
Automatic differentiation approach for reconstructing spectral functions with neural networks

Lingxiao Wang

Frankfurt Institute for Advanced Studies

Xidian-FIAS Joint Research Center
Frankfurt am Main, D-60438, Germany
lwang@fias.uni-frankfurt.de

Shuzhe Shi

Department of Physics
McGill University
Quebec H3A 2T8, Canada

Center for Nuclear Theory
Department of Physics and Astronomy
Stony Brook University
New York, 11784, USA
shuzhe.shi@stonybrook.edu

Kai Zhou

Frankfurt Institute for Advanced Studies
Frankfurt am Main, D-60438, Germany
zhou@fias.uni-frankfurt.de

Abstract

Reconstructing spectral functions from Euclidean Green's functions is an important inverse problem in physics. The prior knowledge for specific physical systems routinely offers essential regularization schemes for solving the ill-posed problem approximately. Aiming at this point, we propose an automatic differentiation framework as a generic tool for the reconstruction from observable data. We represent the spectra by neural networks and set chi-square as loss function to optimize the parameters with backward automatic differentiation unsupervisedly. In the training process, there is no explicit physical prior embedding into neural networks except the positive-definite form. The reconstruction accuracy is assessed through Kullback–Leibler(KL) divergence and mean square error(MSE) at multiple noise levels. It should be noted that the automatic differential framework and the freedom of introducing regularization are inherent advantages of the present approach and may lead to improvements of solving inverse problem in the future.

1 Introduction

The numerical solution to inverse problems is a vital area of research in many domains of science. In physics, especially quantum many-body physics, it's necessary to perform an analytic continuation of function from finite observations which however is ill-posed [1, 2, 3]. It is encountered for example, in Euclidean Quantum Field Theory (QFT) when one aims at rebuilding spectral functions based on some discrete data points along the Euclidean axis. More specifically, the inverse problem occurs when we take a non-perturbative Monte Carlo simulations (e.g., lattice QCD) and try to bridge the correlator data points with physical spectra [2, 3]. The knowledge of spectra will be further applied in transport process and non-equilibrium phenomena in heavy ion collisions [2, 4]. Moreover, the inverse problem of rebuilding spectral function is not unique to strong interaction many-body systems, but have similar counterparts in quantum liquid and superconductivity [1].

Related works In past two decades, the most common approach in such reconstruction project is Bayesian inference which is a classical statistical learning method. It comprises the extra prior knowledge from the physical domain about the spectral function, so as to regularize the inversion task [2, 5, 6]. For example, as two of axioms, the scale invariance and proper constant form prior are both embedded into the Bayesian approach with Shannon-Jaynes entropy, which is termed as maximum entropy method (MEM) [4, 6]. Besides, recent several studies have investigated reconstructing spectral functions through a neural network [7, 8, 9, 10, 11]. In a supervised learning framework, the prior knowledge is encoded in amounts of training data and the inverse transformation is explicitly unfolded through a training process [7, 8, 9]. To alleviate the dependence of redundant training data, there are also studies adopting the radial basis functions and Gaussian process [11, 12].

2 Problem statement

Our inverse problem set-up is based on a Fredholm equation of the first kind, which takes the following form,

$$g(t) = K \int_a^b f(s) ds, \quad (1)$$

and the problem is to reconstruct the function $f(s)$ given the continuous kernel function $K(t, s)$ and the function $g(t)$. In physical systems, $g(t)$ is always available in a discrete form. When dealing with a finite set of data points with finite uncertainty, the inverse transform becomes ill-conditioned or degenerated [9, 13]. In other words, the formal inversion is not a stable operation.

Källen–Lehmann(KL) spectral representation Henceforth in the paper, we discuss the uniqueness and accuracy of the spectral function by building the Källen–Lehmann(KL) representation [14],

$$D(p) = \int_0^1 K(p, \omega) \rho(\omega) d\omega = \int_0^1 \frac{\omega \rho(\omega)}{\omega^2 + p^2} \frac{d\omega}{\pi}. \quad (2)$$

Mock spectral functions are constructed using a superposed collection of Breit-Wigner peaks based on a parametrization obtained directly from one-loop perturbative quantum field theory [3, 7]. Each individual Breit-Wigner spectral function is given by,

$$\rho^{(\text{BW})}(\omega) = \frac{4A \omega}{(M^2 + \omega^2)^2 + 4 \omega^2}, \quad (3)$$

here M denotes the mass of the corresponding state, Γ is its width and A amounts to a positive normalization constant. The multi-peak structure is built by combining different single peak modules together.

