry equivariant on three examples, including the pre-trained POSEIDON foundation model [8].

We emphasize that our proposed approach is generic, in that it has the potential to apply to a wide range of models and learning tasks. The energy canonicalization framework developed here is readily applicable for an **arbitrary** group with an **arbitrary** action, including those not acting linearly, therefore being extendable to gauge and local symmetries. For Lie groups with smooth actions, one only requires knowledge of the action of some basis of the Lie algebra, without requiring

a global parametrization of the group. This flexibility stems from the choice of energy functional, which directly influences empirical performance and the size of resulting canonicalizations. We offer guidelines for constructing energy functionals and for performing Lie group optimization in [4].

## REFERENCES

- [1] J. Brandstetter, M. Welling, and D. E. Worrall, *Lie point symmetry data augmentation for neural PDE solvers*, International Conference on Machine Learning (2022), pp. 2241–2256.
- [2] T. Akhound-Sadegh, L. Perreault-Levasseur, J. Brandstetter, M. Welling, and S. Ravanbakhsh, *Lie Point Symmetry and Physics-Informed Networks*, Advances in Neural Information Processing Systems (2023), vol. 36, Curran Associates, Inc..
- [3] S. D. Koval, and R. O. Popovych, *Point and generalized symmetries of the heat equation revisited*, Journal of Mathematical Analysis and Applications (2023), vol. 527(2), pp. 127430.
- [4] Z. Shumaylov, P. Zaika, J. Rowbottom, F. Sherry, M. Weber, and C.-B. Schönlieb, *Lie Algebra Canonicalization: Equivariant Neural Operators under arbitrary Lie Groups*, arXiv preprint arXiv:2410.02698 (2024).
- [5] S.-O. Kaba, A. K. Mondal, Y. Zhand, Y. Bengio, and S. Ravanbakhsh, *Equivariance with Learned Canonicalization Functions*, Proceedings of the 40th International Conference on Machine Learning (2023), pp. 15546–15566.
- [6] N. Dym, H. Lawrence, and J. W. Siegel, *Equivariant Frames and the Impossibility of Continuous Canonicalization*, ArXiv preprint arXiv:2402.16077 (2024).
- [7] G. Ma, Y. Wang, D. Lim, S. Jegelka, and Y. Wang, *A Canonization Perspective on Invariant and Equivariant Learning*, arXiv preprint arXiv:2405.18378 (2024).
- [8] M. Herde, B. Raonič, T. Rohner, R. Käppeli, R. Molinaro, E. de Bézenac, and S. Mishra, *Poseidon: Efficient Foundation Models for PDEs*, arXiv preprint arXiv:2405.19101 (2024).

## Neural sampling from Boltzmann densities: Fisher-Rao curves in the Wasserstein geometry

GABRIELE STEIDL

(joint work with Jannis Chemseddine, Christian Wald, Richard Duong)

We are interested in sampling from a Boltzmann density  $\rho_D = e^{-f_D}/Z_D$  with unknown normalizing constant  $Z_D$ . We aim to construct a curve

$$\rho_t = e^{-f_t}/Z_t, \quad Z_t := \int_{\mathbb{R}^d} e^{-f_t} dx$$

interpolating between a simple density  $\rho_Z$  and the target  $\rho_D$ , i.e.,  $\rho_0 = \rho_Z$  and  $\rho_1 = \rho_D$ . In this case, straightforward computation gives

$$\partial \mu_t = \partial_t \rho_t = -(\partial_t f_t - \mathbb{E}_{\rho_t}[\partial_t f_t]) \rho_t,$$

where it maybe useful to note that  $\mathbb{E}_{\rho_t}[\partial_t f_t] = -\partial_t \log Z_t$ . This differential equation is a Fisher-Rao flow equation. For more information on Fisher-Rao flows see, e.g., [2, 7].

If such curve admits a velocity field  $v_t$  and there exists a solution  $\varphi$  of

$$(1) \quad \partial_t \varphi_t = v_t(\varphi_t), \quad \varphi_0 = \text{Id}$$

and  $(\varphi_t)_\#(\rho_Z dx) = \rho_t dx$ . Then we can use  $\varphi_1$  and  $\rho_Z$  to sample from  $\rho_D$ . An important question is, whether for a given family of functions  $f_t$ , such velocity

fields exists. While, for a fixed time  $t$ , the existence of such velocity fields is well established, see e.g. [3], we addressed global existence with integrability of  $v : [0, 1] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$  in time and space in [1]. This is directly related to the question whether a large class of Fisher-Rao curves  $\rho_t$  is absolutely continuous in the Wasserstein geometry, and in particular, fulfills a continuity equation

$$\partial_t \mu_t + \nabla \cdot (\mu_t v_t) = 0.$$

Note that for Boltzmann densities, the second summand of the continuity equation reads as

$$(2) \quad \nabla \cdot (\mu_t v_t) = \nabla \cdot (\rho_t v_t) = (-\langle \nabla f_t, v_t \rangle + \nabla \cdot v_t) \rho_t.$$

