

Representational power of selected neural network quantum states in second quantization

Zhendong Li,^{1, a)} Tong Zhao,^{2, b)} and Bohan Zhang¹

¹⁾Key Laboratory of Theoretical and Computational Photochemistry, Ministry of Education, College of Chemistry, Beijing Normal University, Beijing 100875, China

²⁾State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100095, China

Neural network quantum states emerge as a promising tool for solving quantum many-body problems. However, its successes and limitations are still not well-understood in particular for Fermions with complex sign structures. Based on our recent work [J. Chem. Theory Comput. 21, 10252-10262 (2025)], we generalize the restricted Boltzmann machine to a more general class of states for Fermions, formed by product of ‘neurons’ and hence will be referred to as neuron product states (NPS). NPS builds correlation in a very different way, compared with the closely related correlator product states (CPS) [H. J. Changlani, et al. Phys. Rev. B, 80, 245116 (2009)], which use full-rank local correlators. In contrast, each correlator in NPS contains long-range correlations across all the sites, with its representational power constrained by the simple function form. We prove that products of such simple nonlocal correlators can approximate any wavefunction arbitrarily well under certain mild conditions on the form of activation functions. In addition, we also provide elementary proofs for the universal approximation capabilities of feedforward neural network (FNN) and neural network backflow (NNBF) in second quantization. Together, these results provide a deeper insight into the neural network representation of many-body wavefunctions in second quantization.

I. INTRODUCTION

Accurate and efficient simulation of quantum many-body problems on classical computers has been a long-standing challenge for computational physics and chemistry due to the exponential growth of the size of the Hilbert space as system size increases. During the past decades, a plethora of methods have been developed with their own advantages and disadvantages. From the early days of quantum chemistry, the configuration interaction (CI) method¹ is the most conceptually simple approach to treating correlated electrons. Later, the hierarchy of coupled cluster (CC) methods² becomes more dominant due to their size extensivity. These two methods are “universal”, in the sense that any wavefunction can be approximated by increasing the excitation rank. Tensor network states³ (TNS), which include matrix product states (MPS) as a representative, are better choices for strongly correlated systems⁴. They are also universal, as long as the bond dimension can be arbitrarily large. The universal approximation capability of a wavefunction ansatz is important, because it provides a solid theoretical guarantee for its ultimate accuracy.

Recently, neural networks (NN) become an emerging technique for simulation of quantum many-body problems. Carleo and Troyer employed restricted Boltzmann machine⁵ (RBM), a generative model that can learn a probability distribution over a set of inputs^{6,7}, as a variational ansatz for interacting spin problems on lattices, and achieved good accuracy compared with the

state-of-the-art TNS. Unlike TNS, which encodes area-law entanglement efficiently, RBM can describe volume-law states⁸. Later, other machine learning architectures have also been exploited for spin systems, including feedforward NN (FNN)^{9,10}, convolutional NN (CNN)^{11,12}, recurrent NN (RNN)^{13,14}, autoregressive NN^{15,16}, neural network backflow (NNBF)^{17–20}, etc. The success and practical limitations of these neural network quantum states (NQS) are not fully understood yet, and are being actively studied^{21–25}. Compared with spin systems, the application of neural networks for Fermions remains largely unexplored until very recently^{16,26–31}. In this work, we focus on Fermions on lattices or electrons in molecules described within the second quantization framework, and we refer the readers to Ref.³² for applications of NN in the first quantization.

Inspired by RBM, we recently introduce a more general class of states for fermions composed by product of ‘neurons’, which will be referred to as neuron product states³⁴ (NPS). NPS take the following general form

$$\Psi_{\text{NPS}}(n_1, \dots, n_K) = \prod_{\alpha=1}^{N_h} \phi \left(b_{\alpha} + \sum_{k=1}^K W_{\alpha k} n_k \right), \quad (1)$$

where $\Psi_{\text{NPS}}(n_1, \dots, n_K)$ is the wavefunction in the occupation number representation, $n_k \in \{0, 1\}$, K is the number of spin-orbitals, $\phi(x)$ is an activation function, b_{α} and $W_{\alpha k}$ are real parameters. Each factor ϕ in Eq. (1) is referred to as a neuron, and N_h is the number of neurons. We will denote $\Psi_{\text{NPS}}(n_1, \dots, n_K)$ as $\Psi_{\text{NPS}}(\vec{n})$ and omit the lower/upper limit of summation/product for brevity. This form generalizes the standard RBM for discrete probability, which after tracing out the hidden units can be written as

$$P_{\text{RBM}}(\vec{n}) = e^{\sum_k a_k n_k} \prod_{\alpha} (1 + e^{b_{\alpha} + \sum_k W_{\alpha k} n_k}). \quad (2)$$

