

Fermilab

Inference of response functions with the help of machine learning algorithms

FERMILAB-PUB-25-0019-ETD

arXiv:2501.10583

This manuscript has been authored by Fermi Forward Discovery Group, LLC under Contract No. 89243024CSC000002 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

Inference of response functions with the help of machine learning algorithms

Doga Murat Kurkcuoglu and Gabriel N. Perdue
Fermi National Accelerator Laboratory, Batavia, IL, 60510, USA

Alessandro Roggero
Physics Department, University of Trento, Via Sommarive 14, I-38123 Trento, Italy
INFN-TIFPA Trento Institute of Fundamental Physics and Applications, Via Sommarive 14, I-38123 Trento, Italy and
InQubator for Quantum Simulation (IQuS), Department of Physics, University of Washington, Seattle, WA 98195, USA

Rajan Gupta
Theoretical Division T-2, Los Alamos National Laboratory, Los Alamos, NM 87545, USA
 (Dated: January 22, 2025)

Response functions are a key quantity to describe the near-equilibrium dynamics of strongly-interacting many-body systems. Recent techniques that attempt to overcome the challenges of calculating these *ab initio* have employed expansions in terms of orthogonal polynomials. We employ a neural network prediction algorithm to reconstruct a response function $S(\omega)$ defined over a range in frequencies ω . We represent the calculated response function as a truncated Chebyshev series whose coefficients can be optimized to reduce the representation error. We compare the quality of response functions obtained using coefficients calculated using a neural network (NN) algorithm with those computed using the Gaussian Integral Transform (GIT) method. In the regime where only a small number of terms in the Chebyshev series are retained, we find that the NN scheme outperforms the GIT method.

There are a number of problems of scientific interest where quantum computers may play a key role in finding solutions. In particular, quantum simulations offer a promising set of algorithms for computing quantities of interest that may be beyond the capabilities of classical computers in tractable times. Indeed, quantum simulation may be one of the first quantum computations to show practical quantum advantage [1]. Many of the possible use cases in High Energy Physics are discussed in recent reviews [2–4]. These include simulations of the real time dynamics of quantum systems and alleviating the sign problem limiting classical simulations. A motivation for this work is that quantum computers may provide valuable contributions to problems relevant to the long-baseline neutrino oscillation experiments [5, 6]. One of the prominent examples of this is direct *ab-initio* calculations of inclusive scattering cross sections on nuclear targets.

Several schemes for calculating the linear response of strongly correlated many-body systems using quantum computers have been proposed in the recent past [6–11]. An effective use of near term quantum computers for this goal relies on the extraction of suitably defined energy moments which contain the non-perturbative information and many-body observables, like scattering cross sections, can then be reconstructed with the help of classical algorithms [12–17]. This strategy has been employed in various forms in classical algorithms for a long time and remarkable progress has been achieved in recent years using classical many-body techniques like Quantum Monte Carlo [18–22] and Coupled Cluster theory [23–30]. These methods are well suited for calculations of static properties of nuclear many-body systems, however, information about the scattering cross sections, as encoded in the

so-called response function $S(\omega)$, are accessed through suitably defined integral transform. The dependence on the energy transfer ω of the scattering cross section is captured by the following response function

$$S(\omega) = \frac{\langle \Psi_0 | \hat{O}^\dagger \delta(\hat{H} - \omega) \hat{O} | \Psi_0 \rangle}{\langle \Psi_0 | \hat{O}^\dagger \hat{O} | \Psi_0 \rangle}, \quad (1)$$

with $|\Psi_0\rangle$ the ground state of the nuclear target, \hat{H} the Hamiltonian describing the target and \hat{O} an excitation operator describing the interaction between the target and the probe. With this definition, the integral of $S(\omega)$ over frequency is equal to one. An integral transform $\Phi(\omega)$ can then be defined as

$$\begin{aligned} \Phi(\omega) &= \int K(\omega, \nu) S(\nu) d\nu \\ &= \frac{\langle \Psi_0 | \hat{O}^\dagger K(\omega, \hat{H}) \hat{O} | \Psi_0 \rangle}{\langle \Psi_0 | \hat{O}^\dagger \hat{O} | \Psi_0 \rangle}, \end{aligned} \quad (2)$$

with the kernel $K(\omega, \nu)$ defining the integral transform.

The main advantage is $\Phi(\omega)$ can be estimated as a ratio between ground state expectation values provided the, so far arbitrary, kernel K is chosen appropriately.

For numerical schemes based on imaginary-time Monte Carlo sampling, an exponential kernel — leading to the Laplace transform [31, 32] — is typically employed, while for schemes working in occupation number basis — like the Coupled Cluster method — a Lorentzian or Gaussian kernel is used [13, 33, 34].

A popular numerical technique in condensed matter physics relies instead on integral kernels defined as finite linear combinations of orthogonal polynomials (see e.g. [35]). This approach has also been used recently for

nuclear physics applications on both classical [26, 27, 30] and quantum [16, 36] computing platforms, including the possibility of quantifying the errors induced by the integral transform [13, 37]. The central goal of this class of methods is to use the computed integral transform $\Phi(\omega)$ to reconstruct the original response function $S(\omega)$.