3 Methods

In this section, we demonstrate vectorized formalism of our methodology which can be easily implemented by differential programming in Pytorch or other frameworks. For simplicity we take one N_p point for $D(p)$ observation as example. One can directly extend it to multiple data points by making summation over them in calculating the gradients.

Spectral representations we develop 3 architectures with different levels of non-local correlations among ρ_{ω_i} to represent the spectral functions with artificial neural networks(ANNs). The first form is **List**, it is equivalent to set $L = 1$ without bias node, meanwhile, the differentiable variables are $\vec{\rho} = [\rho_1, \rho_2, \dots, \rho_{N_\omega}]$ as Figure 1 left panel a) shown. If one approximates the integration over frequencies ω_i to be summation over N_ω points at fixed frequency interval $d\omega$, then it is suitable to the vectorized framework. The second representation is named as **NN**, in which we use L -layers neural network to represent the spectral function $\rho(\omega)$ with a constant input node $a^0 = C$ and multiple output nodes $a^L = [\rho_1, \rho_2, \dots, \rho_{N_\omega}]$. The width of the l -th layer is n_l , in which the correlation among discrete outputs is contained in a concealed form. The last way is to set input node as $a^0 = \omega_i$ and single output node as $a^L = \rho_i$. It is termed as point-to-point neural networks (**NN-P2P**), in which the continuity of function $\rho(\omega)$ is a regularization defined in domains of input and output.

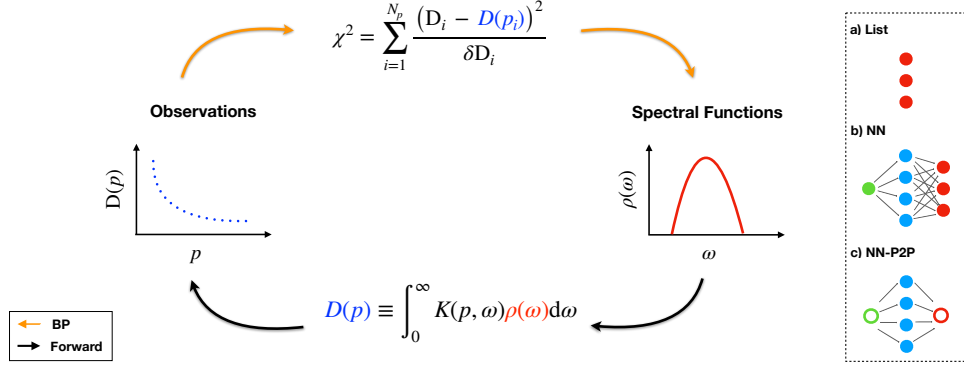


Figure 1: Automatic differential framework to reconstruct spectra from observations. Different architectures of representing spectrum with neural networks: a) **List**. 1-layers neural network which is equivalent to the list of spectrum. b) **NN**. L -layers neural networks and the output is list of spectrum. c) **NN-P2P**. L -layers neural networks and the end-to-end nodes are (ω_i, ρ_i) pairwise.

Automatic differentiation The output of above representations is $\vec{\rho} = [\rho_1, \rho_2, \dots, \rho_{N_\omega}]$, from which we can calculate the correlator as $D(p) = \sum_i^{N_\omega} \vec{\rho} \cdot K(p, \vec{\omega})$ with $\vec{\omega} = [\omega_1, \omega_2, \dots, \omega_{N_\omega}]$, where ‘ \cdot ’ represents element-wise product. After the forward process, we can get both $\vec{\rho}$ and Loss $L = \chi^2 = \sum_i^{N_p} (D_i - D(p_i))^2 / D(p_i)$, where D_i is observed data at p_i with N_p points. To optimize the parameters of presentations θ with loss function, we implement the backward propagation (BP). The gradients for layer- l is $\frac{\partial L}{\partial \theta^{[l]}} = [L]$ and the input for backward propagation is,

$$[L] = \frac{\partial L}{\partial D(p)} K(p, \vec{\omega}). \quad (4)$$

With iteration loops in backward direction the gradients, $[L] = \theta^{[l+1] \times [l+1]} \sigma^l(Z^{[l]})$ can be used to optimize parameters θ , where ‘ \times ’ represents the transpose, $\theta^{[l]}$ is weights matrix at layer- l , $Z^{[l]}$ is output of layer- l and $\sigma(\cdot)$ is the corresponding activation function.