^{a)}Electronic mail: zhendongli@bnu.edu.cn

^{b)}Electronic mail: zhaotong@ict.ac.cn

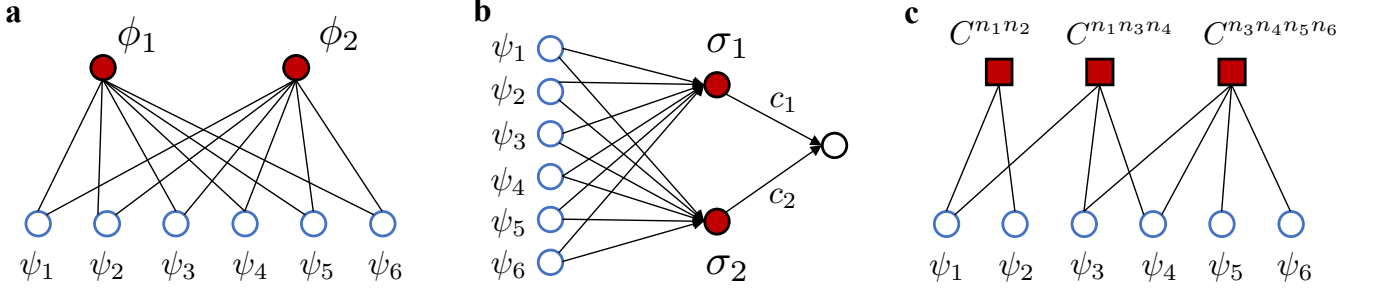


FIG. 1. Illustration of different neural-network wavefunction ansätze for $K = 6$: (a) restricted Boltzmann machine (RBM) and neuron product states (NPS); (b) feedforward neural networks (FNN); (c) correlator product states³³ (CPS). Each red circle in (a) and (b) represents a hidden neuron.

Note that each factor in RBM is positive. Carleo and Troyer⁵ use complex parameters in RBM for wavefunctions that can take negative values. A general $\phi(x)$ is used in NPS, which can take both positive and negative values. In Ref.³⁴, we use $\phi(x) = \cos(x)$ and (1) is used as a correlator multiplied on another wavefunction ansatz. NPS can be pictorially represented in Fig. (1)a. In this work, we focus on NPS itself and ask a fundamental question that whether it can be a universal approximator.

Formally, we can also rewrite Eq. (1) as

$$\Psi_{\text{NPS}}(\vec{n}) = \exp \left[\sum_{\alpha} \kappa \left(b_{\alpha} + \sum_k W_{\alpha k} n_k \right) \right]. \quad (3)$$

with $\kappa(x) = \ln \phi(x)$. The sum $\sum_{\alpha} \kappa(b_{\alpha} + \sum_k W_{\alpha k} n_k)$ displays an apparent similarity with FNN. Specifically, the FNN with one hidden layer, see Fig. 1b, can be written in a similar form as

$$f_{\text{FNN}}(\vec{x}) = \sum_{\alpha} c_{\alpha} \sigma(b_{\alpha} + \sum_k W_{\alpha k} x_k), \quad (4)$$

where σ is an activation function. However, an important difference is that there is no linear combination coefficients c_{α} in Eq. (3). Thus, the classical universal approximation theorem (UAT)^{35–39} for multilayer FNN cannot be applied. Similarly, we find that the proof RBM as universal approximators of discrete distributions^{6,7,40} cannot be applied to NPS (1), as these proofs^{6,40} hinge on the form of factors $1 + e^x$ in Eq. (2), such that one can modify a particular amplitude of wavefunction by choosing appropriate b_{α} and $W_{\alpha k}$.