Unfortunately, this procedure leads to an ill-posed inversion problem that is, in general, extremely sensitive to small errors in the input data [38, 39]. This issue can be circumvented when the goal is to only estimate some observables connected with integrated properties of the response, e.g., the electric dipole polarizability of nuclei [40] or the impurity contribution to the thermal conductivity in the outer crust of neutron stars [41]. Since these can be obtained directly from the integral transform, one avoids the inversion step.

For the general case, multiple techniques have been developed to perform numerically stable inversions at the price of introducing systematic errors which are not always easy to quantify [42–47]. When the kernel satisfies certain regularity conditions it can be shown that stable reconstructions can be performed by limiting the final energy resolution in a way that takes into account the systematic errors [13]. For this purpose, it was found convenient to define integral kernels to be Σ -*accurate* with *resolution* Δ if the following holds [13]

$$\inf_{\omega_0 \in [E_0, E_{max}]} \int_{\omega_0 - \Delta/2}^{\omega_0 + \Delta/2} d\nu K(\nu, \omega_0) \geq 1 - \Sigma. \quad (3)$$

In this expression the frequency ω_0 over which one is extremizing is taken to be the full spectrum of the Hamiltonian $[E_0, E_{max}]$. The intuition behind this definition is that for a small value of Σ and an arbitrary but fixed frequency ω_0 , a kernel satisfying Eq. (3) has most of its support in an interval of size Δ around the target frequency. One can then expect the integral transform $\Phi(\omega)$ obtained employing such a kernel to be an approximation of the original response $S(\omega)$ smoothed over a frequency range of order Δ . This intuition can be made more rigorous and allows one to show that when using these integral kernels, it is possible to produce histograms of $S(\omega)$ with rigorous error bars [37].

In cases where either the required energy resolution is small or the preparation of the ground state $|\Psi_0\rangle$ is complicated, simulations on quantum computers could prove useful for carrying out the calculation in an efficient manner, thus opening the possibility of expanding the range of targets where the calculation of scattering cross sections becomes possible (see e.g. [2, 3] for recent reviews).

Inspired by recent work employing techniques from Machine Learning (ML) for the problem of inversion of integral transforms in Quantum Monte Carlo simulations [45–47], we here develop a numerical technique that uses Neural Networks to tailor the integral kernel used in the definition of $\Phi(\omega)$ to a specific application by training it over template response function generated from

an underlying model. The hope is that using such tailored kernel functions, and for fixed number of moments, a higher energy resolution will be achievable than using general purpose kernels like the Gaussian.

I. INTEGRAL TRANSFORM METHOD

In this section, we present in more detail the Integral Transform method using Chebyshev polynomials following the presentation in Ref. [37]. In order to simplify the notation used in the following derivation, it is convenient to introduce a normalized state obtained by perturbing the ground-state with \hat{O} as follows

$$|\psi_O\rangle = \frac{\hat{O}|\Psi_0\rangle}{\sqrt{\langle\Psi_0|\hat{O}^\dagger\hat{O}|\Psi_0\rangle}}. \quad (4)$$

Since we are interested in simulations where the many-body system is defined over a finite basis, the Hamiltonian of the system, \hat{H} , can be represented as a finite matrix of dimension N . In order to avoid complications when using Chebyshev polynomials, the spectrum of the Hamiltonian $[E_0, E_{max}]$ is mapped into the interval $[-1, 1]$ by defining a shifted and scaled Hamiltonian

$$\tilde{H} = \frac{2\hat{H} - (E_0 + E_{max})}{E_{max} - E_0}. \quad (5)$$

In situations when suitably tight bounds for E_0 and E_{max} are not available, we can instead use $\tilde{H} = \hat{H}/\Lambda$ with $\Lambda \geq \|\hat{H}\|$ as a computable upper bound on the spectral norm of the Hamiltonian (for applications on quantum computers this is always obtainable in an efficient way). We will assume from here on that this redefinition of the Hamiltonian of the system has been performed and refer to it, for brevity, by \tilde{H} .

The discrete spectrum allows us to express the response function as

$$S(\omega) = \sum_{k=1}^N s_k \delta(\lambda_k - \omega), \quad (6)$$

where λ_k are the N eigenvalues of \tilde{H} with eigenvectors $|\Psi_k\rangle$. The probabilities s_k are given by the squared overlaps

$$s_k = |\langle\Psi_k|\psi_O\rangle|^2 \quad \text{with} \quad \sum_{k=1}^N s_k = 1, \quad (7)$$

Using this notation, an integral transform of the response function $S(\omega)$ then becomes

$$\Phi(\omega) = \langle\psi_O|K(\omega, \tilde{H})|\psi_O\rangle = \sum_{k=1}^N s_k K(\omega, \lambda_k). \quad (8)$$

Given our goal to design an integral transform informed by the expected properties of the response function, we aim to determine an integral kernel K such that its corresponding integral transform minimizes the following cost function

$$C(\Delta) = \max_{\omega_0 \in [-1, 1]} \left| \int_{\omega_0 - \Delta/2}^{\omega_0 + \Delta/2} d\omega (\Phi(\omega) - S(\omega)) \right|. \quad (9)$$

The intuition behind this definition is the same that led to the introduction of the condition in Eq. (3) in Ref. [13]: when $C(\Delta)$ is small, the strength of the response function in an interval of size Δ centered anywhere within the range of the spectrum is well approximated by the strength evaluated using the integral transform $\Phi(\omega)$ instead. Indeed if the kernel is Σ -accurate with resolution Δ we have $C(\Delta) \leq \Sigma$.