This can be reduced to finding solutions of a family of PDEs

$$(\partial_t f_t - \mathbb{E}_{\rho_t}[\partial_t f_t]) \rho_t = \nabla \cdot (\rho_t v_t).$$

In the ideal case that we can find a vector field  $v_t = \nabla s_t$ , this can be rewritten with (2) and  $\rho_t > 0$  as the family of Poisson equations

$$(\partial_t f_t - \mathbb{E}_{\rho_t}[\partial_t f_t]) = -\langle \nabla f_t, \nabla s_t \rangle + \Delta s_t,$$

which needs to be solved for  $s_t$ . For fixed time  $t$ , the solution of the Poisson equation has already been examined, e.g., in [3] and for more recent work, see also [5, 6]. However, the pointwise solution for each fixed  $t$  does not ensure that  $(\rho_t, \nabla s_t)$  gives rise to an absolutely continuous Wasserstein curve, since by definition this requires additional properties of  $\nabla_x s : [0, 1] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$  globally in  $t$ : i)  $\nabla_x s$  is Borel measurable on  $[0, 1] \times \mathbb{R}^d$ , and ii)  $t \mapsto \|\nabla_x s(t, \cdot)\|_{L_2(\rho_t, \mathbb{R}^d)}^2 \in L_1([0, 1])$ . If also, iii)  $\nabla_x s(t, \cdot) \in T_{\rho_t} \mathcal{P}_2(\mathbb{R}^d)$  for a.e.  $t \in [0, 1]$ , then it is known that the velocity field  $\nabla_x s$  is the minimal one. With the appropriate Hilbert spaces at hand, we showed the existence of the solutions of these PDEs and the existence of an integrable velocity field  $v_t$  in time and space which admits moreover a minimality property of  $\|v_t\|_{L_2(\mathbb{R}^d, \rho_t)}$  in [1].

Finding  $\varphi_1$  requires both determining a curve  $f_t$  and the associated vector field  $v_t$ . One way is to choose a curve interpolating between  $f_Z$  and  $f_D$  first, and then to learn the velocity field. For the **linear interpolation**

$$f_t = (1 - t)f_Z + tf_D$$

we showed that there exists indeed an integrable velocity field implying the absolute continuity of the curve in the Wasserstein space. This approach leads to learning neural networks  $(v_t^{\theta_1}, C_t^{\theta_2})$  by minimizing the loss function

$$L(\theta) := \mathbb{E}_{t \in \mathcal{U}[0, 1], x \in \mathcal{U}[a, b]^d} [\mathcal{E}(\theta, x, t)],$$

$$\mathcal{E}(\theta, x, t) := |f_1 - f_0 - C_t^{\theta_2} + \langle \nabla f_t, v_t^{\theta_1} \rangle - \nabla \cdot v_t^{\theta_1}|^2, \quad \theta := (\theta_1, \theta_2).$$

Unfortunately, the velocity field corresponding to the linear interpolation can admit an exploding norm  $\|v_t\|_{L_2(\mathbb{R}^d, \rho_t)}$ , leading to a “non-smooth” particle transport, see Figure 1.

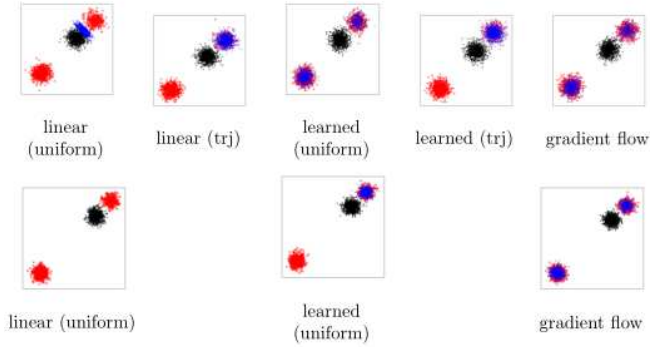


FIGURE 1. Sampling results for  $m = 8$  (top) and  $m = 15$  (bottom): source (black), target (red) and estimated (blue). We cannot learn a meaningful vector field for the linear interpolation for both sampling strategies (uniform and along trajectories). Sampling along the trajectories leads to mode collapse for the learned interpolation. While the learned interpolation with uniform sampling covers the modes for  $m = 8$ , increasing  $m$  leads to mode collapse again. In contrast, the gradient flow interpolation does not mode collapse for both  $m$ .

Another approach, recently proposed by [4], considers **learned interpolation**

$$f_t = (1 - t)f_Z + tf_D + t(1 - t)\psi_t$$

with unknown  $\psi$  and learns  $(\psi_t, v_t)$  simultaneously. However, it is theoretically unclear whether the learned curve is well-behaved or the velocity field is optimal with respect to the above norm.