In this work, we focus on activation functions that are *analytic* in certain domain, which means that they can be represented by Taylor series locally, i.e., $\phi(x) = \sum_{n=0}^{\infty} \frac{\phi^{(n)}(x_0)}{n!} (x - x_0)^n$. Typical examples include elementary functions (polynomials, exponential, trigonometric, etc.) and piecewise defined functions. As in neural networks, they are the most commonly used forms of activation functions⁴¹, e.g., logistic sigmoid $1/(1 + e^{-x})$, hyperbolic tangent $\tanh(x)$, and ReLU⁴² (rectified linear unit). We will show that the necessary and sufficient conditions for NPS (1) as a universal approximator for

quantum states is that (1) $\phi(x)$ can change signs and (2) the logarithm of the activation function $\kappa(x) = \ln \phi(x)$ is not a polynomial with degree less than the number of orbitals K . This result generalizes the universal property of RBM to a more general class of variational ansatz.

Before presenting the proof, we discuss the connections of NPS with other variational ansätze. A closely related class of wavefunctions is the correlator product states (CPS)^{33,43} (including Jastrow factor and entangled plaquette states^{44,45} as special cases), which express the wavefunction as a product of correlators. The tensors $C^{n_1 n_2}$ and $C^{n_1 n_3 n_4}$ in Fig. 1c are examples of local two-site and nonlocal three-site correlators, respectively. A typical two-site CPS reads

$$\Psi_{\text{CPS}}(\vec{n}) = \prod_{i < j} C^{n_i n_j}. \quad (5)$$

In the limit of K -site correlators, CPS becomes exact. We can compare NPS with CPS, which builds the correlation in a drastically different way. Each term in the product (1) contains long-range correlations across all the sites, with the representational power constrained by the simple function form. Thus, a number of products are needed to accurately describe the wavefunction. Another interesting connection is that, by restricting the sum in $\sum_k W_{\alpha k} n_k$ in Eq. (1) to certain sites, a UAT for NPS will imply that every correlator can be approximated by products of ϕ . For instance, the four-site correlator $C^{n_3 n_4 n_5 n_6}$ in Fig. 1c can be approximated arbitrarily well by products of functions of form $\phi(b_{\alpha} + \sum_{k \in \{3,4,5,6\}} W_{\alpha k} n_k)$. This connection will be clearer in the proof of UAT for NPS.

The remaining part of the paper is organized as follows. In Secs. II and III, we provide elementary proofs for UAT of FNN and NNBF in second quantization, respectively, which introduce basic notations and important techniques for proving UAT for NPS in Sec. IV for activation functions under simple conditions. The proof of UAT for NPS for general activation functions is more technically involved and hence is presented in Appendix. Conclusion and outlook are given in the last section.

II. FEEDFORWARD NEURAL NETWORK (FNN) STATES

Applying FNN in Eq. (4) for many-body wavefunction in second quantization leads to the FNN ansatz^{9,10}

$$\Psi_{\text{FNN}}(\vec{n}) = \sum_{\alpha} c_{\alpha} \sigma(b_{\alpha} + \vec{w}_{\alpha}^T \vec{n}), \quad (6)$$

or more compactly

$$\Psi_{\text{FNN}}(\vec{n}) = \vec{c}^T \sigma(\vec{b} + W\vec{n}), \quad (7)$$

where $\vec{c} \in \mathbb{R}^{N_h}$, $W \in \mathbb{R}^{N_h \times K}$, $\vec{b} \in \mathbb{R}^{N_h}$ with N_h being the number of hidden neurons. The 2^K occupation number vectors $\vec{n}$ form a Boolean hypercube $\mathcal{B}^K = \{0, 1\}^K$. We denote the dimension of the Fock space by $D_K = 2^K$. We assume that the activation function σ satisfies

$$\lim_{x \rightarrow -\infty} \sigma(x) = 0, \quad \lim_{x \rightarrow +\infty} \sigma(x) = 1. \quad (8)$$

A typical example is the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$.