In this work we attempt to optimize integral kernels expressed as a finite sum of Chebyshev polynomials $T_i(x)$ as follows

$$K_M(\omega, \nu) = \sum_{i,j=0}^{M-1} b_{ij} T_i(\omega) T_j(\nu). \quad (10)$$

Note that this is not equivalent to the standard construction used in the Kernel Polynomial Method [35] or the Gaussian Integral Transform [13, 37] since both of these include non polynomial contributions in ν . We chose this simplified class of kernels because the discussion of the optimization via neural-networks is easier in this case, but the strategy we propose in this work can be generalized easily to account for the differences. The main advantage of using Chebyshev polynomials to capture the ν dependence of the kernel is that the integral transform of the response function can be evaluated as

$$\Phi(\omega) = \sum_{i,j=0}^{M-1} b_{ij} T_i(\omega) \langle \psi_O | T_j(\hat{H}) | \psi_O \rangle, \quad (11)$$

where the expectation value appearing on the right hand side is the j -th Chebyshev moment of the Hamiltonian evaluated on the state $|\psi_O\rangle$, which we will denote as

$$m_j = \langle \psi_O | T_j(\hat{H}) | \psi_O \rangle. \quad (12)$$

The use of Chebyshev polynomials as a basis is particularly convenient since the moments m_j which encode the information about the response can be calculated efficiently on quantum computers [13, 48–50] and, at least for some systems, approximated accurately using Coupled Cluster theory [26, 27, 30].

II. NEURAL NETWORK

In this section, we describe the use of neural networks (NN) in this problem. NNs are useful in making predictions in physics to find the thermodynamic quantities, or

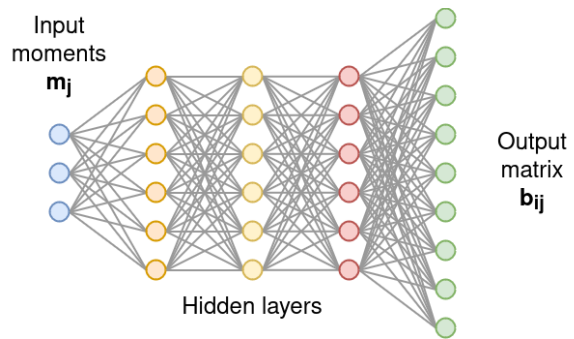


FIG. 1. A schematic of our inference neural network algorithm. The inputs are the M moments m_j and the output is the M^2 component b_{ij} vector.

to infer the quantum state based on the circuit training data. Here, we use the NN to infer the response signal from a few given moments, m_j . In order to compute the m_j moments from Eq. (12), only the $|\psi_O\rangle$ state, and the Hamiltonian \hat{H} are needed. The elements b_{ij} of the coefficient matrix defining the integral kernel in Eq. (10) are left as unknown parameters that will be optimized and predicted by the neural network.

Ideally, the optimization procedure would work as follows. First, generate a set of response functions $S^n(\omega)$ that are used for training, validation and testing. For each $S^n(\omega)$ in the set, compute the corresponding Chebyshev moments \vec{m}^n . Then choose a target resolution Δ_T and a budget of moments M and determine the optimal response function as a map

$$\vec{m} \longrightarrow b_{ij}^{opt}(\vec{m}) \quad (13)$$

depending on an input set of M Chebyshev moments \vec{m} . The goal being to create a multilayer NN, train it with m_j moments and infer the b_{ij} matrix from it.

In this work we represent this map using a multilayer perceptron with three hidden layers as depicted in the diagram in Fig. 1 and described in Table I. In all, the NN has a three hidden, one input and one output MLP layers. The input layer with M nodes normalizes the m_j moments. The three hidden layers use ReLU activation functions and have 30, 25 and 40 nodes, respectively. We use Adam with a variable learning rate to minimize the cost function of the NN.

In principle, the NN architecture may be optimized, but for initial demonstration purposes an easy-to-understand design is desirable. Similarly, hyperparameter optimization would likely find different preferred values than the ones presented in Table I. However, in a given application, users will likely be sensitive to data generation costs and processes, and those features will likely differ from this case. For demonstration purposes, we chose one set of parameters based on intuition and experience. Very little tuning was done as the goal is not to surpass a state-of-the-art benchmark, but rather to establish that the technique functions as hypothesized.

L_1	30
L_2	25
L_3	40
learning rate	0.00025
batch size	80
epochs	300
number of data for training	2000
number of data for validation	400
number of data for test	100

TABLE I. Parameters of the NN used in this work. L_1 , L_2 and L_3 are the number of hidden nodes in each layer. We used the Adam optimizer, choose a small learning rate and a large number of epochs to have smooth controlled convergence. The split of the 2500 data points for each (M, Δ) is given in the last three lines.