As an alternative approach, we propose instead to deal directly with a mathematically accessible curve on  $[0, T]$  and consider the **gradient flow interpolation**

$$f_t = \frac{T - t}{T} f_D + t\psi_t.$$

Fixing the velocity field as  $v_t := \nabla(f_t - f_Z)$  we only have to learn  $\psi_t$ . The resulting PDE is the well-known Fokker-Planck equation related to Langevin dynamics which has a solution with Boltzmann densities  $\rho_t$ , if, e.g.,  $\rho_Z$  is a Gaussian. Then, the corresponding SDE is the Ornstein-Uhlenbeck (OU) process. Here, the backward ODE of (1) must be applied for sampling. We learned the networks for the above three cases and demonstrated by numerical examples the effectiveness of our approach in [1].

## REFERENCES

- [1] J. Chemseddine, C. Wald, R. Duong, and G. Steidl. Neural sampling from Boltzmann densities: Fisher-Rao curves in the Wasserstein geometry. *ArXiv 2410.03282*, 2024.
- [2] T. O. Gallouët and L. Monsaingeon. A JKO splitting scheme for Kantorovich–Fisher–Rao gradient flows. *SIAM Journal on Mathematical Analysis*, 49(2):1100–1130, 2017.
- [3] R. S. Laugesen, P. G. Mehta, S. P. Meyn, and M. Raginsky. Poisson’s equation in nonlinear filtering. *arXiv preprint arXiv:1412.5845*, 2014.
- [4] B. Máté and F. Fleuret. Learning interpolations between Boltzmann densities. *arXiv preprint arXiv:2301.07388*, 2023.
- [5] S. Reich. A dynamical systems framework for intermittent data assimilation. *BIT Numerical Mathematics*, 51:235–249, 2011.
- [6] A. Taghvaei and P. G. Mehta. A survey of feedback particle filter and related controlled interacting particle systems. *Annual Reviews in Control*, 55:356–378, 2023.
- [7] Y. Wang and W. Li. Accelerated information gradient flow. *Journal of Scientific Computing*, 90:1–47, 2022.

## Subspace diffusion posterior sampling for travel-time tomography

XIAOQUN ZHANG

(joint work with Xiang Cao)

## ABSTRACT

Diffusion models have been widely studied as effective generative tools for solving inverse problems. The main ideas focus on performing the reverse sampling process conditioned on noisy measurements, using well-established numerical solvers for gradient updates. Although diffusion-based sampling methods can produce high-quality reconstructions, challenges persist in nonlinear PDE-based inverse problems and sampling speed. In this work, we explore solving PDE-based travel-time tomography based on subspace diffusion generative models. Our main contributions are twofold: First, we propose a posterior sampling process for PDE-based inverse problems by solving the associated adjoint-state equation in a plug-and-play fashion. Second, we present a subspace-based dimension reduction technique, enabling solving the PDE-based inverse problems from coarse to refined grids, for conditional sampling acceleration. Our numerical experiments showed satisfactory advancements in improving the travel-time imaging quality and reducing the sampling time for reconstruction.

**Keywords:** Score-based diffusion models; Diffusion posterior sampling; Subspace diffusion generative models; Nonlinear PDE-based inverse problems; Travel-time tomography; Adjoint-state method.

## 1. INTRODUCTION

In the context of solving inverse problems [8], we aim to recover the original parameter data  $\mathbf{x}_0$  from the noisy or incomplete measurement  $\mathbf{y}^\delta$ , correlated through the forward model  $\mathcal{A}$  and the measurement noise  $\mathbf{n}$ :

$$(1) \quad \mathbf{y}^\delta = \mathcal{A}(\mathbf{x}_0) + \mathbf{n}.$$

In the application of diffusion models to solving inverse problems, the conditional reverse sampling process for reconstruction is framed as Bayesian posterior sampling [2]. This approach models the conditional score function  $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}^\delta)$  as a factorization based on Bayes' rule:

$$(2) \quad \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}^\delta) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}^\delta|\mathbf{x}_t)$$

Various studies [2, 6, 1] have been devoted to approximating the intractable gradient term  $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}^\delta|\mathbf{x}_t)$  to enable sampling from the posterior distribution  $p(\mathbf{x}_0|\mathbf{y}^\delta)$ .

However, applying these methods directly to nonlinear PDE-based scenarios is challenging, which stems from the following two aspects:

- *Nonlinear PDE-based constraints.* The inherent nonlinearity of the forward operator  $\mathcal{A}$  leads to the dependence of the corresponding Fréchet derivatives  $(\partial\mathcal{A})_{\mathbf{x}_0}$  on the data  $\mathbf{x}_0$ , that is:

$$(3) \quad \nabla_{\mathbf{x}_0} \|\mathbf{y}^\delta - \mathcal{A}(\mathbf{x}_0)\|_2^2 = 2(\partial\mathcal{A})_{\mathbf{x}_0}^* (\mathcal{A}(\mathbf{x}_0) - \mathbf{y}^\delta)$$

under the  $l_2$  norm. Auto-differentiation approach fails due to the implicit dependence of  $\mathcal{A}(\mathbf{x}_0)$  on  $\mathbf{x}_0$  in the field of PDE-based inverse problems.

