Predicting quantum potentials by deep neural network and metropolis sampling

Rui Hong[1], Peng-Fei Zhou[1], Bin Xi[2], Jie Hu[1], An-Chun Ji[1] and Shi-Ju Ran[1]⋆

1 Department of Physics, Capital Normal University, Beijing 100048, China
2 College of Physics Science and Technology, Yangzhou University, Yangzhou 225002, China

⋆ sjran@cnu.edu.cn

Abstract

The hybridizations of machine learning and quantum physics have caused essential impacts to the methodology in both fields. Inspired by quantum potential neural network, we here propose to solve the potential in the Schrödinger equation provided the eigenstate, by combining Metropolis sampling with deep neural network, which we dub as Metropolis potential neural network (MPNN). A loss function is proposed to explicitly involve the energy in the optimization for its accurate evaluation. Benchmarking on the harmonic oscillator and hydrogen atom, MPNN shows excellent accuracy and stability on predicting not just the potential to satisfy the Schrödinger equation, but also the eigen-energy. Our proposal could be potentially applied to the *ab-initio* simulations, and to inversely solving other partial differential equations in physics and beyond.

Copyright R. Hong *et al.*
This work is licensed under the Creative Commons
Attribution 4.0 International License.
Published by the SciPost Foundation.

1 Introduction

In recent years, machine learning (ML) has been increasingly applied to the field of quantum physics [1]. On one hand, it provides alternative or more powerful tools to solve the prob-

lems that are challenging for the conventional approaches. For instance, neural network (NN), which is widely accepted as the most powerful ML model, is utilized to design functional materials with much higher efficiency than human experts [2–7]. One popular way is to apply to ML model to fit the relations between the experimental or numerical data and the target physical quantities. There are also some works that are directly aimed to solve physical equations, such as Schrödinger equation [8–12] or those in the *ab-initio* simulations [13,14], using ML. For the strongly correlated systems, NN has also been used as efficient state ansatz to solve the eigenstates of given Hamiltonians [15,16].

On the other hand, the hybridizations with ML bring powerful numerical tools to investigate the inverse problems. These problems are critical in many numerical and experimental setups, such as designing the exchange-correlation potentials in the *ab-initio* simulations of material [17,18], the analytic continuation of the imaginary Green's function into the real frequency domain [19], and designing quantum simulators [20]. One topic that currently attracts wide interests is to estimate the Hamiltonian given the states or their properties [21–26]. Considering the quantum lattice models, for example, it has been proposed to predict the coupling constants from the measurements of the target states [27–29] or the local reduced density matrices [30]. Sehanobish *et al.* consider the Schrödinger equation and propose the quantum potential NN (QPNN) to predict the potential term provided the eigen wave-function [31]. These works indicate the feasibility of using ML to investigate quantum phenomena by reformulating the quantum mechanical systems as the solutions of certain inverse problems.

In this work, we propose to combine Metropolis sampling with deep NN to gain higher accuracy and efficiency on the predictions of quantum potential, which we dub as Metropolis potential neural network (MPNN). The goal is to estimate the potential in the continuous space. The data to train the NN contain multiple coordinates with the labels as the expected values of the potential function. Metropolis sampling [32] allows to efficiently obtain the training data (see some applications of Metropolis sampling to ML and quantum computation in, e.g., [15,33–36], to name but a few) and evaluate the energies of the given wave-functions, same as the quantum Monte Carlo approaches [37–39]. A loss function that explicitly involves the energy is proposed to characterize the violation of the Schrödinger equation. The variational parameters in the NN are optimized by minimizing the loss function using back propagation [40]. Benchmarking on the harmonic oscillator and hydrogen atom, MPNN exhibits higher accuracy and stability on predicting the potential and evaluating the eigen-energy.

2 Brief review on quantum potential neural network

Consider the time-independent Schrödinger equation in D dimensions

$$\left[-\frac{\nabla^2}{2} + V(\mathbf{r})\right]\Psi(\mathbf{r}) = E\Psi(\mathbf{r}), \tag{1}$$

with the coordinates $\mathbf{r} = (x_1, \cdots, x_D)$ and $\frac{\hbar}{m} = 1$ as the energy scale. Normally, the task is to solve the eigenstates and energies given the potential $V(\mathbf{r})$. Here, we considered an inverse problem, which is to solve the potential so that the given wave-function $\Psi(\mathbf{r})$ is the eigenstate of the Hamiltonian.