The cost function $C(\Delta)$ in Eq. (9) is rather demanding to evaluate precisely. In order to approximately minimize the cost function over ω_0 we can calculate this quantity on a uniform grid of points with spacing $d < \Delta$. For simplicity, we pick K points so that the lattice spacing in the energy interval $[-1, 1]$ is $d = 2/K$, and we choose $\Delta = 2L/K$ for some integer $1 \leq L \ll K$. This means that an interval of size Δ is given by points L units apart and the cost function can be then approximated as

$$C(\Delta) = \max_{l=0, \dots, K} \left| \int_{-1+ld}^{-1+(l+L)d} d\omega (\Phi(\omega) - S(\omega)) \right|. \quad (14)$$

The integral inside the absolute value can then be done exactly integrating separately the response function $S(\omega)$ from Eq. (6) and the Chebyshev moments $T_i(\omega)$ in the expansion of the integral transform Eq. (11). During the NN training we opted to use instead an upperbound $C_U(\Delta)$, which can be evaluated more efficiently and is also smoother. In order to obtain $C_U(\Delta)$ we first notice that the cost function in Eq. (14) can be expressed as $C(\Delta) = \|\vec{C}\|_\infty$ where $\|\cdot\|_\infty$ denotes the infinity vector norm and

$$C_l = \int_{-1+ld}^{-1+(l+L)d} d\omega (\Phi(\omega) - S(\omega)). \quad (15)$$

We can then introduce $C_U(\Delta)$ as

$$C_U(\Delta) = \|\vec{C}\|_2 \geq \|\vec{C}\|_\infty = C(\Delta), \quad (16)$$

where $\|\cdot\|_2$ is the standard vector 2 norm. The main advantage of this change in cost function is that $C_U^2(\Delta)$ is a quadratic function of the coefficient matrix b_{ij} and can be minimized straightforwardly using the least squares method. The resulting optimal matrix $b_{ij}^{(min)}$ is used as the target for the NN training with a mean square error (MSE) cost function.

III. RESPONSE FUNCTIONS

For the numerical calculations carried out in this work, the dimension of the Hamiltonian matrix is fixed to $N = 200$. These 200 energy eigenvalues, λ_k , with $k = 1, \dots, N$, are chosen using a uniform random sampling in the full energy interval normalized to $[-1, 1]$. For each N eigenvalues $\{\lambda_k\}$, $N_S = 2500$ data points, labeled λ_k^n with $n = 1, \dots, N_S$ are generated as described below.

Similarly, for each set of eigenvalues, $\{\lambda_k\}$, we generate a response function

$$S^n(\omega) = \sum_{k=1}^N s_k^n \delta(\lambda_k^n - \omega), \quad (17)$$

by computing the probabilities s_k^n according to a distribution that resembles the typical features of the physical response function we are interested in. For applications of this scheme to study the nuclear response in the quasi-elastic regime, which is particularly important for planned and ongoing neutrino-nucleus experiments, we use a model based on a skewed Gaussian function

$$S_S(\omega; \mu, \sigma, \alpha, \Gamma) = \Gamma e^{-\frac{(\omega-\mu)^2}{2\sigma^2}} \left[1 + \operatorname{erf} \left(\frac{\omega\alpha}{\sigma\sqrt{2}} \right) \right], \quad (18)$$

depending on 4 parameters: a location μ , a width σ , a skewness parameter α and a normalization constant Γ that is determined by satisfying (19).

The 2500 data points, and consequently the individual response functions $S^n(\omega)$, are generated by choosing fifty values each for α and σ . The fifty α are chosen uniformly within the range $\alpha \in [0, 2.5]$ to mimic a stronger high energy tail in the response. The fifty σ are chosen uniformly within $\sigma \in [\sigma_{\min}, 0.4]$ for a suitable σ_{\min} that is chosen to be $\sigma_{\min} = \Delta$. This choice is motivated by the fact that for signals with $\sigma < \Delta$, a good approximation of the response function that minimizes our cost $C(\Delta)$ can be obtained using only the first two moments m_0 and m_1 leading to an undetermined optimization problem. For each of these 2500 data points, μ is chosen randomly within the interval $\mu \in [-0.3, 0.3]$ ensuring most of the strength is in the central region of the energy range.

For each response function S^n , the individual probabilities s_k^n are constructed as

$$\widetilde{s}_k^n = S_S(\lambda_k^n; \mu_n, \sigma_n, \alpha_n, 1) \Rightarrow s_k^n = \frac{\widetilde{s}_k^n}{\sum_k \widetilde{s}_k^n}, \quad (19)$$

in order to ensure the normalization condition in Eq. (7).

The corresponding integral transform is then given by

$$\Phi_M^n(\omega; b_{ij}) = \sum_{i,j=0}^M b_{ij} T_i(\omega) m_j^n, \quad (20)$$

where the m_j^n moments are calculated independently for each $S^n(\omega)$. This gives a total of $M \times N_S$ numbers. In other words, for each of the N_S points, the M moments,

m_j , are calculated directly from the model response function S^n as follows:

$$m_j = \int_{-1}^1 \frac{d\omega}{\sqrt{1-\omega^2}} S^n(\omega) T_j(\omega). \quad (21)$$

This integral becomes a sum when Eq. (17) is used. The calculation and comparison of the cost function using the trained NN and the GIT [13, 37] algorithm starts with these $M \times N_S$ numbers.

