Communications in Computational Chemistry
Commun. Comput. Chem., Vol. 1, No. 2, 120–126
doi: 10.4208/cicc.2025.92.01
Regular Article

Evaluation of Phase Networks in Transformer-Based Neural Network Quantum States

Lizhong Fu[1], Honghui Shang[1,*] and Jinlong Yang[1,2,*]

[1]*State Key Laboratory of Precision and Intelligent Chemistry, University of Science and Technology of China, Hefei, Anhui 230026, China;*

[2]*Hefei National Laboratory, University of Science and Technology of China, Hefei 230088, China.*

[*] Corresponding authors: shanghui.ustc@gmail.com; jlyang@ustc.edu.cn

Received on 02 April 2025; Accepted on 25 April 2025

Abstract: Neural network quantum states represent a powerful approach for solving electronic structures in strongly correlated molecular and material systems. For a neural network ansatz to be accurate, it must effectively learn the phase of a complex wave function. In this work, we demonstrate several different network structures as the phase network for a Transformer-based neural network quantum state implementation. We compare the accuracy of ground state energies, the number of parameters, and computational time across several small molecules. Furthermore, we propose a phase network setup that combines cross attention and multilayer perceptron structures, with the number of parameters remaining constant across different systems. Such an architecture may help reduce computational costs and enable transfer learning to larger quantum chemical systems.

Key words: neural network quantum states, phase network, electronic structure calculation.

1. Introduction

Solving the electronic structure of correlated molecular and material systems has long been an essential task in computational chemistry. In these systems, mean-field theories such as density functional theory (DFT) and the Hartree-Fock (HF) self-consistent field method fail to accurately describe the correlated electronic wave function, while the theoretically accurate full configuration interaction (FCI) method requires computational resources that scale exponentially with system size, making its application to large systems impractical.

The past decade has faced an explosive growth of applications of neural networks in different fields, where it has demonstrated remarkable ability in representing complicated functions encountered in various situations. In the context of computational many-body problems, this expressive power has been leveraged to represent the highly correlated electronic wave function. This approach, introduced in 2017 by Carleo and Troyer, is known as neural network quantum states (NNQS) [1]. NNQS methods have been applied to both spin and Fermionic models, such as the J1-J2 Heisenberg model and the Hubbard model [2–3]. In the context of computational chemistry, there have also been demonstrations in small molecular and material systems. Some of these works feature a real-space representation of the electronic wave function, in which a neural network takes the coordinates of electrons $(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \dots)$ as input, and outputs the wave function $\Psi(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \dots)$. Notable implementations of this method include FermiNet [4], PauliNet [5], LapNet [6], and DeepErwin [7]. Other works adopt a second-quantization formulation of the electronic structure problem, expressing both the wave function and the Hamiltonian in the occupation number representation over a given set of single-particle orbitals [8–9]. These methods introduce a basis set that enables direct comparison with standard quantum chemistry approaches and allows for systematic accuracy improvements by selecting increasingly comprehensive basis sets. However, the basis set approximation introduces an error that is absent in the real-space representation of the wave function.

One essential difference between NNQS and other applications of neural networks is that the electronic wave function is complex-valued, requiring a complex-valued network, whereas most deep learning applications focus on real-valued network outputs. This results in a lack of complex-valued network designs and limits the available choices for NNQS implementations. In many NNQS methods, this issue is circumvented by parameterizing the complex wave function using two separate real-valued networks: one for the amplitude and one for the phase. It has been shown in spin systems that learning the phase of the wave function is a challenging task for the network, and providing a reasonable initial guess based on the sign rule dramatically improves convergence behavior [2]. However, in quantum chemistry systems, no such simple rule exists for the phase, leaving it entirely up to the phase network to find the ground state. Therefore, choosing an appropriate structure for the phase network can improve both the energy landscape and the expressive ability of the NNQS wave function, leading to a faster convergence to the true wave function.