Optimization strategy The components of the framework are differentiable and therefore amenable to gradient descent. Due to the feasibility of regularizers in neural network representations, the optimization makes use of the Adam algorithm [15] with weight decay¹. In training process, we obey an annealing strategy which is setting a tight regularization at beginning and loosen it repeatedly in first 20000 epochs. The weight decay rate set as 10^{-4} and learning rate is 10^{-3} for all cases. The smoothness regularization contributed to loss function is written as $\lambda_s \sum_{i=1}^{N_\omega} (\rho_i - \rho_{i-1})^2$. Tight initial regularization is $\lambda_s = 10^{-2}$. Besides the existing regularization of neural network itself, the only physical prior we enforce into the framework is the positive-definiteness of hadron spectral functions, which is introduced through using **Softplus** activation function at last layer as **Softplus**(x) = $\log(1 + e^x)$. It should be mentioned that the biases induced by using gradient descent-type optimizers are not avoided in our framework, but it could be improved by embedding ensemble learning strategies.

¹Weight decay is equivalent to L^2 regularization in stochastic gradient descent(SGD) when re-scaled by the learning rate [16].

4 Numerical Results

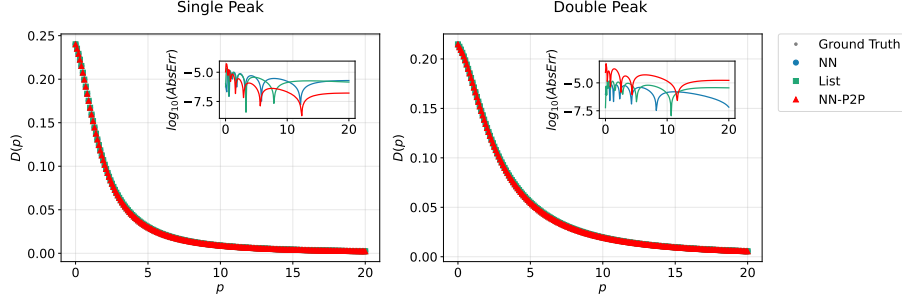


Figure 2: Performance of rebuilding correlation functions in two samples. The insert figures are showing absolute errors between observed correlation functions and the rebuilt.

In this section, to demonstrate the performance of our framework, we prepare two profiles of spectral functions from Eq. 3. In Figure 2, the left correlation function is from a single peak spectrum with $A = 1$, $\alpha = 0.3$, $M = 2.0$ and the right hand-side is from double peak profile with $A_1 = 1, A_2 = 1.5$, $\alpha_1 = 0.4$, $\alpha_2 = 0.5$, $M_1 = 2.5, M_2 = 5.0$. Three representations are marked by green, blue and red dots, which are plot in high consistencies with observed correlators. Besides, to imitate the real-world observable data, we add Gaussian-type noise into mock data with $D(p_i) = D(p_i) + noise$ and $noise = \mathcal{N}(0, \epsilon)$. The reconstruction absolute error reaches 10^{-6} magnitude in all representations in the case with noise = 10^{-6} . The corresponding rebuilt spectral functions are list in following Figure 3.