IV. NUMERICAL RESULTS

The choice $N_s = 2500$ data points for each fixed (M, Δ) , selected as explained above, was a reasonable match to our NN's capacity. This dataset was split into 2000 training, 400 validation, and 100 test points, and the NN trained using the hyperparameters presented in Table I. The response signal is inferred from the training by minimizing $C_U(\Delta)$ defined in Eq. (16) with $K = 70$.

The results for the cost function C_Δ obtained with the integral kernel expressed in terms of the NN are compared with the general purpose GIT method based on a Gaussian kernel [13, 37]. The truncated kernel that approximates a Gaussian with variance λ is expressed in terms of Chebyshev polynomials as

$$K_M^{(G)}(\omega, \nu; \lambda) = \sum_{i=0}^{M-1} c_i^{[M,M]}(\nu; \lambda) T_i(\omega) \quad (22)$$

with coefficient functions $c_i^{[M,M]}(\nu; \lambda)$ chosen to provide a good representation of a Gaussian over M moments and given explicitly in Eq.(A20) of Ref. [37]. For a given choice of resolution Δ and number of moments M , we obtain the value of λ to minimize the approximation error made when the response function is replaced with an integral transform in Eq. (3) where Σ accounts for the contribution of the tails (for more details see Ref. [37]).

Figure 2 shows a comparison of the results obtained with the GIT method and the new NN-based technique. The two panels show the 90% confidence intervals for the value of the cost function, Eq. (16), obtained using the 100 data points reserved for testing, as a function of the number of moments M used in the reconstruction. The left panel corresponds to a wider resolution $\Delta = 12/70 \approx 0.1714$ while the right panel shows results for a narrower resolution $\Delta = 4/70$. As Δ is decreased, the GIT method provides increasingly better results when a large number of moments are included, i.e., the right panel shows this happens for $M > 60$ for $\Delta = 4/70$. On the other hand, the new NN approach shows saturation early, that will eventually be outperformed by the GIT method at large values of M . To summarize our main result: for narrower Δ , as shown for $\Delta = 4/70$, the output of the NN method saturates for $M \approx 10$. Thus, when doing calculations keeping a small number of moments, $M \lesssim 15 - 20$, the NN method is able to provide a good integral transform

much faster. We can interpret the result as a consequence of the additional information employed in the training of the network which is not used for a general purpose kernel like the Gaussian used in GIT.

V. CONCLUSIONS

Several state of the art numerical techniques for studying the linear response of strongly correlated systems in nuclear physics rely on the physical information encoded in energy moments of a perturbed initial state. In both classical and quantum algorithms, the use of Chebyshev polynomials for the definition of these energy weights, and the subsequent reconstruction of the nuclear response function, has been proposed as an effective and controllable strategy. In this work we explored the use of Neural Networks to increase the allowed energy resolution when working with a limited number of input moments by relying on the availability of a suitable training set of response functions that can be motivated by the physics of the problem. Employing this physics inspired training, we were able to show that when a limited amount of information (in the form of Chebyshev moments) is available, NN-based strategies can outperform more general reconstruction schemes (like the GIT) that are more agnostic in design and do not directly exploit the features present in physically realistic situations.

A. Future work

The goal of this work was not to produce the most optimal ML model or to outperform a published state of the art (SOTA) result. Instead, our primary focus was on demonstrating the technique works under at least some set of reasonable conditions, and from that perspective, using a simple and easy-to-understand ML algorithm was suitable. We, however, outline a number of improvements that should be considered when deploying this technique in practice, and for future research.

First, it would be appropriate to use a process like neural architecture search (see [51] for a recent review) to optimize the number and size of the hidden layers, or to consider the use of specialized layers. Furthermore, there are multiple hyperparameters (e.g., learning rate, learning rate schedule, batch size, etc.) that should be tuned using an efficient search strategy (see, e.g. [52] and similar codes).

The next obvious improvement would be to enlarge the training dataset (and increase model capacity accordingly). Our dataset was chosen for ease of generation on laptop-scale resources, but a modest investment of time would parallelize the code for dataset generation and the analysis, and allow the utilization of high performance computing (HPC) resources to dramatically increase the dataset size while still getting results in very little “wall-clock” time.