- *Computational cost.* During the conditional reverse sampling process, each gradient evaluation step requires solving the forward and adjoint PDEs multiple times under different boundary conditions, making the inference much slower than the unconditional sampling. The dimensionality of both numerical PDE solvers and score functions [4] is critical in determining the computational complexity of each sampling step, thereby influencing the overall runtime.

Specifically, we consider solving the PDE-based travel-time tomography [7] problem within the diffusion posterior sampling (DPS) [2] framework. In the case of single point source condition,  $N$  different source points  $\{(u_n^s, v_n^s)\}_{n=1}^N$  are placed in  $\Omega := [0, 1]^2$  or on  $\partial\Omega$ , and for each source point, the governing first travel-time field  $\mathbf{Y}_n(u, v)$  can be described as the following Eikonal equation [5]

$$(4) \quad \begin{aligned} \mathbf{X}_0(u, v) \|\nabla \mathbf{Y}_n(u, v)\|_2 &= 1, & \forall (u, v) \in \Omega, \\ \text{s.t. } \mathbf{Y}_n(u_n, v_n) &= 0, & \text{when } (u, v) = (u_n^s, v_n^s). \end{aligned}$$

where  $\|\cdot\|_2$  refers to the  $L_2$  norm,  $\mathbf{X}_0(u, v)$  refers to the positive speed field in the square  $\Omega$ , and its value is normalized into the range  $[0, 1]$ . If we further denote the Eikonal operator  $\mathcal{P}_n : \mathbf{X}_0(u, v) \rightarrow \mathbf{Y}_n(u, v)$  given by (4) and collect the travel-time information on the finite receiver subset  $\Omega_r := \{(u_m^r, v_m^r)\}_{m=1}^M \subset \partial\Omega$ , the forward model for the source-receiver pair  $\{(u_n^s, v_n^s), (u_m^r, v_m^r)\}$  can be stated as:

$$(5) \quad \mathbf{y}_{m,n}^\delta := \mathcal{P}_n(\mathbf{X}_0)(u_m^r, v_m^r) + \delta\boldsymbol{\eta}_{m,n}, \quad \mathbf{X}_0(u, v) : \mathbb{R}^2 \mapsto \mathbb{R},$$

where  $\boldsymbol{\eta}_{m,n} \in \mathbb{R}$  is an unknown noise model and  $\delta$  controls the noise level. The collected data  $\mathbf{y}_{m,n}^\delta$  from multiple source-receiver pairs constitutes a data matrix  $\mathbf{y}^\delta \in \mathbb{R}^{M \times N}$ , and the objective of travel-time tomography is to reconstruct the underlying speed field  $\mathbf{X}_0(u, v)$  from  $\mathbf{y}^\delta$ .

## 2. METHODS

For problem-solving, we consider solving the PDE-based inverse problems within the DPS framework. *The adjoint-state method* will be introduced in Section 2.1 as the formulation is designed to tackle this issue. The multi-resolution and discretization will be presented in Section 2.2 to capture the transition from the continuous multi-resolution formulation of the PDE to the discrete subspace diffusion settings.

For sampling acceleration, we propose integrating the subspace-based technique into the PDE-based diffusion posterior sampling process in Section 2.3

**2.1. Adjoint-state method for travel-time tomography.** For the associated inverse problem, we aim to minimize the following misfit functional on receivers

$$(6) \quad \mathcal{J}(\mathbf{X}_0) = \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^M (\mathbf{y}_{m,n}^\delta - \mathbf{y}_{m,n})^2,$$

To minimize (6), we perturb the speed field  $\mathbf{X}_0$  by  $\varepsilon \tilde{\mathbf{X}}_0$  and the resulting change of the travel-time field  $\varepsilon \tilde{\mathbf{Y}}_n$  satisfies the following perturbation PDE:

$$(7) \quad \nabla \mathbf{Y}_n \cdot \nabla \tilde{\mathbf{Y}}_n = -\frac{\tilde{\mathbf{X}}_0}{\mathbf{X}_0^3}.$$

Here,  $\boldsymbol{\Lambda}_n$  is required to satisfy the following adjoint PDE

$$(8) \quad \nabla \cdot (\boldsymbol{\Lambda}_n \nabla \mathbf{Y}_n) = 0$$

with the Dirichlet boundary condition

$$(9) \quad \frac{\partial \mathbf{Y}_n}{\partial \mathbf{n}} \boldsymbol{\Lambda}_n = \sum_{m=1}^M \delta(u - u_m^r, v - v_m^r) (\mathbf{y}_{m,n}^\delta - \mathbf{y}_{m,n})$$

where  $\mathbf{n}$  is the unit outward normal of the boundary. Then, we conclude that

$$(10) \quad \partial \mathcal{J}(\mathbf{X}_0)(\tilde{\mathbf{X}}_0) = \sum_{n=1}^N \int_{\Omega} \frac{\tilde{\mathbf{X}}_0 \boldsymbol{\Lambda}_n}{\mathbf{X}_0^3} du dv.$$

To obtain the well-posed descent direction, we use the elliptical smoothing regularization to choose the descent direction

$$(11) \quad \tilde{\mathbf{X}}_0 = -(\mathbf{I} - \mu \Delta)^{-1} \left( \sum_{n=1}^N \frac{\boldsymbol{\Lambda}_n}{\mathbf{X}_0^3} \right),$$

where  $\Delta$  refers to the Laplacian operator.