In Ref. [31], the authors propose to use a deep neural network named as quantum potential neural network (QPNN) to predict the unknown potential $V(\mathbf{r})$. In detail, the QPNN maps the coordinates to the values of the potential, denoted as $U_{\boldsymbol{\theta}}(\mathbf{r})$ with $\boldsymbol{\theta}$ the variational parameters of the QPNN. With a trial potential, a spatial-dependent energy is introduced as

$$E(\mathbf{r}) = -\frac{\nabla^2\Psi(\mathbf{r})}{2\Psi(\mathbf{r})} + U_{\boldsymbol{\theta}}(\mathbf{r}). \tag{2}$$
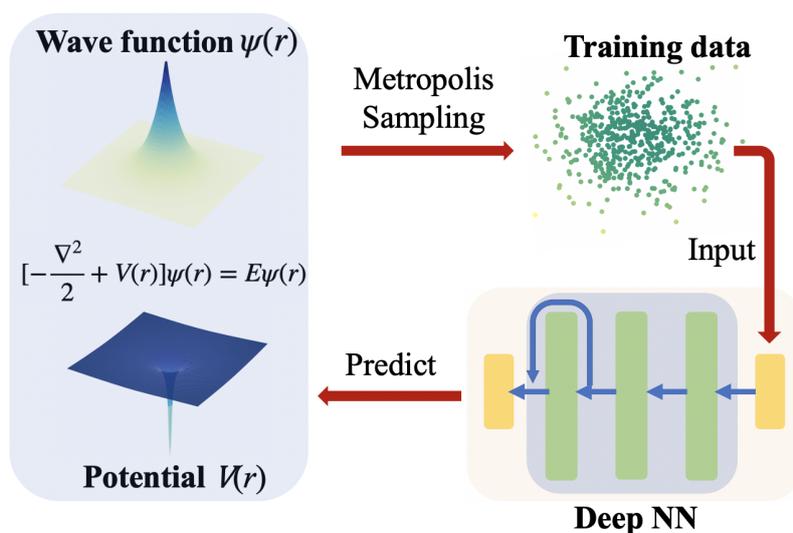
Figure 1: (Color online) The illustration of the main procedures of MPNN. With the potential $U_\theta(\mathbf{r})$ predicted by the neural network, the target wave-function $\Psi(\mathbf{r})$ is expected to be the eigenstate of the Hamiltonian $\hat{H} = -\frac{\nabla^2}{2} + V(\mathbf{r})$.

or

$$E'(\mathbf{r}) = -\frac{\nabla^2 |\Psi(\mathbf{r})|}{2|\Psi(\mathbf{r})|} + U_\theta(\mathbf{r}). \tag{3}$$

One can see that $E(\mathbf{r})$ and $E'(\mathbf{r})$ is the same if $\Psi(\mathbf{r})$ is real for any $\mathbf{r}$. The authors of Ref. [31] proposed to use $E'(\mathbf{r})$, considering that the square root of the probability density is much easier to access in experiments.

To characterize the extent of how the Schrödinger equation is satisfied, the loss function is defined as

$$L = \int |\nabla E(\mathbf{r})|^2 \, d\mathbf{r} + [U_\theta(\mathbf{r}_0) - V(\mathbf{r}_0)]^2, \tag{4}$$

with $\mathbf{r}_0$ a given coordinate at which the value of potential $V(\mathbf{r}_0)$ is previous known. In the practical simulations, one should choose a finite region and discretize the space into pieces with identical width. The loss function is then approximated as

$$L = \sum_{n=1}^{N} |\nabla E(\mathbf{r})|^2_{\mathbf{r}=\mathbf{r}_n} + [U_\theta(\mathbf{r}_0) - V(\mathbf{r}_0)]^2, \tag{5}$$

where $\{\mathbf{r}_n\}$ are sampled randomly from the discretized positions.

The loss minimally takes $L = 0$. In this case, one has $E(\mathbf{r}) = E$ as a constant independent on $\mathbf{r}$, and $U_\theta(\mathbf{r}_0) = V(\mathbf{r}_0)$. Then $\Psi(\mathbf{r})$ is strictly the eigenstate of the Hamiltonian $\hat{H}_\theta = -\frac{\nabla^2}{2} + U_\theta(\mathbf{r})$, with $U_\theta(\mathbf{r}_0)$ given by the QPNN and E the eigen-energy. With a nonzero loss, one normally has E as a function of the coordinates $\mathbf{r}$, and possibly a deviation between $U_\theta(\mathbf{r}_0)$ and its expected value $V(\mathbf{r}_0)$. In general, the loss function L in Eq. (5) characterizes how well the potential from the QPNN gives the target wave-function as an eigenstate, and should be minimized. The QPNN can be updated using the gradient decent method as

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \frac{\partial L}{\partial \boldsymbol{\theta}}, \tag{6}$$

with η the learning rate.