Theorem 1. *Given sufficiently large N_h , the FNN ansatz (6) is universal for representing wavefunction in second quantization, denoted as a vector by $\vec{\Psi} \in \mathbb{R}^{D_K}$ in the occupation number representation, in the sense that for any wavefunction $\Psi : \mathcal{B}^K \rightarrow \mathbb{R}$ and any $\epsilon > 0$, there exists a FNN such that the network output $\Psi_{\text{FNN}}(\vec{n})$ satisfies $|\Psi_{\text{FNN}}(\vec{n}) - \Psi(\vec{n})| < \epsilon$ for every $\vec{n} \in \mathcal{B}^K$.*

In fact, this follows directly from the UAT for functions with continuous variables, by embedding the Boolean hypercube $\mathcal{B}^K$ into $\mathbb{R}^K$. But we give a very elementary proof of it for discrete variables using the following lemma.

Lemma 1. *Let $\vec{z} = 2\vec{n} - \vec{e} \in \{-1, 1\}^K$ with $\vec{e} = (1, \dots, 1)^T$, then for $\vec{z} \in \{\vec{z}_j\}_{j=0}^{D_K-1}$, $\vec{z}_i \cdot \vec{z} \in \{K, K-2, \dots, -(K-2), -K\}$. The maximal value K is reached for $\vec{z} = \vec{z}_i$, while the minimal value $-K$ is reached for $\vec{z} = -\vec{z}_i$.*

Without the loss of generality, the D_K vectors $\vec{n}_j$ and $\vec{z}_j$ are assumed to be labeled lexicographically. This lemma can be verified by simple calculations. Now we prove Theorem 1.

Proof. Introduce a matrix $F \in \mathbb{R}^{N_h \times D_K}$ with elements

$$F_{\alpha j} = \sigma(b_{\alpha} + \vec{w}_{\alpha}^T \vec{n}_j). \quad (9)$$

We want to show that with $N_h = 2^K$, F can be made arbitrarily close to an identity matrix by choosing $(\vec{b}, W)$ appropriately. Then, $\vec{c} = \vec{\Psi}$ showing that the weights of FNN store the wavefunction in such case.

For each i , we have $\vec{z}_i^T(\vec{z} - \vec{z}_i) \in \{0, -2, \dots, -2K\}$ using Lemma 1 and $\vec{z}_i^T \vec{z}_i = K$, such that $\vec{z}_i^T(\vec{z} - \vec{z}_i) + 1 = 2\vec{z}_i^T \vec{n} - \vec{z}_i^T \vec{e} - K + 1 \in \{1, -1, \dots, -2K + 1\}$. By choosing

$\vec{w}_i = 2\theta \vec{z}_i$, $b_i = \theta(-\vec{z}_i^T \vec{e} - K + 1)$, and let $\theta \rightarrow +\infty$, we obtain

$$\vec{w}_i^T \vec{n}_j + b_i \rightarrow +\infty, \quad \sigma(\vec{w}_i^T \vec{n}_j + b_i) \rightarrow 1, \quad j = i, \quad (10)$$

$$\vec{w}_i^T \vec{n}_j + b_i \rightarrow -\infty, \quad \sigma(\vec{w}_i^T \vec{n}_j + b_i) \rightarrow 0, \quad j \neq i. \quad (11)$$

Then, F can be made arbitrarily close to an identity matrix; hence the FNN ansatz is universal up to arbitrary precision. $\square$

III. NEURAL NETWORK BACKFLOW (NNBF) STATES

The NNBF ansatz¹⁷⁻²⁰ in second quantization is defined by

$$\Psi_{\text{NNBF}}(\vec{n}) = \det[\phi_{p_k m}(\vec{n})], \quad (12)$$

where $m \in \{1, \dots, N\}$ with N being the number of electrons, p_k ($k \in \{1, \dots, N\}$) represent the indices of occupied orbitals in $\vec{n}$, and $\phi_{p_m}(\vec{n})$ can be viewed as a set of configuration-dependent orbitals generated by a FNN via

$$\phi_{p_m}(\vec{n}) = \vec{c}_{p_m}^T \sigma(\vec{b} + W\vec{n}), \quad (13)$$

where $\vec{c}_{p_m} \in \mathbb{R}^{N_h}$ and $p \in \{1, \dots, K\}$.