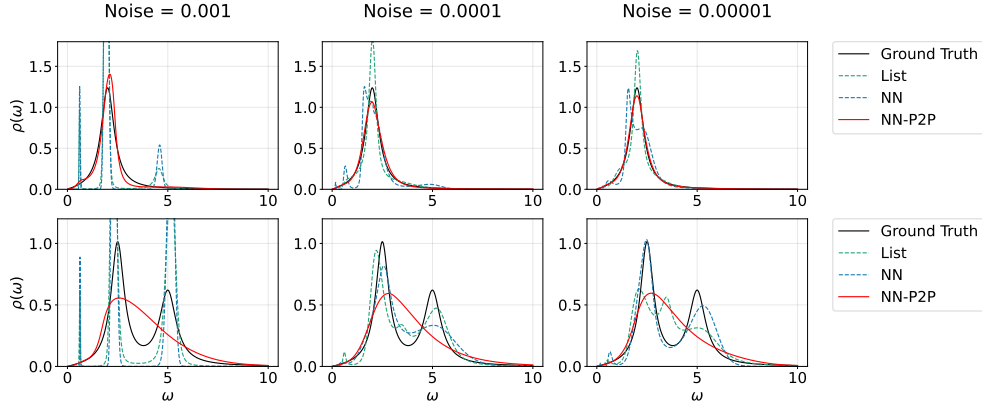


Figure 3: The predicted spectral functions from NN, List and NN-P2P. For left to right panels, different Gaussian noises are added to the correlation data with $\epsilon = 0.001, 0.0001$ and 0.00001 .

Three representations are marked by green, blue and red lines in Figure 3. They all show remarkable reconstruction performances for single peak case at noise level >0.0001 . In which, **List** and **NN** behave oscillations around zero-point under different noise backgrounds. The rebuilding spectrum from **NN-P2P** do not oscillate even with noise smaller than 0.0001 . This is especially important for such a task of extracting the transport coefficients from real-world lattice calculation data. Although the **List** representation has intense oscillations in double peak data, it successfully unfold the two peaks information from correlators even with noise $\epsilon = 0.001$.

To assess the reconstruction performance quantitatively, we introduce MSE and Kullback-Leibler(KL) divergence for rebuilt spectra q and the ground truth p as $D_{KL}(PkQ) = \int_0^T p(\omega) \log(p(\omega)/q(\omega)) d\omega$. At multi-magnitude noises, the NN-P2P keeps consistent performances compared with the other two representations. Although it misses the second peak which may appear in the case of bimodal, the calculations of different order momentum from spectral function will not be disturbed. Intuitively speaking, in NN-P2P representation, there are series of 1-order

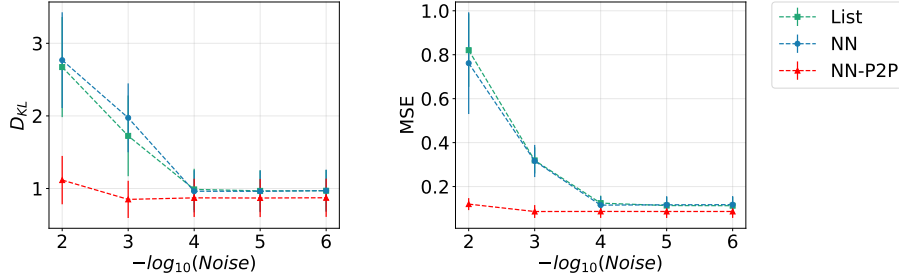


Figure 4: Evaluation to the reconstruction with KL divergence and MSE on mock data sets. Red triangle marks the NN-P2P, blue circles are NN representation and green cube labels the List.

differentiable modules between input ω node and output ρ node, in which the continuity of function $\rho(\omega)$ is naturally preserved. It makes the NN-P2P has a better performance in single peak case but failed in reconstructing double peaks which needs a looser restriction.

5 Conclusions

We present an automatic differentiation framework as a generic tool for unfolding spectral functions from observable data. The representations of spectra are used with 3 different neural network architectures, in which the modern optimization algorithm will be naturally employed. Although the inverse problems cannot be fully-solved in our framework, the remarkable performances of reconstructing spectral functions suggest that the framework and the freedom of introducing regularization are inherent advantages of the present approach. In future works, we will explore more neural network representations and a potential direction is to design specific neural networks with physics rules. This new paradigm provides us with a practical toolbox, in which solving inverse problem transfers into designing proper representations of solutions. It may lead to improvements in solving actual problems in e.g. optics, material design, and medical imaging.