3 Metropolis Potential Neural Network Method

The MPNN method is illustrated in Fig. 1. Our goal is solving the potential $V(\mathbf{r})$ while knowing the target wave-function $\Psi(\mathbf{r})$ as the eigenstate of the Hamiltonian. The first step is sampling N positions $\{\mathbf{r}_n\}$ according to the probability distribution

$$P(\mathbf{r}) = |\Psi(\mathbf{r})|^2. \tag{7}$$

The sampling process can be implemented on a quantum platform if one can make sufficiently many copies of the state $\Psi(\mathbf{r})$, or on a classical computer when $\Psi(\mathbf{r})$ is analytically or numerically accessible. Then a neural network (NN) is applied to predict the values of potential at these positions $\{U_\theta(\mathbf{r}_n)\}$.

To estimate how the potential predicted by the NN satisfies the Schrödinger equation, we define the loss function as mean-square error of the deviations that reads

$$L = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left|[\hat{H}_\theta\Psi(\mathbf{r})]_{\mathbf{r}=\mathbf{r}_n} - \tilde{E}\Psi(\mathbf{r}_n)\right|^2}$$
$$+ \lambda\left[U_\theta(\mathbf{r}_0) - V(\mathbf{r}_0)\right]^2. \tag{8}$$

Since any global constant shift of the potential (i.e. $V(\mathbf{r}) \leftarrow V(\mathbf{r}) + const.$) would only cause a shift on the energy, a Lagrangian multiplier is added to fix the constant. In other words, we need to know the value of the ground-true potential $V(\mathbf{r}_0)$ at one certain coordinate $\mathbf{r}_0$. The λ is a tunable hyper-parameter to control the strength of this constraint. In $\hat{H}_\theta = -\frac{\nabla^2}{2} + U_\theta$, the kinetic energy can be estimated while knowing $\Psi(\mathbf{r})$, and U_θ is given by the NN.

In L, we explicitly evaluate the energy $\tilde{E}$ of the target state given $U_\theta(\mathbf{r}_n)$ as

$$\bar{E} = \langle\hat{H}\rangle = \int E(\mathbf{r})P(\mathbf{r})d\mathbf{r} \simeq \frac{1}{N}\sum_{n=1}^{N}E(\mathbf{r}_n), \tag{9}$$

with $E(\mathbf{r}_n)$ given by Eq. (2) and the positions $\{\mathbf{r}_n\}$ sampled from the probability distribution $P(\mathbf{r})$ in Eq. (7). With the loss $L \to 0$, the NN would give a potential $U_\theta(\mathbf{r}_n) \to V(\mathbf{r})$ satisfying the Schrödinger equation (note $V(\mathbf{r})$ denotes the "correct" potential that we expect the NN to give). Meanwhile, the constraint is satisfied, i.e., $|U_\theta(\mathbf{r}_0) - V(\mathbf{r}_0)| \to 0$, with $L \to 0$.

4 Numerical results

To benchmark the performance of MPNN, we take the ground states of the hydrogen atom and 1D harmonic oscillator (HO) as examples. Note for the hydrogen atom, we do not use the spherical coordinate to transform the Schrödinger equation in three spatial dimensions to a 1D radial equation, just to test the performance on predicting the 3D potentials.

To compare with QPNN, here we use the same architecture of the NN. There are three hidden layers in the NN, where the number of the hidden variables in each layer is no more than 128. A residual channel is added between second and third layers. We use Adam as the optimizer to control the learning rate η [see Eq. (6)]. The testing set are sampled independently from the training set. In other words, the coordinates in the testing set are different from those in the training set.

To show the accuracy, we demonstrate in Table 1 (a) the error of potential as

$$\varepsilon = \frac{1}{N}\sum_{n=1}^{N}|U_\theta(\mathbf{r}_n) - V(\mathbf{r}_n)|. \tag{10}$$

Table 1: (a) The error of potential ε in Eq. (10) and (b) the energy $\bar{E}$ obtained by QPNN, QPNN with Metropolis sampling (MS) as another baseline for comparison, and MPNN. We take the ground state of the hydrogen atom and the second excited state of 1D harmonic oscillator (HO) as examples. The numbers of hidden variables in the three layers of the NN are taken as 32, 128, and 128, respectively, same as those in Ref. [31]. See more details about the evaluations of the error and energy for these three methods in the main text.