**2.2. Multi-resolution and discretization.** Given the continuous speed field  $\mathbf{X}_0$  as the parameter distribution of (4), we use the following notations based on convolution with downsampling and upsampling to define the multi-resolution:

$$(12) \quad (\varphi * \mathbf{X}_0)(u, v) = \int_{\mathbb{R}^2} \varphi(\tilde{u}, \tilde{v}) \mathbf{X}_0(u - \tilde{u}, v - \tilde{v}) d\tilde{u} d\tilde{v}$$

with downsampling and upsampling operators

$$(13) \quad \begin{aligned} (\downarrow \mathbf{X}_0)(u, v) &:= \mathbf{X}_0(2u, 2v), \\ (\uparrow \mathbf{X}_0)(u, v) &:= \mathbf{X}_0(2^{-1}u, 2^{-1}v). \end{aligned}$$

We continue to define the resolution reduction operator  $\Phi(\cdot)$ , which maps from the  $(k-1)$ -th resolution to the  $k$ -th resolution, as follows:

$$(14) \quad \mathbf{X}_0^k := \Phi(\mathbf{X}_0^{k-1})(u, v) = \frac{1}{2} \downarrow (\varphi * \mathbf{X}_0^{k-1}),$$

Since the diffusion generative process is defined on the discrete grids, we should consider the discretization of the  $k$ -th resolution speed field samples  $\mathbf{X}_0^k$  as the projection onto the subspaces. For the corresponding inverse problem in downsampled subspace, the misfit function  $\mathcal{J}_k^h(\cdot) : \mathbb{R}^{2^{s-k} \times 2^{s-k}} \mapsto \mathbb{R}$ :

$$(15) \quad \mathcal{J}_k^h(\mathbf{x}_0^k) = \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^M (\mathbf{y}_{m,n}^\delta - \mathbf{y}_n^k[l_m^{r,k}, j_m^{r,k}])^2$$

is defined on the discrete sample  $\mathbf{x}_0^k$ . With the help of Theorem 2.6 in [3], we can derive the upper bound estimate of the gap between  $\mathcal{J}_k^h(\mathbf{x}_0^k)$  and  $\mathcal{J}(\mathbf{X}_0)$ :

**Theorem 1.** *For the given convolution kernel*

$$\varphi_h(u, v) = \frac{1}{4} \sum_{(a,b) \in \{0,1\}^2} \delta(u + ah, v + bh),$$

the gap between  $\mathcal{J}_k^h(\mathbf{x}_0^k)$  and  $\mathcal{J}(\mathbf{X}_0)$  is upper bounded by

$$(16) \quad |\mathcal{J}_k^h(\mathbf{x}_0^k) - \mathcal{J}(\mathbf{X}_0)| \leq C_1 \sqrt{h} + C_{k,\mu} h \|\tilde{\mathbf{X}}_0\|_{H_\mu^1(\Omega)}^2 \|\mathbf{X}_0\|_{H_\mu^2(\Omega)} + O(h)$$

where  $C_{k,\mu}$  is a fixed constant given  $k, \mu$ , and  $C_1$  depends on  $\mathbf{X}_0^k$  and  $\mathbf{y}^\delta$ .

**2.3. Subspace diffusion posterior sampling.** We consider utilizing the discrete regularized adjoint-state method to provide  $\nabla_{\hat{\mathbf{x}}_0} \|\mathbf{y}^\delta - \mathcal{A}(\hat{\mathbf{x}}_0)\|_2^2$  for guidance during the diffusion posterior sampling process, that is

$$(17) \quad \begin{aligned} \nabla_{\mathbf{x}_t} \log p(\mathbf{y}^\delta | \hat{\mathbf{x}}_0(\mathbf{x}_t)) &\simeq -2\rho \frac{\partial \hat{\mathbf{x}}_0(\mathbf{x}_t)}{\partial \mathbf{x}_t} (\mathbf{I} - \mu\Delta)^{-1} \nabla_{\hat{\mathbf{x}}_0} \mathcal{J}_0^h(\hat{\mathbf{x}}_0) \\ &= 2\rho \frac{\partial \hat{\mathbf{x}}_0(\mathbf{x}_t)}{\partial \mathbf{x}_t} (\mathbf{I} - \mu\Delta)^{-1} \left( \sum_{n=1}^N \boldsymbol{\lambda}_n / \hat{\mathbf{x}}_0^3 \right) \end{aligned}$$

where  $(\mathbf{I} - \mu\Delta)^{-1}$  is defined in (11) with a homogeneous boundary condition. To accelerate the sampling process, we consider the reverse posterior SDE with the

same form for  $t \in [t_k, t_{k-1}]$  in each subspace. Besides,  $\mathbf{x}_t^k$  is obtained via