In this work, we compare several different implementations of phase network in QiankunNet, a transformer based NNQS method [10]. We compare ground state energy results on 17 typical small molecules. We also compare the number of parameters and time costs for each of these structures, in searching for a phase network structure that is both numerically accurate and computationally efficient.

2. Theoretical method

2.1 Transformer-based neural network quantum states

In computational chemistry, the wave function of a correlated system $|\Psi\rangle$ can be expressed as a state vector consisting of 2^n complex coefficients, where n is the number of spin orbitals used to define the system. NNQS method employs a neural network to represent such a state vector. The network takes the electron configuration on spin orbitals as input and returns the corresponding coefficients for each configuration. These coefficients are then used to calculate expectation values of operators, including the Hamiltonian. During a training process, one first sample a batch of configurations, then calculate the energy expectation values. The energy estimator serves as a loss function, whose gradient is calculated and used to update parameters in the network. Specifically, in our implementation, the wave function ansatz is of the following form:

$$\Psi_\theta = A_\theta e^{i\phi_\theta}$$

Where A_θ is the network representing amplitude of the wave function. We use GPT-style decoder-like layers for the amplitude network. Such a structure not only shows ability to capture long-range dependencies, but also enables the adoption of autoregressive sampling. Compared to classical Malkov Chain Monte Carlo (MCMC) sampling, the BAS algorithm has been shown to accelerate the sampling process significantly, thereby facilitating application of NNQS methods to larger systems [11-12]. The ϕ_θ part of the wave function ansatz is a network specifically designed to represent the phase of the wave function, which is the primary focus in this work.

It is worth noting that for systems with open boundary conditions, such as molecules, the electronic wave function can be treated as a real-valued function. In this case, the phase term $e^{i\phi_\theta}$ reduces to a simple ±1 sign before each amplitude coefficient. This raises the question of whether it is possible to directly represent the sign instead of the full phase, or even absorb the sign into the amplitude using a single real-valued network. However, directly optimizing the sign is challenging due to its inherently combinatorial nature. Moreover, absorbing the sign into the amplitude network leads to a rugged energy landscape, as the network must pass through zero to switch signs—potentially creating energy barriers that hinder optimization. Maintaining a separate phase network also enables extension to systems with periodic boundary conditions, where the wave function must remain genuinely complex-valued.

2.2 Phase network structures

In previous work, we used a simple multilayer perceptron (MLP) as the phase network. Such a structure consists only of fully-connected layers (FC) and can be expressed as

$$\log \phi = A_n(\cdots (A_2(\mathrm{A}_1 x + \mathrm{b}_1) + b_2) \cdots) + b_n$$

in which A_i, b_i are learnable parameters, and x is the input configuration. 4 hidden layers and a hidden layer dimension of 512 is used, thus the total number of trainable parameters is approximately $789\mathrm{k} + 512 \times n$, where n is the number of spin orbitals. This network structure is depicted in **Figure 1(a)**.

Since the decoder-like amplitude network has only about 50,000 parameters, it raises the question of whether such a large number of parameters in the phase network is truly necessary. To address this, we test three alternative approaches that replace the MLP with networks containing fewer parameters. The first approach is directly inspired by the amplitude network. We use several encoder layers to represent the phase, as illustrated in **Figure 1(b)**. In an encoder layer, the learnable parameters are located in the position-wise fully connected layers of dimension $n_{\mathrm{embedding}} \times n_{\mathrm{embedding}}$, as well as the position-wise feed forward network (FFN) [13]. By choosing embedding dimensions and FFN hidden dimensions smaller than the MLP hidden dimension (512), this structure can have significantly fewer parameters than an MLP. In our amplitude network implementation, the attention embedding dimensions are taken to be 32 or 48, while the FFN hidden dimension is 128. This suggests that similar hyperparameters may be also applicable in the case of phase dimensions.