Acknowledgments and Disclosure of Funding

We thank Heng-Tong Ding, Swagato Mukherjee and Gergely Endrödi for helpful discussions. The work is supported by (i) the BMBF under the ErUM-Data project (K. Z. and L. W.), (ii) the AI grant of SAMSON AG, Frankfurt (K. Z.), (iii) Natural Sciences and Engineering Research Council of Canada (S. S.), (iv) the Bourses d'excellence pour étudiants étrangers (PBEEE) from Le Fonds de Recherche du Québec - Nature et technologies (FRQNT) (S. S.), (v) U.S. Department of Energy, Office of Science, Office of Nuclear Physics, grant No. DE-FG88ER40388. (S. S.). K. Z. also thanks the donation of NVIDIA GPUs from NVIDIA

References

- [1] Mark Jarrell and J. E. Gubernatis. Bayesian inference and the analytic continuation of imaginary-time quantum Monte Carlo data. *Physics Reports*, 269(3):133–195, May 1996.
- [2] M. Asakawa, Y. Nakahara, and T. Hatsuda. Maximum entropy analysis of the spectral functions in lattice QCD. *Progress in Particle and Nuclear Physics*, 46(2):459–508, January 2001.
- [3] Ralf-Arno Tripolt, Philipp Gubler, Maksim Ulybyshev, and Lorenz von Smekal. Numerical analytic continuation of Euclidean data. *Computer Physics Communications*, 237:129–142, April 2019.
- [4] Alexander Rothkopf. Bayesian techniques and applications to QCD. *ArXiv190302293 Hep-Ph Physicsnucl-Th Physicsphysics*, March 2019.
- [5] Yannis Burnier and Alexander Rothkopf. Bayesian Approach to Spectral Function Reconstruction for Euclidean Quantum Field Theories. *Phys. Rev. Lett.*, 111(18):182003, October 2013.

- [6] Yannis Burnier, Olaf Kaczmarek, and Alexander Rothkopf. Static quark-antiquark potential in the quark-gluon plasma from lattice QCD. *Phys. Rev. Lett.*, 114(BI-TP-2014-21):082001, 2015.
- [7] Lukas Kades, Jan M. Pawłowski, Alexander Rothkopf, Manuel Scherzer, Julian M. Urban, Sebastian J. Wetzel, Nicolas Wink, and Felix P. G. Ziegler. Spectral reconstruction with deep neural networks. *Phys. Rev. D*, 102(9):096001, November 2020.
- [8] Hongkee Yoon, Jae-Hoon Sim, and Myung Joon Han. Analytic continuation via domain knowledge free machine learning. *Phys. Rev. B*, 98(24):245101, December 2018.
- [9] Romain Fournier, Lei Wang, Oleg V. Yazyev, and QuanSheng Wu. Artificial Neural Network Approach to the Analytic Continuation Problem. *Phys. Rev. Lett.*, 124(5):056401, February 2020.
- [10] Housen Li, Johannes Schwab, Stephan Antholzer, and Markus Haltmeier. NETT: Solving inverse problems with deep neural networks. *Inverse Problems*, 36(6):065005, June 2020.
- [11] Meng Zhou, Fei Gao, Jingyi Chao, Yu-Xin Liu, and Huichao Song. Application of radial basis functions neural networks in spectral functions. *ArXiv210608168 Hep-Ph Physicsnucl-Th*, June 2021.
- [12] Jan Horak, Jan M. Pawłowski, José Rodríguez-Quintero, Jonas Turnwald, Julian M. Urban, Nicolas Wink, and Savvas Zafeiropoulos. Reconstructing QCD Spectral Functions with Gaussian Processes. *ArXiv210713464 Hep-Lat Physicshep-Ph Physicshep-Th*, July 2021.
- [13] P. J. Caudrey. The inverse problem for a general $N \times N$ spectral equation. *Physica D: Nonlinear Phenomena*, 6(1):51–66, October 1982.
- [14] Michael E. Peskin and Dan V. Schroeder. *An Introduction To Quantum Field Theory*. Westview Press, Reading, Mass, first edition edition edition, October 1995.
- [15] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *ArXiv14126980 Cs*, January 2017.
- [16] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. *ArXiv171105101 Cs Math*, January 2019.

Checklist

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