$$(18) \quad \mathbf{x}_t^k := \mathbf{U}_k^T \mathbf{x}_t = \frac{1}{2^k} \left( \prod_{l=0}^{k-1} \mathcal{D}_l \right) \mathbf{x}_t$$

where the average-pooling operator  $\mathcal{D}_k$ . Here, we choose the surrogate misfit function  $\mathcal{J}_k^h(\cdot)$  for sampling, that is

$$(19) \quad -2\rho_k \frac{\partial \hat{\mathbf{x}}_0^k(\mathbf{x}_t^k)}{\partial \mathbf{x}_t^k} (\mathbf{I} - \mu \Delta)^{-1} \left( \nabla_{\hat{\mathbf{x}}_0^k} \mathcal{J}_k^h(\hat{\mathbf{x}}_0^k) \right),$$

We should note that the primary benefit of the subspace-based diffusion posterior sampling lies in its capability to simultaneously reduce the dimensionality of score function evaluations and the complexity of solving numerical PDEs. Furthermore, it is important to emphasize that this subspace posterior sampling approach is **applicable to other PDE-based inverse problems**, as long as the adjoint equations can be explicitly formulated.

### 3. EXPERIMENT RESULTS

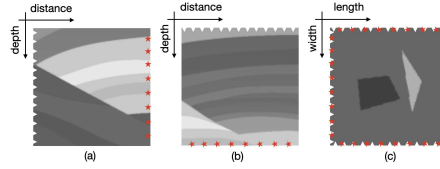


FIGURE 1. There are three geometries for placing the source-receiver pairs: (a)(b) For the Marmousi dataset, they are arranged in horizontal and vertical patterns; (c) For the KIT4 dataset, they are arranged in a circular pattern. As illustrated, the red pentagrams represent the locations of the sources, and the white triangles represent the locations of the receivers.

Method	Marmousi (horizontal)		Marmousi (vertical)		KIT4 (surround)	
	RMSE ↓	SSIM ↑	RMSE ↓	SSIM ↑	RMSE ↓	SSIM ↑
L-BFGS	0.128	0.457	0.166	0.359	0.054	0.713
DIP	0.147	0.351	0.166	0.264	0.066	0.342
RED-Diff	0.173	0.193	0.178	0.168	0.105	0.085
Supervised Post-Processing	0.039	0.655	<u>0.104</u>	<u>0.491</u>	<b>0.031</b>	<b>0.925</b>
PDE-based Full-space DPS	<b>0.014</b>	<b>0.768</b>	<b>0.089</b>	<b>0.525</b>	<u>0.050</u>	<u>0.864</u>
PDE-based Subspace DPS	<u>0.024</u>	<u>0.679</u>	0.113	0.462	0.062	0.837

TABLE 1. The quantitative results (RMSE, SSIM) are evaluated on the Marmousi and KIT4 test datasets for different methods. **Bold**: best, underline: second best.

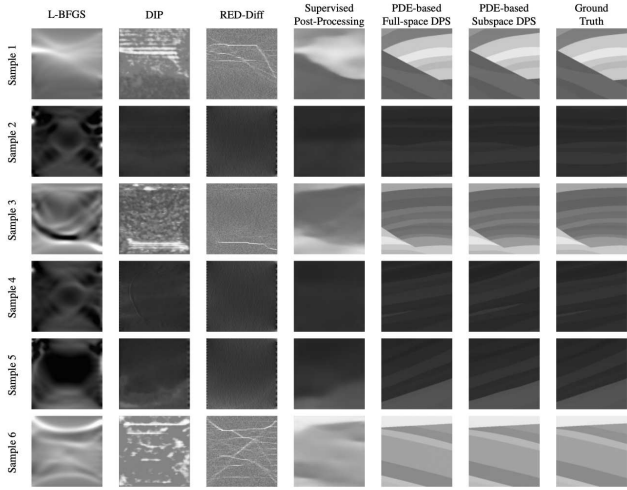


FIGURE 2. The reconstruction results for each Marmousi test sample are compared across comparison methods and ours, using measurements from horizontally placed sources and receivers.

Method	KIT4 (surround)		Marmousi (vertical)		Marmousi (horizontal)	
	32 → 64 → 128	64 → 128	32 → 64 → 128	64 → 128	32 → 64 → 128	64 → 128
PDE-based Subspace DPS						
Average Time (sec)	351.06	583.01	193.39	301.11	190.12	293.58
PDE-based Full-space DPS						
Average Time (sec)	(1060.75)	(1060.75)	(551.73)	(551.73)	(548.46)	(548.46)
Average Time Ratio	0.331	0.550	0.350	0.546	0.346	0.535

TABLE 2. The computational times of the PDE-based full-space and subspace DPS methods are compared on the Marmousi and KIT4 test datasets, respectively.