Another approach is to use decoder layers as the phase ansatz. A typical decoder layer in the Transformer architecture consists of a self-attention layer, a cross attention layer, and a position-wise fully connected network. Since our task does not involve "transforming" one sequence into another, there are no encoder outputs available to serve as queries for the cross attention layer. Therefore, we use an array of zeros, with the same length as the input sequence, as the query for the cross attention layer. This structure is implemented directly using torch.nn.TransformerDecoderLayer, and a depiction can be found in **Figure 1(c)**. It is important to note that this phase network—referred to as "decoder" layers below and in the figures—is not the same as the decoder-like amplitude network. The amplitude network consists of GPT-style decoder-like Transformer layers, which, although commonly referred to as "decoder layers" in many contexts, are essentially encoder layers with masked self-attention modules.

The third method is inspired by applications in multimodal learning. In a cross attention mechanism, the length of the output sequence matches that of the query sequence, which does not necessarily have to be the same as the key and value sequences. In the

original MLP phase network, the number of parameters in the first fully connected layer increases with the number of spin orbitals. To address this, we use a fixed-length sequence of learnable parameters as the query and perform cross attention with the configuration sequence. This approach limits the length of the output sequence to a fixed value, except for the embedding layer, which contains only a few thousand parameters. Using this fixed-length sequence as input to the MLP eliminates the dependency of the number of parameters on system size. A depiction of this structure is shown in **Figure 1(d)**.

2.3 Computational details

In all calculations, we use the STO-3G basis set. All structures are downloaded from Pubchem database [14]. The 1- and 2-electron integrals are calculated using PySCF package [15], and openfermion is used to parse the integrals to a qubit operator [16], which is read by QiankunNet. The network is set up using PyTorch [17], and the training is performed using AdamW optimizer at learning rate 0.0001-0.005, which is tuned according to convergence behavior for each network. The number of unique samples is regulated in a recursive manner to be less than 50000 and more than 6000. All networks are randomly initialized and trained for 30000 epochs. The resulting energy is taken to be the lowest 100-step averaged energy achieved during the training process. The calculation is carried out using 2 Intel Xeon Scale 8358 CPU and one NVIDIA A100 GPU, with 200GB memory specified. The restricted Hartree-Fock (RHF), restricted coupled cluster with single and double excitations (RCCSD) and FCI calculations used for benchmark is carried out using PySCF.