Theorem 2. *The NNBF ansatz (12) is universal for sufficiently large N_h .*

Proof. By the proof in the previous theorem, one can choose $N_h = D_K$ and $(b_{\alpha}, \vec{w}_{\alpha})$ such that for given $\vec{n}_i$, only one term in the summation (13) contributes

$$\phi_{p_m}(\vec{n}_i) = c_{p_m, i}, \quad (14)$$

then the wavefunction amplitude is simply

$$\Psi_{\text{NNBF}}(\vec{n}_i) = \det \begin{bmatrix} c_{p_1 1, i} & c_{p_1 2, i} & \cdots & c_{p_1 N, i} \\ c_{p_2 1, i} & c_{p_2 2, i} & \cdots & c_{p_2 N, i} \\ \vdots & \vdots & \ddots & \vdots \\ c_{p_N 1, i} & c_{p_N 2, i} & \cdots & c_{p_N N, i} \end{bmatrix}. \quad (15)$$

By choosing $c_{p_1 1, i} = \Psi(n_i)$, $c_{p_m m, i} = 1$ for $m = 2, \dots, N$, and $c_{p_k m, i} = 0$ for other entries, it is seen that the NNBF ansatz is universal. $\square$

Remarks: For a single (Hartree-Fock) determinant ansatz, $c_{p_k m, i}$ is independent of i .

IV. NEURAL NETWORK PRODUCT STATES (NPS)

We assume that the target wavefunction $\Psi(\vec{n})$ is real, and all the parameters $(W, \vec{b})$ are also real in NPS defined in Eq. (1). Besides, we assume that activation function σ is in $(-1, 1)$ and satisfies

$$\lim_{x \rightarrow +\infty} \sigma(x) = 1. \quad (16)$$

A typical example is $f(x) = \tanh(x)$. Under such condition, we can give an elementary proof that

Theorem 3. *The NPS ansatz with activation function $\sigma \in (-1, 1)$ and satisfying Eq. (16) is universal for sufficiently large N_h .*

In fact, NPS with more general activation function can also be proved universal.

Theorem 4. *The NPS ansatz is universal if and only if*

1. $\phi(x)$ can produce both positive and negative values (i.e., $\exists x_1, x_2$ such that $\phi(x_1) > 0$ and $\phi(x_2) < 0$),
2. $\ln \phi(x)$ is not a polynomial of degree less than K .

However, its proof is more technically involved, and hence we present it in Appendix. The proof of Theorem 3 is simpler based on the following lemma.

Lemma 2 (Gordan's lemma). *Let $A \in \mathbb{R}^{m \times n}$ be a matrix. Then exactly one of the following statement is true:*

1. *There exists $\vec{x} \in \mathbb{R}^n$ such that $A\vec{x} > 0$ (componentwise).*
2. *There exists $\vec{y} \in \mathbb{R}^m$, $\vec{y} \neq \vec{0}$, with $y_k \geq 0$ (componentwise) such that $A^T \vec{y} = 0$.*

This is a fundamental result in linear algebra and convex analysis⁴⁶. Using it, we first prove the following useful result.

Lemma 3. *Let V denotes the set of vectors*

$$V = \{\vec{n}_j - \vec{n}_i : j \neq i\}, \quad (17)$$

then there exists a vector $\vec{u} \in \mathbb{R}^K$ such that $\vec{u}^T \vec{v} > 0$, $\forall \vec{v} \in V$.

Proof. If no such $\vec{u}$ exists, then by Gordan's lemma, case 2 holds, that is, there exist nonnegative coefficients $\lambda_j \geq 0$ ($j \neq i$, not all zero) such that

$$\sum_{j \neq i} \lambda_j (\vec{n}_j - \vec{n}_i) = \vec{0}. \quad (18)$$

We now show that this will lead to contradictions.

Let $S = \sum_{j \neq i} \lambda_j$. Since not all λ_j are zero, we have $S > 0$. Rearranging Eq. (18), we obtain

$$\sum_{j \neq i} \lambda_j \vec{n}_j = S \vec{n}_i. \quad (19)$$

Consider the p -th component of this equation, if $(\vec{n}_i)_p = 1$, then $\sum_{j \neq i} \lambda_j (\vec{n}_j)_p = S$. Since $\sum_{j \neq i} \lambda_j (\vec{n}_j)_p \leq \sum_{j \neq i} \lambda_j = S$, we have $(\vec{n}_j)_p = 1$ for all j with $\lambda_j > 0$. Likewise, if $(\vec{n}_i)_p = 0$, then $\sum_{j \neq i} \lambda_j (\vec{n}_j)_p = 0$, suggesting that for all j with $\lambda_j > 0$, $(\vec{n}_j)_p = 0$. In sum, for every j with $\lambda_j > 0$, Eq. (18) implies that $\vec{n}_j = \vec{n}_i$, which contradicts the fact that $j \neq i$. Therefore, our assumption that no such $\vec{u}$ exists must be false. $\square$