(a) Error of potential

ε	QPNN	QPNN+MS	MPNN
hydrogen (ground state)	0.072	0.060	0.028
1D HO (2nd excitation)	0.042	0.016	0.006

(b) Energy

$\bar{E}$	Exact	QPNN	QPNN+MS	MPNN
hydrogen (ground state)	-0.5	-0.486	-0.534	-0.493
1D HO (2nd excitation)	2.5	2.474	2.519	2.506

We evaluate ε by averagely taking $20 \times 20 \times 20 = 8000$ coordinates $\{\mathbf{r}_n\}$ in $x, y, z \in [-1, 1]$. Besides QPNN and MPNN, we also test a modified version of QPNN by simply replacing the purely random sampling by Metropolis sampling, which we denote as QPNN+MS. In specific, the coordinates $\{\mathbf{r}_n\}$ to evaluate the loss function in Eq. (5) are randomly obtained according to the probability in Eq. (7). Other parts including the NN are the same as the QPNN. For MPNN, we use the loss function given in Eq. (8) where $\{\mathbf{r}_n\}$ are also obtained by Metropolis sampling. Our results indicate that one should explicitly involve the energy in the loss function as Eq. (8) to give full play to the advantages of Metropolis sampling. The lowest losses is stably obtained by MPNN for these two systems.

Table 1 (b) shows the energies by QPNN, QPNN with Metropolis sampling, and MPNN. For QPNN, if $\Psi(\mathbf{r})$ is an eigenstate of $\hat{H}_{\boldsymbol{\theta}}$, one will have a zero loss and $E(\mathbf{r})$ in Eq. (2) as a constant. Therefore, it is a reasonable evaluation of the energy for QPNN using the average of $E(\mathbf{r})$ as

$$\bar{E}_{\text{QPNN}} = \frac{1}{N} \sum_{n=1}^{N} E(\mathbf{r}_n). \tag{11}$$

A more reasonable choice to evaluate the average of the Hamiltonian $\langle \hat{H}_{\boldsymbol{\theta}} \rangle$ for $\Psi(\mathbf{r})$, for which the correct way is to calculate a weighted average as Eq. (9). For this reason, we use Metropolis sampling to get the positions $\{\mathbf{r}_n\}$ in QPNN+MS. Another potential advantage of using Metropolis sampling is that the positions with $|\Psi(\mathbf{r})| \to 0$ will be avoided since the probability of having these positions will be vanishing. For the nonzero $\{\mathbf{r}_n\}$, we have $E(\mathbf{r}_n) = E'(\mathbf{r}_n)$.

From our results, we do not see obvious improvement on evaluating the energy by introducing Metropolis sampling to the QPNN. The error compared with the exact solution is around $O(10^{-1})$. One possible reason is that the energy is not explicitly involved in the loss function, i.e., in the optimization. The MPNN method gives the most accurate among these three approaches, with the error around $O(10^{-3})$.