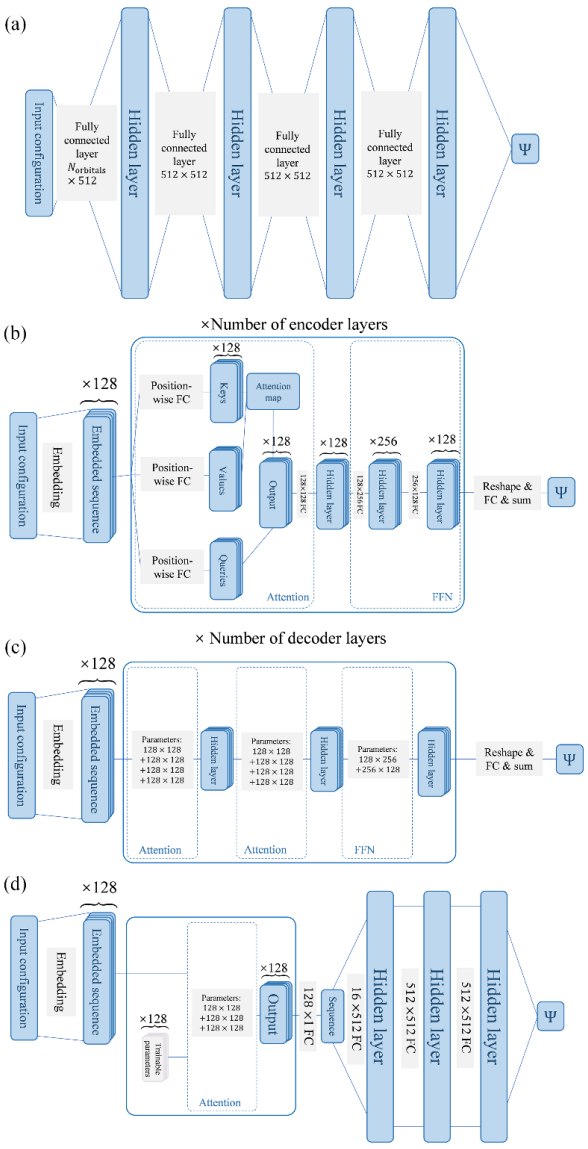

Figure 1. Different phase network structures used in this work. (a) Simple MLP consisting of 4 hidden layers of dimension 512. (b) Pure encoder layers with embedding size 128 and FFN hidden dimension 256. (c) Pure decoder layers, with two self-attention of embedding size 128 and one FFN with hidden dimension 256. (d) Cross attention + MLP structure, with a learnable sequence of length 16 as query for the cross attention. Output of the attention part is passed through an FFN with 3 hidden layers of dimension 512.

3. Results and discussion

3.1 Numerical accuracy of different methods

To compare accuracy of different network structures, we carried out ground state energy calculations on 17 small molecules. This test set spans a range of chemical complexities, from simple diatomic molecules to heavier systems of up to 30 spin orbitals, and includes molecules with diverse bonding types and electronic structures. Information for these test systems are found in **Table 1**. The results are compared with FCI values, and errors are plotted in **Figure 2**. Among all methods, the original MLP phase network shows the most consistent results, achieving chemical accuracy across all 17 test systems. The cross attention + MLP network also reaches chemical accuracy on all except one test molecules. Given that such a phase network structure has fixed parameter count regardless of system size, it is promising that this method may be applied to large system with partly

Table 1. Details of the 17 molecules tested in this work. The energy reference is computed using FCI.

Formula	Number of spin orbitals	Number of electrons	Energy reference [Hartree]
BeH_2	14	6	-14.47294742
C_2	20	12	-74.69078192
CH_2	14	8	-37.50443472
CH_4	18	10	-39.80625925
F_2	20	18	-195.6610863
H_2	4	2	-1.10115033
H_2O	14	10	-75.01553019
H_2S	22	18	-394.3546235
HCl	20	18	-455.1561885
Li_2O	30	14	-87.89269325
LiCl	28	20	-460.8496182
LiF	20	12	-105.1661721
LiH	12	4	-7.78446028
N_2	20	14	-107.6602064
NH_3	16	10	-55.52114983
O_2	20	16	-147.7502346
PH_3	24	18	-338.6983999

transferable parameters. The other two methods, the encoder-only and decoder-only network, fails to converge within chemical accuracy for some of the system. This indicates that, though thought to possess stronger expressive ability, attention mechanism is not necessarily better than simple MLP in such a scenario.

In order to demonstrate the capability of different methods in describing systems of different levels of correlation, we also computed the potential energy curve for C_2 and N_2 molecule. The result is shown in **Figure 3**. It can be seen that both RHF and RCCSD produce growing error with extending bond length, while the NNQS methods, especially the ones with MLP and cross attention + MLP phase network, behave consistently across all bond length. This