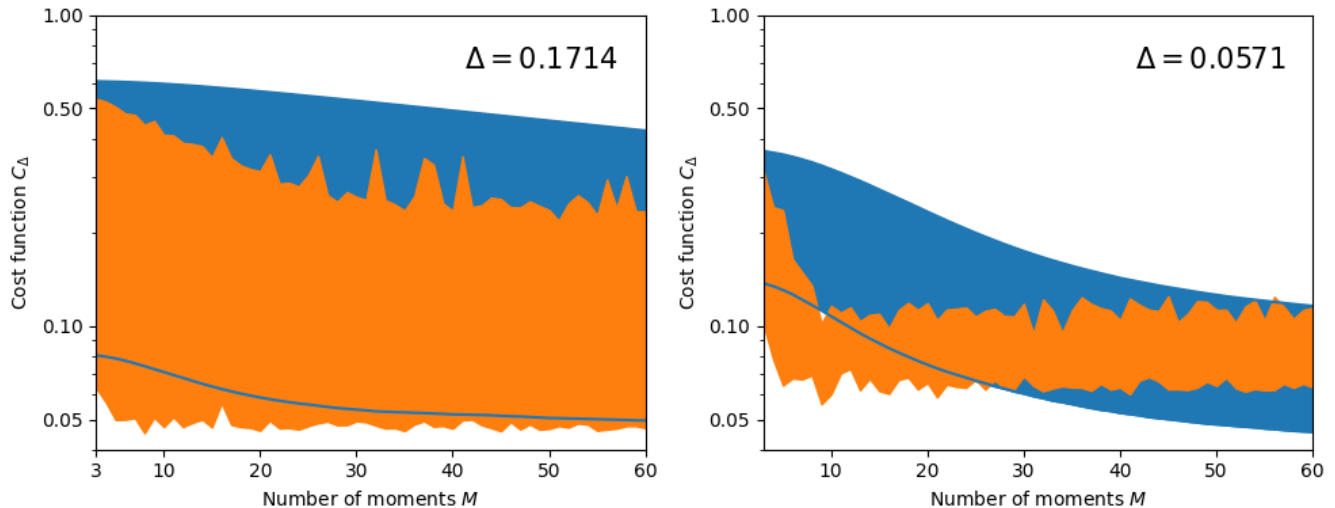


FIG. 2. (Color online) The cost function vs. the number of moments M for two values of resolution: $\Delta = 0.1714$ (left panel) and $\Delta = 0.0571$ (right panel). The orange band shows the 90% confidence interval of results obtained using the trained NN while the blue band corresponds to the same confidence interval over the same 100 test data points but for results obtained using the GIT method [13, 37].

Finally, there are interesting opportunities for architectural experimentation. For example, one could deploy an auto-encoder structure to find an encoding into a fixed latent dimensionality that could improve feature selection for a variety of tasks. We could also study linear combinations of moments, and study symmetries in the dataset and try to engineer network layers built to keep those symmetries invariant (a so-called “physics-inspired” network).

ACKNOWLEDGEMENTS

We thank Joseph Carlson for many discussions. This document was prepared using the resources of the Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy (DOE), Office of Science, HEP User

Facility. Fermilab is managed by Fermi Research Alliance, LLC (FRA), acting under Contract No. DE-AC02-07CH11359. This work was supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Quantum Science Center. This manuscript has been authored by Fermi Forward Discovery Group, LLC under Contract No. 89243024CSC000002 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

D.M.K. and G.N.P. were supported for this work by the DOE/HEP QuantISED program grant “HEP Machine Learning and Optimization Go Quantum,” identification number 0000240323. R.G. was supported by the U.S. Department of Energy, Office of Science, High Energy Physics, QuantISED Contract KA2401032 and AI/ML for HEP contract KA2401045 to Los Alamos National Laboratory.

-
- [1] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, Toward the first quantum simulation with quantum speedup, *Proceedings of the National Academy of Sciences* **115**, 9456 (2018).
- [2] N. Klco, A. Roggero, and M. J. Savage, Standard model physics and the digital quantum revolution: thoughts about the interface, *Reports on Progress in Physics* **85**, 064301 (2022).
- [3] C. W. Bauer, Z. Davoudi, A. B. Balantekin, T. Bhattacharya, M. Carena, W. A. de Jong, P. Draper, A. El-Khadra, N. Gemelke, M. Hanada, D. Kharzeev, H. Lamm, Y.-Y. Li, J. Liu, M. Lukin, Y. Meurice, C. Monroe, B. Nachman, G. Pagano, J. Preskill, E. Rinaldi, A. Roggero, D. I. Santiago, M. J. Savage, I. Siddiqi, G. Siopsis, D. Van Zanten, N. Wiebe, Y. Yamauchi, K. Yeter-Aydeniz, and S. Zorzetti, Quantum simulation for high-energy physics, *PRX Quantum* **4**, 027001 (2023).
- [4] A. Di Meglio, K. Jansen, I. Tavernelli, C. Alexandrou, S. Arunachalam, C. W. Bauer, K. Borras, S. Carrazza, A. Crippa, V. Croft, R. de Putter, A. Delgado, V. Dunjko, D. J. Egger, E. Fernández-Combarro, E. Fuchs, L. Funcke, D. González-Cuadra, M. Grossi, J. C. Halimeh, Z. Holmes, S. Kühn, D. Lacroix, R. Lewis, D. Lucchesi, M. L. Martinez, F. Meloni, A. Mezzacapo, S. Montangero, L. Nagano, V. R. Pascuzzi, V. Radescu, E. R. Ortega, A. Roggero, J. Schuhmacher, J. Seixas, P. Silvi, P. Spentzouris, F. Tacchino, K. Temme, K. Terashi, J. Tura, C. Tüysüz, S. Vallecorsa, U.-J. Wiese, S. Yoo,