MPNN also shows its advantage on the sampling efficiency. Fig. 2 demonstrates the average of the error ε [Eq. (10)] with different numbers of samples N used to optimize NN. We implement 10 independent simulations to calculate the average and variance of ε for each N.
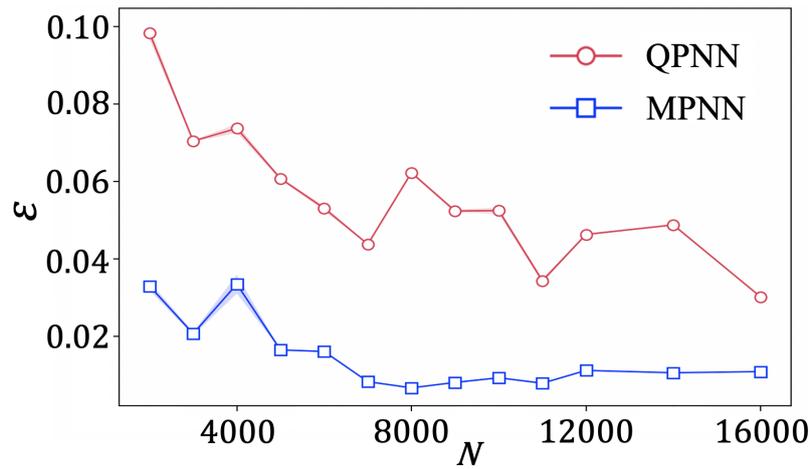
Figure 2: (Color online) The average of the error ε [Eq. (10)] by MPNN and QPNN with different numbers of samples N used to optimize NN. For each point in the curve, we implement 10 independent simulations to obtain the average and the variance that is $O(10^{-3})$ or less (illustrated by the shadows). The numbers of hidden variables in the three layers of the NN are taken as 128, 128, and 128, respectively.
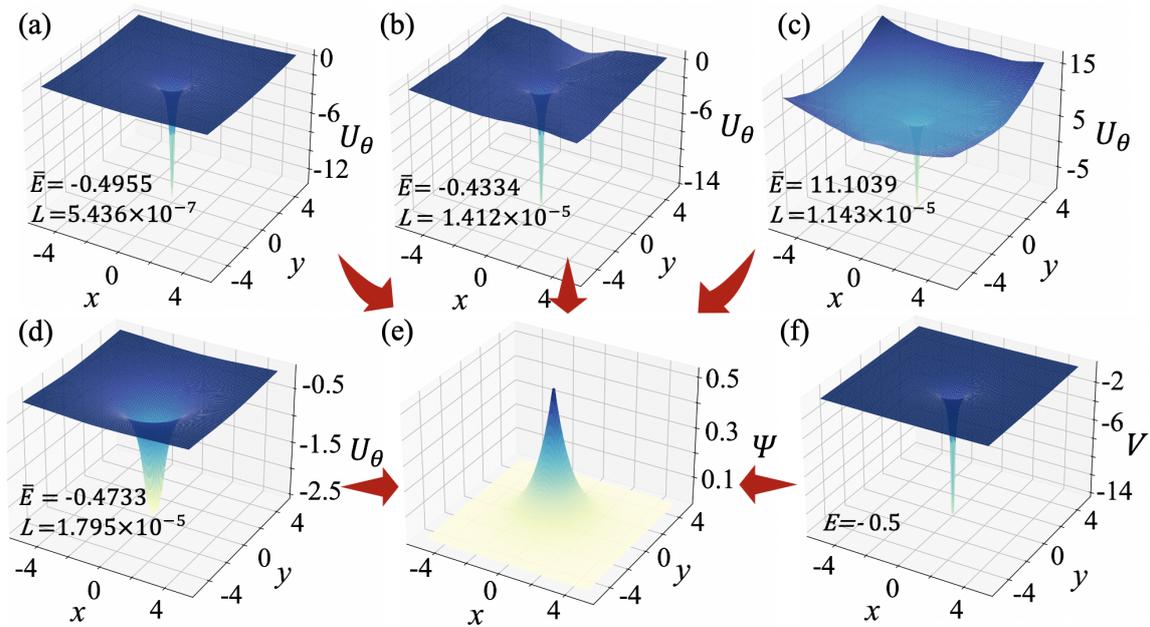
Figure 3: (Color online) (a)-(d) show four different potentials $U_\theta(\mathbf{r})$ that approximately give the ground-state wave-function $\Psi(\mathbf{r})$ of the hydrogen atom with the losses $L \simeq O(10^{-5}) - O(10^{-6})$. The potential in (a) is obtained by our MPNN, which gives the lowest loss and the best estimation near the singular point $\mathbf{r} = 0$. The numbers of hidden variables are taken as the same values as those in Fig. 2. See more details of (a)-(d) in the main text. We show in (e) the target wave-function $\Psi(\mathbf{r})$, and in (f) the exact potential $V(\mathbf{r}) = -\frac{1}{|\mathbf{r}|}$. In all sub-figures, we fix $z = 0$ to illustrate the x- and y-dependence of the potentials or wave-function.

Note the fluctuations of ε are from the randomness in the initialization of the variational parameters in the NN and the sampling processes. The variances are illustrated by the shadows, which are around $O(10^{-3})$ or less. With a same N, MPNN achieves a lower error than QPNN.
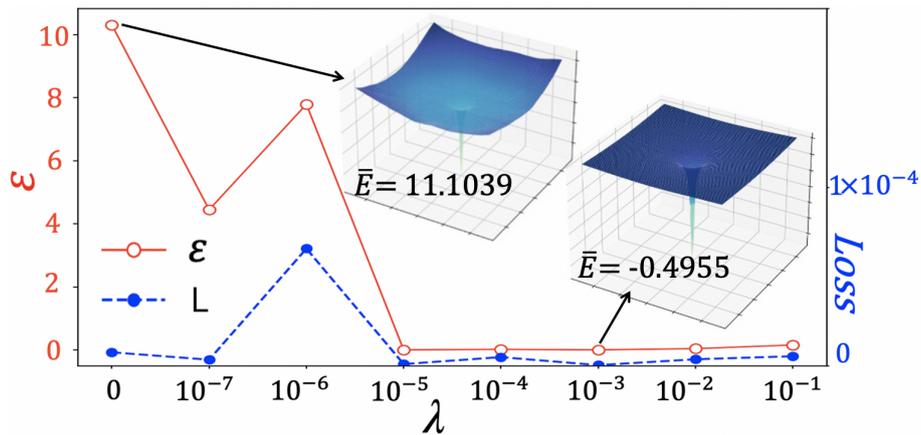
Figure 4: (Color online) The error of the potential [Eq. (10)] and the loss [Eq. (8)] with different values of the Lagrangian multiplier λ. The insets show two examples of the predicted potential with $z = 0$. We take the numbers of hidden variables to be the same values as those in Fig. 2.