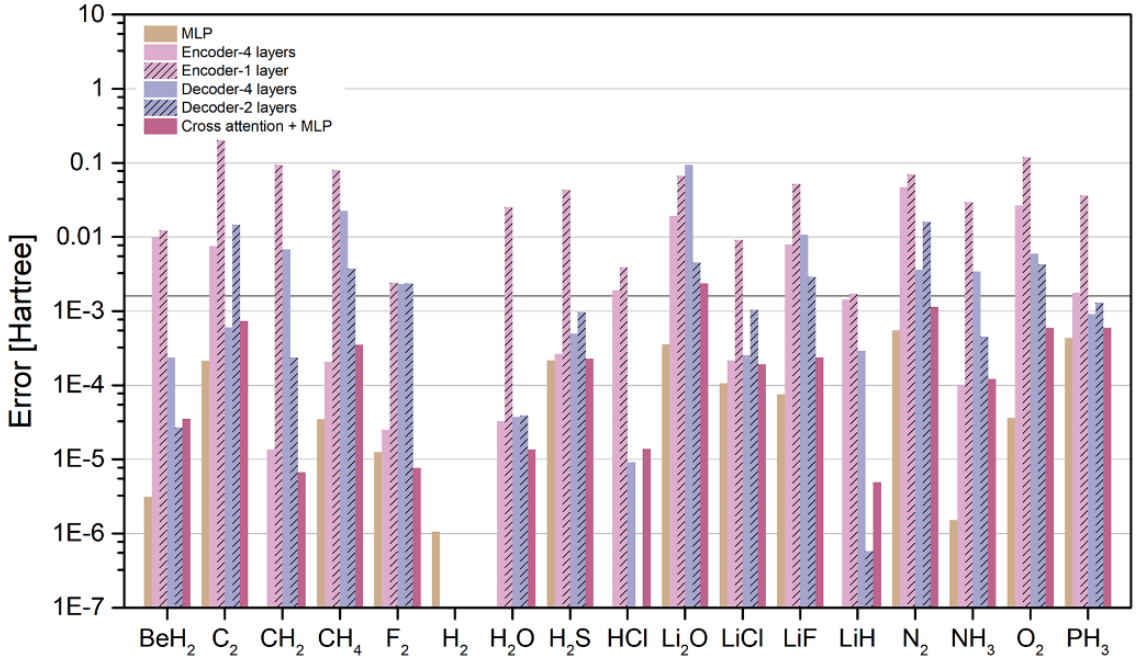

Figure 2. Error in ground state energy calculated using different phase network. Error below 10^{-8} Hartree is not shown. The horizontal line indicates chemical accuracy (1.6×10^{-3}Hartree).

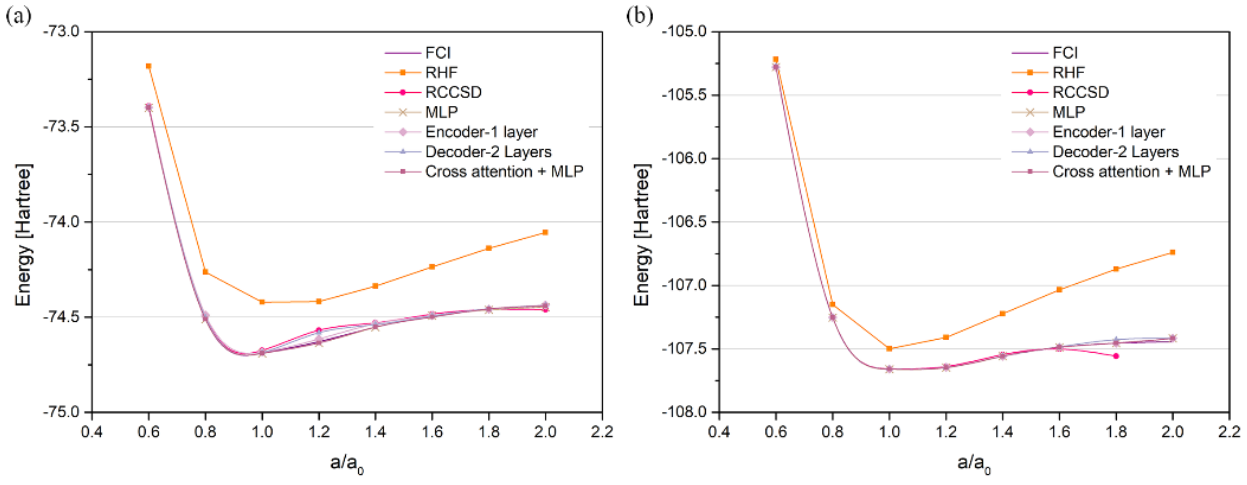

Figure 3. Potential energy curve for (a) C_2 and (b) N_2 calculated using various methods, including different NNQS architectures, RHF, RCCSD and FCI. The horizontal axis represents the bond length, given in units of the equilibrium bond length, which is obtained from the PubChem database.

demonstrates the ability of NNQS methods to be applied in both weakly and strongly correlated situations.