- and J. Zhang, Quantum computing for high-energy physics: State of the art and challenges, *PRX Quantum* **5**, 037001 (2024).
- [5] A. Roggero, A. C. Y. Li, J. Carlson, R. Gupta, and G. N. Perdue, Quantum computing for neutrino-nucleus scattering, *Phys. Rev. D* **101**, 074038 (2020).
- [6] A. Baroni, J. Carlson, R. Gupta, A. C. Y. Li, G. N. Perdue, and A. Roggero, Nuclear two point correlation functions on a quantum computer, *Phys. Rev. D* **105**, 074503 (2022).
- [7] A. Chiesa, F. Tacchino, M. Grossi, P. Santini, I. Tavernelli, D. Gerace, and S. Carretta, Quantum hardware simulating four-dimensional inelastic neutron scattering, *Nature Physics* **15**, 455 (2019).
- [8] A. Roggero and J. Carlson, Dynamic linear response quantum algorithm, *Phys. Rev. C* **100**, 034610 (2019).
- [9] A. Ciavarella, Algorithm for quantum computation of particle decays, *Phys. Rev. D* **102**, 094505 (2020).
- [10] Y. Tong, D. An, N. Wiebe, and L. Lin, Fast inversion, preconditioned quantum linear system solvers, fast green's-function computation, and fast evaluation of matrix functions, *Phys. Rev. A* **104**, 032422 (2021).
- [11] F. Libbi, J. Rizzo, F. Tacchino, N. Marzari, and I. Tavernelli, Effective calculation of the green's function in the time domain on near-term quantum processors, *Phys. Rev. Res.* **4**, 043038 (2022).
- [12] R. D. Somma, Quantum eigenvalue estimation via time series analysis, *New Journal of Physics* **21**, 123025 (2019).
- [13] A. Roggero, Spectral-density estimation with the gaussian integral transform, *Phys. Rev. A* **102**, 022409 (2020).
- [14] S. Lu, M. C. Bañuls, and J. I. Cirac, Algorithms for quantum simulation at finite energies, *PRX Quantum* **2**, 020321 (2021).
- [15] L. Lin and Y. Tong, Heisenberg-limited ground-state energy estimation for early fault-tolerant quantum computers, *PRX Quantum* **3**, 010318 (2022).
- [16] O. Kiss, M. Grossi, and A. Roggero, Quantum error mitigation for fourier moment computation (2024), arXiv:2401.13048 [quant-ph].
- [17] O. Kiss, U. Azad, B. Requena, A. Roggero, D. Wakeham, and J. M. Arrazola, Early fault-tolerant quantum algorithms in practice: Application to ground-state energy estimation (2024), arXiv:2405.03754 [quant-ph].
- [18] A. Lovato, S. Gandolfi, J. Carlson, S. C. Pieper, and R. Schiavilla, Electromagnetic and neutral-weak response functions of ${}^4\text{He}$ and ${}^{12}\text{C}$, *Phys. Rev. C* **91**, 062501 (2015).
- [19] A. Lovato, S. Gandolfi, J. Carlson, S. C. Pieper, and R. Schiavilla, Electromagnetic response of ${}^{12}\text{C}$: A first-principles calculation, *Phys. Rev. Lett.* **117**, 082501 (2016).
- [20] S. Pastore, J. Carlson, S. Gandolfi, R. Schiavilla, and R. B. Wiringa, Quasielastic lepton scattering and back-to-back nucleons in the short-time approximation, *Phys. Rev. C* **101**, 044612 (2020).
- [21] A. Lovato, J. Carlson, S. Gandolfi, N. Rocco, and R. Schiavilla, Ab initio study of (ν_e, ℓ^-) and $(\bar{\nu}_e, \ell^+)$ inclusive scattering in ${}^{12}\text{C}$: Confronting the miniboone and t2k ccqe data, *Phys. Rev. X* **10**, 031068 (2020).
- [22] L. Andreoli, J. Carlson, A. Lovato, S. Pastore, N. Rocco, and R. B. Wiringa, Electron scattering on $a = 3$ nuclei from quantum monte carlo based approaches, *Phys. Rev. C* **105**, 014002 (2022).
- [23] S. Bacca, N. Barnea, G. Hagen, G. Orlandini, and T. Papenbrock, First principles description of the giant dipole resonance in ${}^{16}\text{O}$, *Phys. Rev. Lett.* **111**, 122502 (2013).
- [24] S. Bacca, N. Barnea, G. Hagen, M. Miorelli, G. Orlandini, and T. Papenbrock, Giant and pigmy dipole resonances in ${}^4\text{He}$, ${}^{16,22}\text{O}$, and ${}^{40}\text{Ca}$ from chiral nucleon-nucleon interactions, *Phys. Rev. C* **90**, 064619 (2014).
- [25] J. E. Sobczyk, B. Acharya, S. Bacca, and G. Hagen, Ab initio computation of the longitudinal response function in ${}^{40}\text{Ca}$, *Phys. Rev. Lett.* **127**, 072501 (2021).
- [26] J. E. Sobczyk, S. Bacca, G. Hagen, and T. Papenbrock, Spectral function for ${}^4\text{He}$ using the chebyshev expansion in coupled-cluster theory, *Phys. Rev. C* **106**, 034310 (2022).
- [27] J. E. Sobczyk and S. Bacca, ${}^{16}\text{O}$ spectral function from coupled-cluster theory: applications to lepton-nucleus scattering (2023), arXiv:2309.00355 [nucl-th].
- [28] J. E. Sobczyk, B. Acharya, S. Bacca, and G. Hagen, ${}^{40}\text{Ca}$ transverse response function from coupled-cluster theory, *Phys. Rev. C* **109**, 025502 (2024).
- [29] B. Acharya, J. E. Sobczyk, S. Bacca, G. Hagen, and W. Jiang, ${}^{16}\text{O}$ electroweak response functions from first principles (2024), arXiv:2410.05962 [nucl-th].
- [30] J. E. Sobczyk, W. Jiang, and A. Roggero, Spin response of neutron matter in ab initio approach (2024), arXiv:2407.20986 [nucl-th].
- [31] J. Carlson and R. Schiavilla, Euclidean proton response in light nuclei, *Phys. Rev. Lett.* **68**, 3682 (1992).
- [32] J. Carlson, S. Gandolfi, F. Pederiva, S. C. Pieper, R. Schiavilla, K. E. Schmidt, and R. B. Wiringa, Quantum monte carlo methods for nuclear physics, *Rev. Mod. Phys.* **87**, 1067 (2015).
- [33] V. D. Efros, W. Leidemann, and G. Orlandini, Response functions from integral transforms with a lorentz kernel, *Physics Letters B* **338**, 130 (1994).
- [34] V. D. Efros, W. Leidemann, G. Orlandini, and N. Barnea, The lorentz integral transform (lit) method and its applications to perturbation-induced reactions, *Journal of Physics G: Nuclear and Particle Physics* **34**, R459–R528 (2007).
- [35] A. Weiße, G. Wellein, A. Alvermann, and H. Fehske, The kernel polynomial method, *Rev. Mod. Phys.* **78**, 275 (2006).
- [36] J. Hartse and A. Roggero, Faster spectral density calculation using energy moments, *Eur. Phys. J. A* **59**, 41 (2023).
- [37] J. E. Sobczyk and A. Roggero, Spectral density reconstruction with chebyshev polynomials, *Phys. Rev. E* **105**, 055310 (2022).
- [38] W. Glöckle and M. Schwamb, On the ill-posed character of the lorentz integral transform, *Few-Body Systems* **46**, 55–62 (2009).
- [39] N. Barnea, V. D. Efros, W. Leidemann, and G. Orlandini, The lorentz integral transform and its inversion, *Few-Body Systems* **47**, 201–206 (2010).
- [40] M. Miorelli, S. Bacca, N. Barnea, G. Hagen, G. R. Jansen, G. Orlandini, and T. Papenbrock, Electric dipole polarizability from first principles calculations, *Phys. Rev. C* **94**, 034317 (2016).
- [41] A. Roggero and S. Reddy, Thermal conductivity and impurity scattering in the accreting neutron star crust, *Physical Review C* **94**, 10.1103/physrevc.94.015803 (2016).
- [42] R. N. Silver, D. S. Sivia, and J. E. Gubernatis, Maximum-entropy method for analytic continuation of quantum