#### 4. CONCLUSIONS

We introduce a Diffusion Posterior Sampling (DPS) strategy for general PDE-based inverse problems, combined with a subspace diffusion technique for faster sampling. This method uses the adjoint-state approach with diffusion generative priors to solve PDE problems and allows PDE-constrained sampling across subspace sequences, reducing acquisition times. This framework can be generalized to general PDE-based problems once the adjoint equations are known. Experiments on Marmousi and KIT4 datasets, along with different geometries, demonstrate significant improvements in imaging quality over existing methods.

## REFERENCES

- [1] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. 2022. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 35:25683–25696.
- [2] Hyungjin Chung, Jeongsol Kim, Michael Thompson McCann, Marc Louis Klasky, and Jong Chul Ye. 2023. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*.
- [3] Klaus Deckelnick, Charles M Elliott, and Vanessa Styles. 2011. Numerical analysis of an inverse problem for the eikonal equation. *Numerische Mathematik*, 119:245–269.
- [4] Bowen Jing, Gabriele Corso, Renato Berlinghieri, and Tommi Jaakkola. 2022. Subspace diffusion generative models. In *European Conference on Computer Vision*, pages 274–289. Springer.
- [5] N. Rawlinsom, S. Pozgay and S. Fishwick, *Seismic tomography: A window into deep earth*, Physics of the Earth and Planetary Interiors (2010), vol. 178(3-4), pp. 101–135.
- [6] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR*.
- [7] Plamen Stefanov, Gunther Uhlmann, Andras Vasy, and Hanming Zhou. 2019. Travel time tomography. *Acta Mathematica Sinica, English Series*, 35(6):1085–1114.
- [8] Albert Tarantola. 2005. *Inverse problem theory and methods for model parameter estimation*. SIAM.

## Participants

**Prof. Dr. Rima Alaifari**

Departement Mathematik  
ETH-Zentrum  
Rämistrasse 101  
8092 Zürich  
SWITZERLAND

**Prof. Dr. Giovanni S. Alberti**

Machine Learning Genoa Center  
Department of Mathematics  
University of Genoa  
Via Dodecaneso 35  
16146 Genova  
ITALY

**Prof. Dr. Simon R. Arridge**

Department of Computer Science  
University College London  
Gower Street  
London WC1E 6BT  
UNITED KINGDOM

**Dr. Silvia Barbeiro**

Departamento de Matemática  
Universidade de Coimbra  
Largo D. Dinis  
3000-143 Coimbra  
PORTUGAL

**Dr. Julius Berner**

California Institute of Technology  
Pasadena CA 91125  
UNITED STATES

**Dr. Nicolas Boullé**

Imperial College London  
Huxley Building  
South Kensington Campus  
London SW7 2AZ  
UNITED KINGDOM

**Dr. Tatiana Bubba**

Università di Ferrara  
Via Machiavelli 35  
44121 Ferrara  
ITALY

**Prof. Dr. Chris J. Budd**

Dept. of Mathematical Sciences  
University of Bath  
Claverton Down  
Bath BA2 7AY  
UNITED KINGDOM

**Prof. Dr. Elena Celledoni**

Department of Mathematical Sciences  
Norwegian University of Science and  
Technology  
A. Getz vei 1  
7491 Trondheim  
NORWAY

**Prof. Dr. Maarten V. de Hoop**

Simons Chair in Computational and  
Applied Mathematics and Earth Science  
Rice University  
Houston TX 77005  
UNITED STATES

**Prof. Dr. Juan Carlos De los Reyes**

Centro de Modelización Matemática  
(MODEMAT)  
Edificio #12, 6to piso  
Escuela Politécnica Nacional  
Ladrón de Guevara E11-253  
170525 Quito  
ECUADOR

**Dr. Alexander Denker**

Department of Computer Science  
University College London  
66-72 Gower Street  
London WC1E 6EA  
UNITED KINGDOM

**Dr. Felix Dietrich**

Technische Universität München  
School of Computation, Information and  
Technology  
Boltzmannstr. 3  
85748 Garching bei München  
GERMANY

**Prof. Dr. Victorita Dolean-Maini**

Department of Mathematics and  
Computer Science  
Eindhoven University of Technology  
P.O. Box 513  
5600 MB Eindhoven  
NETHERLANDS

**Janek Gödeke**

Fachbereich 3  
Mathematik und Informatik  
Universität Bremen  
Postfach 330440  
28334 Bremen  
GERMANY

**Prof. Dr. Hanno Gottschalk**

Fachbereich Mathematik  
Technische Universität Berlin  
Straße des 17. Juni 136  
10623 Berlin  
GERMANY

**Prof. Dr. Eldad Haber**

Department of Mathematics  
University of British Columbia  
121-1984 Mathematics Road  
Vancouver BC V6T 1Z2  
CANADA

**Dr. Jiequn Han**

Flatiron Institute  
162 Fifth Avenue  
New York 10010  
UNITED STATES

**Prof. Dr. Bastian Harrach**

Institut für Mathematik  
Goethe-Universität Frankfurt  
Postfach 111932  
60054 Frankfurt am Main  
GERMANY