Remarks: The set V is a finite set of nonzero vectors, and it does not contain any pair of opposite vectors. Because if $\vec{v}$ and $-\vec{v}$ were both in V , then there exists j and k , such that $\vec{n}_j - \vec{n}_i = -\vec{n}_k + \vec{n}_i$, which implies that $\vec{n}_j + \vec{n}_k = 2\vec{n}_i$. This is impossible for binary vectors $\vec{n}_j \neq \vec{n}_i$ and $\vec{n}_k \neq \vec{n}_i$. Geometrically, since V is finite and contains no opposite vectors, the convex cone generated by V does not contain the origin. Therefore, there exists a hyperplane through the origin that separates the origin from V , meaning all vectors in V are on one side of the hyperplane. The normal vector to this hyperplane (pointing towards V) can be taken as $\vec{u}$, so that $\vec{u}^T \vec{v} > 0$ for all $\vec{v} \in V$.

Based on Lemma 3, we can prove Theorem 3.

Proof. Let $N_h = D_K$. The key idea is to construct N_h functions $\{g_i\}$, where $g_i(\vec{n}) = \phi(\vec{w}_i^T \vec{n} + b_i)$ with appropriately chosen parameters, such that for each i : (1) $g_i(\vec{n}_i) \rightarrow \Psi(\vec{n}_i)$ and (2) $g_i(\vec{n}_j) \rightarrow 1$ for $j \neq i$.

The first part of the above statement is simple to prove. Since $\Psi(\vec{n}_i) \in [-1, 1]$ and ϕ is continuous with range $(-1, 1)$, there exists $x_0 \in \mathbb{R}$ such that $\phi(x_0)$ is arbitrarily close to $\Psi(\vec{n}_i)$.

For the second part, we consider the set of vectors (17). By Lemma (3), there exists a vector $\vec{u}_i \in \mathbb{R}^K$ such that $\vec{u}_i^T \vec{v} > 0$, $\forall \vec{v} \in V$. Then, define

$$\vec{w}_i = \theta \vec{u}_i, \quad b_i = x_0 - \vec{w}_i^T \vec{n}_i, \quad (20)$$

where $\theta > 0$ is a scaling factor to be chosen large. We have $\vec{w}_i^T \vec{n}_i + b_i = x_0$ and for $j \neq i$,

$$\vec{w}_i^T \vec{n}_j + b_i = x_0 + \theta \vec{u}_i^T (\vec{n}_j - \vec{n}_i). \quad (21)$$

Since $\vec{u}_i^T (\vec{n}_j - \vec{n}_i) > 0$, by taking θ sufficiently large, we can make $\vec{w}_i^T \vec{n}_j + b_i$ large enough, such that $\phi(\vec{w}_i^T \vec{n}_j + b_i)$ is arbitrarily close to 1.

Then, by multiplying N_h such functions $g_i(\vec{n})$ in Eq. (1), the NPS ansatz can approximate any wavefunction arbitrarily well. $\square$

The above proof relies on the condition (16), and hence does not generalize to the activation function $f(x) = \cos(x)$ used in Ref.³⁴. In Appendix, we give a more general proof for Theorem 4, which also reveals a deeper connection with CPS.

V. CONCLUSION

In this work, we generalize RBM to a more broad class of states and prove the universal approximation capability of NPS given suitable active function. This lays the foundation for future exploration of NPS for strongly correlated fermions. While in our previous work³⁴ we used such functions with $\phi(x) = \cos(x)$ as correlators to enhance the expressivity of other variational ansätze, the present work demonstrate that the NPS itself is also a valid variational ansatz, which can be optimized using variational Monte Carlo^{47–49}. We expect the choice

of activation function ϕ will be important for NPS in practice, and the combination with composition of functions in deep learning⁵⁰ for more efficient representation is also promising. These open questions will be explored in future, which can extend RBM to more challenging systems.