There exist many local minimums of the loss function. A bad local minimum might give rise to an incorrect or inaccurate energy, even if the value of the loss is small. Figs. 3 (a)-(d) show four different U_{θ} that give the losses $L \sim O(10^{-5}) - O(10^{-6})$ for the ground state of the hydrogen atom. Figs. 3 (e) and (f) show the exact ground-state wave-function $\Psi(\mathbf{r})$ and the potential $V(\mathbf{r}) = -\frac{1}{|\mathbf{r}|}$. We fix $z = 0$ to illustrate the $x - y$ dependence of the potentials or wave-function.

Compared with the expected potential $V(\mathbf{r})$, the best result is obtained by the MPNN, illustrated in Fig. 3 (a). By changing the initialization strategy of the NN, say without multiplying the initial θ with δ, one may obtain a different energy with a similar loss, as shown in Fig. 3 (b). Our simulation results indicate an effective initialization strategy by letting the initial potential be near the hyper-surface of $V = 0$. This can be done by first randomly initializing all θ in the NN and then multiplying them with a small factor, e.g., $\delta = 0.01$.

In Fig. 3 (c), we set $\lambda = 0$, then an extra degree of freedom will appear. It can be easily seen that a potential $V'(\mathbf{r}) = V(\mathbf{r}) + C$ will be the solution of our inverse problem for any constant C if $V(\mathbf{r})$ is the solution. The penalty term $\lambda [U_{\theta}(\mathbf{r}_0) - V(\mathbf{r}_0)]^2$ is to fix this degrees of freedom to give the correct energy.

Fig. 3 (d) shows the U_{θ} obtained by the QPNN. The dominant error is the data that are taken near the center of the potential. For the MPNN, the positions with larger $|\Psi(\mathbf{r})|$ are taken more frequently in the Metropolis sampling. Better prediction is obtained since such data contribute more to the physical properties, such as the observables and the gradients in optimizing the NN, compared with those that have small $|\Psi(\mathbf{r})|$.

The penalty term $\lambda [U_{\theta}(\mathbf{r}_0) - V(\mathbf{r}_0)]^2$ is to fix the degrees of freedom with a global shift of the potential. The coefficient determines how strictly we require the NN to give the correct value at the position $\mathbf{r}_0$. Fig. 4 shows the error ε in Eq. (10) of the hydrogen atom with different values of λ. For $\lambda = 0$, meaning we do not require $U_{\theta}(\mathbf{r}_0) = V(\mathbf{r}_0)$, the predicted potential can be shifted determined by the initial values of θ. Thus one cannot correctly give the energy $\bar{E}$, and the error ε is significant. But the loss is small with $L \sim O(10^{-4})$. This means without knowing the potential at some position, we cannot uniquely give the eigen-energy of Ψ but can still give the potential U_{θ} so that Ψ is the eigenstate of $\hat{H}_{\theta}$.

As λ increases to certain extent, we are able to obtain the expected potential $V(\mathbf{r})$ with small ε. Note the loss is still small, which fluctuates around $L \sim O(10^{-4}) - O(10^{-5})$. In such cases, we obtain accurate predictions of the eigen-energy $\bar{E} \simeq -0.5$.

5 Summary

The hybridization of machine learning with quantum physics brings new possibility to solve the important problems that are challenging using the conventional approaches. Stimulated by the quantum potential neural network, we consider to predict the potential in Schrödinger equation, with which the target state Ψ is the eigenstate of the Hamiltonian. The Metropolis potential neural network (MPNN) is proposed to predict the potential by combining deep neural network and Metropolis sampling. With the benchmark on the harmonic oscillator and hydrogen atom, MPNN exhibits excellent precision and stability on both predicting the potential and evaluating the eigen-energy. Our proposal can be readily generalized to inversely solving the Schrödinger equation of multiple electrons, and the differentiation equations for other physical problems.

Acknowledgment

This work was supported by NSFC (Grant No. 12004266, No. 11774300 No. 11834014 and No. 11875195), Beijing Natural Science Foundation (No. 1192005 and No. Z180013), Foundation of Beijing Education Committees (No. KM202010028013), and the key research project of Academy for Multidisciplinary Studies, Capital Normal University.