3.2 Number of parameters and computational efficiency

Another important aspect of evaluating an NNQS method is its computational efficiency. Currently, NNQS methods are limited to around 30 spin orbitals on standalone workstations, and may be extended to 50–60 orbitals in high-performance computing (HPC)

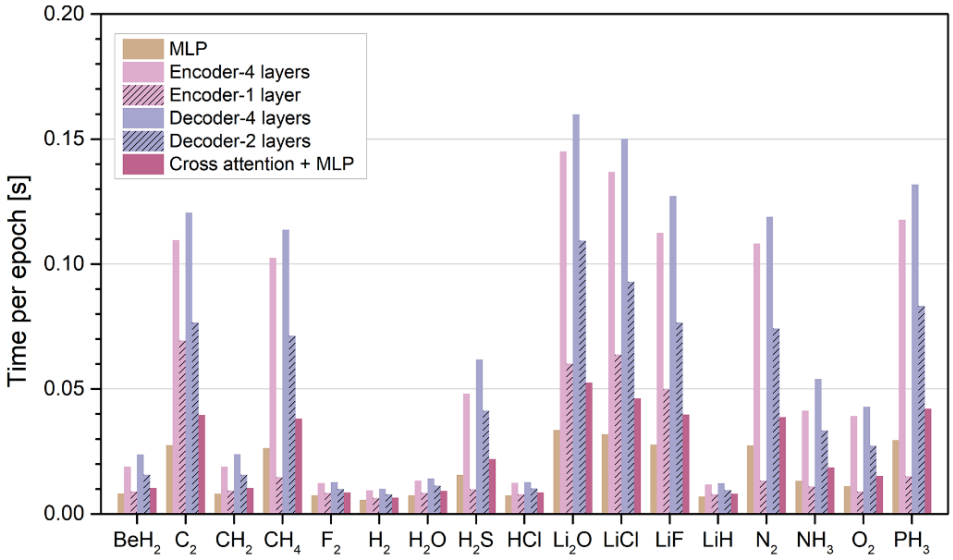

Figure 4. Time consumed for backward propagation and updating the network in each epoch. The data is averaged over the 100th to 200th epoch of each training, during which period the numbers of unique samples remain similar among all training processes.

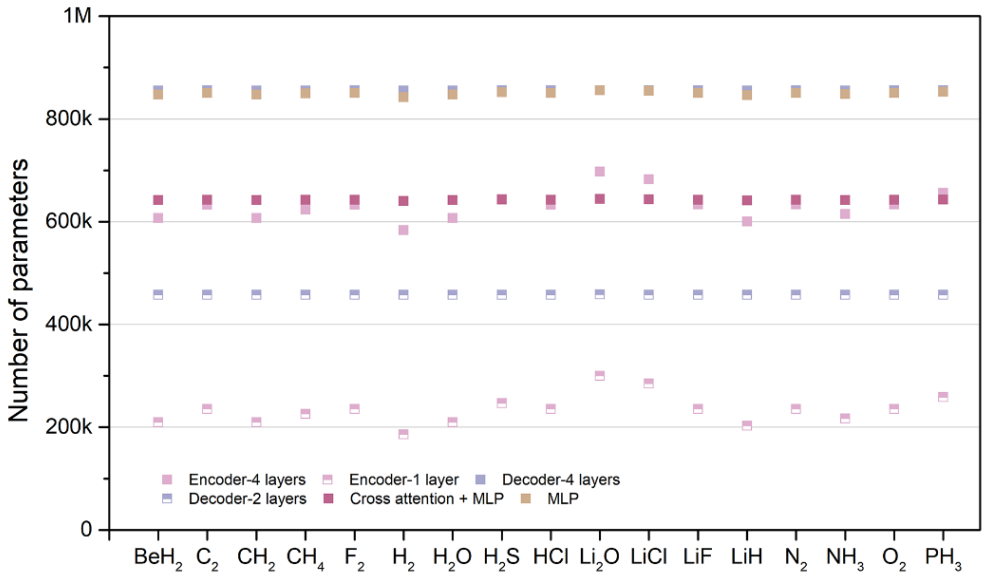

Figure 5. Total parameter counts for QiankunNet with different phase network, when used on different systems. The count is the sum of the amplitude and phase network, and it is seen that phase network structure has a large impact on total number of parameters.