- monte carlo data, Phys. Rev. B **41**, 2380 (1990).
- [43] E. Vitali, M. Rossi, L. Reatto, and D. E. Galli, Ab initio low-energy dynamics of superfluid and solid ^4He , Phys. Rev. B **82**, 174510 (2010).
- [44] Y. Burnier and A. Rothkopf, Bayesian approach to spectral function reconstruction for euclidean quantum field theories, Phys. Rev. Lett. **111**, 182003 (2013).
- [45] L. Kades, J. M. Pawłowski, A. Rothkopf, M. Scherzer, J. M. Urban, S. J. Wetzel, N. Wink, and F. P. G. Ziegler, Spectral reconstruction with deep neural networks, Phys. Rev. D **102**, 096001 (2020).
- [46] K. Raghavan, P. Balaprakash, A. Lovato, N. Rocco, and S. M. Wild, Machine-learning-based inversion of nuclear responses, Phys. Rev. C **103**, 035502 (2021).
- [47] K. Raghavan and A. Lovato, Uncertainty quantification-enabled inversion of nuclear euclidean responses (2023), arXiv:2310.18756 [nucl-th].
- [48] A. M. Childs, R. Kothari, and R. D. Somma, Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, SIAM Journal on Computing **46**, 1920 (2017).
- [49] S. Subramanian, S. Brierley, and R. Jozsa, Implementing smooth functions of a hermitian matrix on a quantum computer, Journal of Physics Communications **3**, 065002 (2019).
- [50] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC '19 (ACM, 2019).
- [51] C. White, M. Safari, R. Sukthanker, B. Ru, T. Elsken, A. Zela, D. Dey, and F. Hutter, Neural architecture search: Insights from 1000 papers (2023), arXiv:2301.08727 [cs.LG].
- [52] P. Balaprakash, M. Salim, T. D. Uram, V. Vishwanath, and S. M. Wild, Deephyper: Asynchronous hyperparameter search for deep neural networks, in *2018 IEEE 25th International Conference on High Performance Computing (HiPC)* (2018) pp. 42–51.