References

[1] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto and L. Zdeborová, *Machine learning and the physical sciences*, Rev. Mod. Phys. **91**, 045002 (2019), doi:10.1103/RevModPhys.91.045002.

[2] R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakkithodi and C. Kim, *Machine learning in materials informatics: recent applications and prospects*, npj Comput. Mater. **3**, 54 (2017), doi:10.1038/s41524-017-0056-5.

[3] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev and A. Walsh, *Machine learning for molecular and materials science*, Nature **559**, 547 (2018), doi:10.1038/s41586-018-0337-2.

[4] T. Xie and J. C. Grossman, *Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties*, Phys. Rev. Lett. **120**, 145301 (2018), doi:10.1103/PhysRevLett.120.145301.

[5] J. E. Gubernatis and T. Lookman, *Machine learning in materials design and discovery: Examples from the present and suggestions for the future*, Phys. Rev. Materials **2**, 120301 (2018), doi:10.1103/PhysRevMaterials.2.120301.

[6] H. A. Doan, G. Agarwal, H. Qian, M. J. Counihan, J. Rodríguez-López, J. S. Moore and R. S. Assary, *Quantum chemistry-informed active learning to accelerate the design and discovery of sustainable energy storage materials*, Chem. Mater. **32**, 6338 (2020), doi:10.1021/acs.chemmater.0c00768.

[7] X.-Y. Ma, H.-Y. Lyu, X.-J. Dong, Z. Zhang, K.-R. Hao, Q.-B. Yan and G. Su, *Voting data-driven regression learning for accelerating discovery of advanced functional materials and applications to two-dimensional ferroelectric materials*, J. Phys. Chem. Lett. **12**, 973 (2021), doi:10.1021/acs.jpclett.0c03136.

[8] J. Han, L. Zhang and W. E., *Solving many-electron Schrödinger equation using deep neural networks*, J. Comp. Phys. **399**, 108929 (2019), doi:10.1016/j.jcp.2019.108929.

[9] K. T. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller and R. J. Maurer, *Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions*, Nat. Commun. **10**, 5024 (2019), doi:10.1038/s41467-019-12875-2.

[10] D. Pfau, J. S. Spencer, A. G. D. G. Matthews and W. M. C. Foulkes, *Ab initio solution of the many-electron Schrödinger equation with deep neural networks*, Phys. Rev. Research **2**, 033429 (2020), doi:10.1103/PhysRevResearch.2.033429.

[11] S. Manzhos, *Machine learning for the solution of the Schrödinger equation*, Mach. Learn.: Sci. Technol. **1**, 013002 (2020), doi:10.1088/2632-2153/ab7d30.

[12] J. Hermann, Z. Schätzle and F. Noé, *Deep-neural-network solution of the electronic Schrödinger equation*, Nat. Chem. **12**, 891 (2020), doi:10.1038/s41557-020-0544-y.

[13] K. Ryczko, D. A. Strubbe and I. Tamblyn, *Deep learning and density-functional theory*, Phys. Rev. A **100**, 022512 (2019), doi:10.1103/PhysRevA.100.022512.

[14] M. Michael Denner, M. H. Fischer and T. Neupert, *Efficient learning of a one-dimensional density functional theory*, Phys. Rev. Research **2**, 033388 (2020), doi:10.1103/PhysRevResearch.2.033388.

[15] G. Carleo and M. Troyer, *Solving the quantum many-body problem with artificial neural networks*, Science **355**, 602 (2017), doi:10.1126/science.aag2302.

[16] I. Glasser, N. Pancotti, M. August, I. D. Rodriguez and J. Ignacio Cirac, *Neural-network quantum states, string-bond states, and chiral topological states*, Phys. Rev. X **8**, 011006 (2018), doi:10.1103/PhysRevX.8.011006.

[17] D. S. Jensen and A. Wasserman, *Numerical methods for the inverse problem of density functional theory*, Int. J. Quantum Chem. **118**, e25425 (2017), doi:10.1002/qua.25425.

[18] Y. Zhang et al., *Machine learning in electronic-quantum-matter imaging experiments*, Nature **570**, 484 (2019), doi:10.1038/s41586-019-1319-8.

[19] R. Fournier, L. Wang, O. V. Yazyev and Q. Wu, *Artificial neural network approach to the analytic continuation problem*, Phys. Rev. Lett. **124**, 056401 (2020), doi:10.1103/PhysRevLett.124.056401.