**Assoc. Prof. Dr. Andreas  
Hauptmann**

Research Unit of Mathematical Sciences  
University of Oulu  
FI-90014 University of Oulu  
Pentti Kaitekan katu 1  
P.O. Box 8000  
90014 Oulu  
FINLAND

**Nick Heilenkötter**

Fachbereich 3  
Mathematik und Informatik  
Universität Bremen  
Postfach 330440  
28334 Bremen  
GERMANY

**Dr. Johannes Hertrich**

University College London  
169 Euston Road  
NW12AE London  
UNITED KINGDOM

**Meira Iske**

Fachbereich 3  
Mathematik und Informatik  
Universität Bremen  
Bibliothekstraße 5  
Postfach 330440  
28359 Bremen  
GERMANY

**Dr. Bangti Jin**

Dept. of Mathematics  
The Chinese University of Hong Kong  
Shatin, N.T., Hong Kong N. A.  
CHINA

**Prof. Dr. Barbara Kaltenbacher**

Institut für Mathematik  
Alpen-Adria-Universität Klagenfurt  
Universitätsstrasse 65-67  
9020 Klagenfurt  
AUSTRIA

**Zeljko Kereta**

Department of Mathematics  
University College London  
London WC1E 6BT  
UNITED KINGDOM

**Dr. Yury Korolev**

Dept. of Mathematical Sciences  
University of Bath  
Bath BA2 7AY  
UNITED KINGDOM

**Dr. Lisa Kreusser**

Dept. of Mathematical Sciences  
University of Bath  
Claverton Down  
Bath BA2 7AY  
UNITED KINGDOM

**Yolanne Lee**

University College London  
London WC1E 6BT  
UNITED KINGDOM

**Prof. Dr. Peter Maaß**

FB 3 – Mathematik und Informatik  
Zentrum für Technomathematik  
Universität Bremen  
Postfach 330 440  
28334 Bremen  
GERMANY

**Dr. Georg Maierhofer**

Mathematical Institute  
University of Cambridge  
16, Mill Lane  
Cambridge CB2 1SB  
UNITED KINGDOM

**Davide Murari**

Department of Applied Mathematics  
and Theoretical Physics  
University of Cambridge  
Centre for Mathematical Sciences  
Wilberforce Rd  
Cambridge CB3 0WA  
UNITED KINGDOM

**Prof. Dr. Sebastian Neumayer**

TU Chemnitz  
Fakultät für Mathematik  
Straße der Nationen 62  
09111 Chemnitz  
GERMANY

**Dr. Tram Nguyen**

Max-Planck-Institut für  
Sonnensystemforschung  
Justus-von-Liebig-Weg 3  
37077 Göttingen  
GERMANY

**Jianfeng Ning**

Dept. of Mathematics  
The Chinese University of Hong Kong  
Shatin, N.T., Hong Kong N. T.  
CHINA

**Prof. Dr. Ozan Öktem**

Department of Mathematics  
KTH - Royal Institute of Technology  
Stockholm, Sweden  
10044 Stockholm  
SWEDEN

**Prof. Dr. Brynjulf Owren**

Department of Mathematical Sciences  
NTNU  
7491 Trondheim  
NORWAY

**Prof. Dr. Thomas Pock**

Institute of Computer Graphics  
and Vision  
Graz University of Technology  
Inffeldgasse 16/II  
8010 Graz  
AUSTRIA

**Dr. Zebang Shen**

Institute for Machine Learning,  
Department of Computer Science,  
ETH-Zürich  
Andreasstrasse 5,  
8050 Zürich  
SWITZERLAND

**James Rowbottom**

Mathematical Institute  
University of Cambridge  
16, Mill Lane  
Cambridge CB2 1SB  
UNITED KINGDOM

**Zakhar Shumaylov**

Centre for Mathematical Sciences  
University of Cambridge  
Wilberforce Road  
Cambridge CB3 0WB  
UNITED KINGDOM

**Christina Runkel**

Centre for Mathematical Sciences  
University of Cambridge  
Wilberforce Road  
Cambridge CB3 0WB  
UNITED KINGDOM

**Prof. Dr. Gabriele Steidl**

TU Berlin  
Institute of Mathematics  
Straße des 17. Juni  
10623 Berlin  
GERMANY

**Prof. Dr. Otmar Scherzer**

Fakultät für Mathematik  
Universität Wien  
Oskar-Morgenstern-Platz 1  
1090 Wien  
AUSTRIA

**Lukas Weigand**

DESY  
Deutsches Elektronen-Synchrotron  
Notkestr. 85  
22607 Hamburg  
GERMANY

**Prof. Dr. Carola-Bibiane Schönlieb**

Department of Applied Mathematics and  
Theoretical Physics (DAMTP)  
Centre for Mathematical Sciences  
Wilberforce Road  
Cambridge CB3 0WA  
UNITED KINGDOM

**Prof. Dr. Xiaoqun Zhang**

Institute of Natural Sciences  
Shanghai Jiao Tong University  
No. 6 Science Building  
800, Dongchuan Road  
Shanghai 200 240  
CHINA