environments. The bottlenecks mainly lie in poor scaling of Monte Carlo sampling, as well as the increasingly demanding gradient evaluation and backward propagation with growing number of trainable parameters. In a typical NNQS iteration, the time cost can be divided into three parts: the sampling part, in which samples of configurations are generated; the local energy part, in which the samples are used to calculate local energies and the energy expectation; and the gradient part, in which gradient of the energy with respect to parameters in the network is calculated and used to update the network. Since the amplitude network and the local energy calculation module remain unchanged, modifying the phase network does not affect the time cost of the first two parts. Therefore, we focus on the third part, and the corresponding time cost is plotted in **Figure 4** for comparison. It is observed that the original MLP implementation takes the least time, while other implementations incorporating attention mechanisms require more time. This can be understood as follows: in an MLP, each layer only requires simple matrix-vector multiplications. However, in an attention mechanism, for each word in a sequence, matrix-vector multiplications must be performed to compute keys, values, and queries. Additionally, a dot product must be computed for each pair of words to construct the attention map. This map is then used to generate the output sequence via matrix-vector multiplications, and the total complexity is of order N^2_{orbital}. The number of trainable parameters for each structure is depicted in **Figure 5**. The cross attention + MLP architecture has significantly fewer parameters than the original MLP phase network, while achieving similar results. Fewer parameters result in lower GPU memory load, faster checkpoint saving and reading, as well as

possibility in applying higher order optimizer such as stochastic reconfiguration (SR), whose computational cost scales with third order of number of parameters. Furthermore, the number of parameters in the cross attention + MLP phase network remains constant across different system sizes. This not only facilitates scaling the architecture to larger systems, but also enables transfer learning across systems with varying numbers of orbitals. Nevertheless, for the 17 small molecules studied in this work, it is also possible to tune the hyperparameters such that a pure MLP phase network contains a similar number of parameters—for example, by extracting the MLP part from the cross attention + MLP architecture. This smaller MLP network has been tested and found to achieve comparable results (see Supporting Information). However, unlike the cross attention + MLP structure, the number of parameters in a pure MLP phase network scale with system size, making it increasingly demanding for larger systems.

4. Conclusion

In this work, we implemented several different phase network architectures in QiankunNet and computed the ground-state energies of 17 molecules. The results indicate that the simple MLP structure performs well across all test systems. Meanwhile, the cross attention-restricted MLP presents a potential approach for maintaining a constant number of trainable parameters across systems with varying numbers of spin orbitals. This could facilitate the extension of the network architecture beyond small molecules. Furthermore, a fixed parameter structure enables transfer learning across different systems, paving the way for the development of a universal model for various quantum chemistry problems.

The phase of the wave function becomes particularly important in solid-state systems, where periodic boundary conditions necessitate a complex-valued wave function to correctly capture translational symmetry. Recent developments such as DeepSolid have demonstrated the potential of NNQS in representing solid-state wave functions, highlighting the critical role of phase modeling in accurately describing band structures and correlated electron behavior [18]. Incorporating a robust phase network into QiankunNet may thus be a key step toward more efficient and accurate calculation modeling of realistic materials [19].

Another interesting outlook is the combination of NNQS with projection-based quantum Monte Carlo methods under the fixed-node approximation, such as fixed-node diffusion Monte Carlo (DMC), where the NNQS wave functions can be used as trail wave functions to define nodal structures [20]. In this context, different phase network architectures in QiankunNet may induce distinct nodal structures and thus lead to varying performance. This suggests that developing an expressive yet efficient phase networks may offer advantages not only in variational training but also as components in hybrid NNQS–DMC frameworks.