[20] Y. Hong Teoh, M. Drygala, R. G. Melko and R. Islam, *Machine learning design of a trapped-ion quantum spin simulator*, Quantum Sci. Technol. **5**, 024001 (2020), doi:10.1088/2058-9565/ab657a.

[21] J. Behler and M. Parrinello, *Generalized neural-network representation of high-dimensional potential-energy surfaces*, Phys. Rev. Lett. **98**, 146401 (2007), doi:10.1103/PhysRevLett.98.146401.

[22] K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K. R. Müller and E. K. U. Gross, *How to represent crystal structures for machine learning: Towards fast prediction of electronic properties*, Phys. Rev. B **89**, 205118 (2014), doi:10.1103/PhysRevB.89.205118.

[23] B. Jiang, J. Li and H. Guo, *Potential energy surfaces from high fidelity fitting ofab initio-points: the permutation invariant polynomial - neural network approach*, Int. Rev. Phys. Chem. **35**, 479 (2016), doi:10.1080/0144235X.2016.1200347.

[24] G. Hegde and R. Chris Bowen, *Machine-learned approximations to density functional theory hamiltonians*, Sci. Rep. **7**, 42669 (2017), doi:10.1038/srep42669.

[25] N. F. Berthusen, Y. Sizyuk, M. Scheurer and P. Orth, *Learning crystal field parameters using convolutional neural networks*, SciPost Phys. **11**, 011 (2021), doi:10.21468/SciPostPhys.11.1.011.

[26] C. Kokail, B. Sundar, T. V. Zache, A. Elben, B. Vermersch, M. Dalmonte, R. van Bijnen and P. Zoller, *Quantum variational learning of the entanglement hamiltonian* (2021), arXiv:2105.04317.

[27] T. Xin, S. Lu, N. Cao, G. Anikeeva, D. Lu, J. Li, G. Long and B. Zeng, *Local-measurement-based quantum state tomography via neural networks*, npj Quantum Inf. **5**, 109 (2019), doi:10.1038/s41534-019-0222-3.

[28] S.-T. Wang, D.-L. Deng and L.-M. Duan, *Hamiltonian tomography for quantum many-body systems with arbitrary couplings*, New J. Phys. **17**, 093017 (2015), doi:10.1088/1367-2630/17/9/093017.

[29] E. Bairey, I. Arad and N. H. Lindner, *Learning a local hamiltonian from local measurements*, Phys. Rev. Lett. **122**, 020504 (2019), doi:10.1103/PhysRevLett.122.020504.

[30] X. Ma, Z. C. Tu and S.-J. Ran, *Deep learning local reduced density matrices for many-body hamiltonian estimation* (2021), arXiv:2012.03019v1.

[31] A. Sehanobish, H. H. Corzo, O. Kara and D. van Dijk, *Learning potentials of quantum systems using deep neural networks* (2020), arXiv:2006.13297.

[32] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *Equation of state calculations by fast computing machines*, J. Chem. Phys. **21**, 1087 (1953) ,doi:10.1063/1.1699114.

[33] E. M. Inack, G. E. Santoro, L. Dell'Anna and S. Pilati, *Projective quantum Monte Carlo simulations guided by unrestricted neural network states*, Phys. Rev. B **98**, 235145 (2018), doi:10.1103/PhysRevB.98.235145.

[34] A. Nagy and V. Savona, *Variational quantum Monte Carlo method with a neural-network ansatz for open quantum systems*, Phys. Rev. Lett. **122**, 250501 (2019), doi:10.1103/PhysRevLett.122.250501.

[35] K. Choo, T. Neupert and G. Carleo, *Two-dimensional frustrated $J_1 - J_2$ model studied with neural network quantum states*, Phys. Rev. B **100**, 125124 (2019), doi:10.1103/PhysRevB.100.125124.

[36] P. A. M. Casares, R. Campos and M. A. Martin-Delgado, *QFold: Quantum walks and deep learning to solve protein folding* (2021), arXiv:2101.10279.

[37] B. A. Berg, *Markov chain Monte Carlo simulations and their statistical analysis: with web-based Fortran code*, World Scientific Publishing Company (2004), doi:10.1142/5602.

[38] D. Landau and K. Binder, *A guide to Monte Carlo simulations in statistical physics*, Cambridge University Press (2014), doi:10.1017/CBO9781139696463.

[39] W. M. C. Foulkes, L. Mitas, R. J. Needs and G. Rajagopal, *Quantum Monte Carlo simulations of solids*, Rev. Mod. Phys. **73**, 33 (2001), doi:10.1103/RevModPhys.73.33.

[40] I. Goodfellow, Y. Bengio and A. Courville, *Deep learning*, MIT Press (2016).