Acknowledgements

The authors thanks support from National Natural Science Foundation of China (Grant No. T2222026), the Strategic Priority Research Program of the Chinese Academy of Sciences (XDB0450101), the Innovation Program for Quantum Science and Technology (2021ZD0303306). This work was supported by the Supercomputing Center of the USTC.

References

[1] Carleo G., Troyer M. Solving the quantum many-body problem with artificial neural networks. *Science,* **355** (2017), 602–606.

[2] Choo K., Neupert T., Carleo G. Two-dimensional frustrated J1–J2 model studied with neural network quantum states. *Phys. Rev. B,* **100** (2019), 125124.

[3] Robledo Moreno J., Carleo G., Georges A., Stokes J. Fermionic wave functions from neural-network constrained hidden states. *Proc. Natl. Acad. Sci. U.S.A.,* **119** (32) (2022), e2122059119.

[4] Pfau D., Spencer J.S., Matthews A.G.d.G., Foulkes W.M.C. Ab-initio solution of the many-electron Schrödinger equation with deep neural networks. *Phys. Rev. Research,* **2** (3) (2020), 033429.

[5] Hermann J., Schätzle Z., Noé F. Deep-neural-network solution of the electronic Schrödinger equation. *Nat. Chem.,* **12** (2020), 891–897.

[6] Li R., Ye H., Jiang D., et al. A computational framework for neural network-based variational Monte Carlo with forward Laplacian. *Nat. Mach. Intell.,* **6** (2024), 209–219.

[7] Scherbela M., Reisenhofer R., Gerard L., et al. Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks. *Nat. Comput. Sci.,* **2** (2022), 331–341.

[8] Choo K., Mezzacapo A., Carleo G. Fermionic neural-network states for ab-initio electronic structure. *Nat. Commun.,* **11** (2020), 2368.

[9] Wu Y., Xu X., Poletti D., Fan Y., Guo C., Shang H. A real neural network state for quantum chemistry. *Mathematics,* **11** (2023), 1417.

[10] Wu Y., Guo C., Fan Y., Zhou P., Shang H. NNQS-Transformer: an efficient and scalable neural network quantum states approach for ab initio quantum chemistry. *Proc. Int. Conf. High Performance Computing, Networking, Storage Anal.,* **2023** (2023).

[11] Barrett T.D., Malyshev A., Lvovsky A. Autoregressive neural-network wavefunctions for ab initio quantum chemistry. *Nat. Mach. Intell.,* **4** (2022), 351–358.

[12] Zhao T., Stokes J., Veerapaneni S. Scalable neural quantum states architecture for quantum chemistry. *Mach. Learn.: Sci. Technol.,* **4** (2023), 025034.

[13] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L., Polosukhin I. Attention is all you need. *arXiv preprint,* **1706.03762** (2017).

[14] Kim S., Chen J., Cheng T., et al. PubChem 2025 update. *Nucleic Acids Res.,* **53** (D1) (2025), D1516–D1525.

[15] Sun Q., Berkelbach T.C., Blunt N.S., Booth G.H., Guo S., Li Z., Liu J., McClain J., Sharma S., Wouters S., Chan G.K-L. PySCF: the Python-based simulations of chemistry framework. *WIREs Comput. Mol. Sci.,* **8** (2018), e1340.

[16] McClean J.R., et al. OpenFermion: the electronic structure package for quantum computers. *Quantum Sci. Technol.,* **5** (2020), 034014.

[17] Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., et al. PyTorch: an imperative style, high-performance deep

learning library. *Adv. Neural Inf. Process. Syst.,* **32** (2019), 8024–8035.

[18] Li X., Li Z., Chen J. Ab initio calculation of real solids via neural network ansatz. *Nat. Commun.,* **13** (2022), 7895.

[19] Fu L., Wu Y., Shang H., Yang J. Transformer-based neural-network quantum state method for electronic band structures of real solids. *J. Chem. Theory Comput.,* **20** (2024), 6218–6226.

[20] Ren W., Fu W., Wu X., et al. Towards the ground state of molecules via diffusion Monte Carlo on neural networks. *Nat. Commun.,* **14** (2023), 1860.