ACKNOWLEDGMENT

This work was supported by the Quantum Science and Technology-National Science and Technology Major Project(2023ZD0300200) and the Fundamental Research Funds for the Central Universities.

APPENDIX: PROOF OF THEOREM 4 FOR NPS WITH GENERAL ACTIVATION FUNCTIONS

A. Representation of Ψ by multilinear polynomials

We introduce a special representation of the many-body wavefunction by interpreting $\Psi(\vec{n})$ as a pseudo-Boolean function⁵¹, which defines a mapping from the Boolean hypercube $\mathcal{B}^K$ to a field F ($\mathbb{R}$ or $\mathbb{C}$). For convenience, we will work with the variable $z_k = -2n_k + 1 \in \{1, -1\}$. Using the inversion $n_k = \frac{1-z_k}{2}$, the same wavefunction but expressed in $\{z_k\}$ can be found as

$$\Phi(z_1, \dots, z_K) = \Psi\left(\frac{1-z_1}{2}, \dots, \frac{1-z_K}{2}\right). \quad (22)$$

It can be represented uniquely as a multilinear polynomial⁵¹

$$\Phi(z_1, \dots, z_K) = \sum_{x_1 \dots x_K} \hat{\Phi}^{x_1 \dots x_K} z_1^{x_1} \dots z_K^{x_K}, \quad (23)$$

where $x_k \in \{0, 1\}$, $\hat{\Phi}^{x_1 \dots x_K}$ is a tensor with 2^K elements, which can be called as the "Fourier coefficients" of the function Φ . Eq. (23) can be proved can realizing that each factor $z_k^{x_k} = \begin{cases} 1, & z_k = 1 \\ (-1)^{x_k}, & z_k = -1 \end{cases}$ can be viewed as

an element of the Hadamard matrix $H_k = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ (up to a constant factor). Then, we can rewrite Eq. (23)

in a compact form

$$\Phi = 2^{K/2} H^{\otimes K} \hat{\Phi}, \quad (24)$$

where both Φ and $\hat{\Phi}$ are viewed as vectors and $H^{\otimes K} = H_1 \otimes \dots \otimes H_K$ is the Walsh-Hadamard transform. By the involutory property $H_k^2 = I$, Eq. (24) can be inverted as

$$\hat{\Phi} = 2^{-K/2} H^{\otimes K} \Phi, \quad (25)$$

which shows that $\hat{\Phi}$ is uniquely determined by Φ . Eq. (25) can be written explicitly as

$$\hat{\Phi}^{x_1 \dots x_K} = 2^{-K} \sum_{z_1 \dots z_K} z_1^{x_1} \dots z_K^{x_K} \Phi(z_1, \dots, z_K). \quad (26)$$

In the terminology of quantum computing⁵², $\hat{\Phi}$ is just the wavefunction in the eigenbasis $\{|+\rangle, |-\rangle\}$ of the Pauli X operator for each qubit, while Φ is the wavefunction in the computational basis $\{|0\rangle, |1\rangle\}$.

B. Representation of functions $f(b + \vec{w}^T \vec{z})$

The above result shows that any discrete function: $\Psi : \{+1, -1\}^n \rightarrow F$ with the field $F = \mathbb{R}$ or $\mathbb{C}$ can be represented by a multilinear polynomial. Apply this result to the function of the form $f(b + \vec{w}^T \vec{z})$, where f can be ϕ in Eq. (1) or $\kappa = \ln \phi$ in Eq. (3), leads to

$$f(b + \vec{w}^T \vec{z}) = 2^{K/2} H^{\otimes K} \hat{f}. \quad (27)$$

The inversion (25) gives

$$\hat{f}^{x_1 x_2 \dots x_K} = 2^{-K} \sum_{z_1 z_2 \dots z_K} z_1^{x_1} z_2^{x_2} \dots z_K^{x_K} f(b + \vec{w}^T \vec{z}). \quad (28)$$

To better illustrate this formula, we give two concrete examples for $K = 1$ and $K = 2$. For $K = 1$,

$$\hat{f}^0 = \frac{1}{2} [f(b + w_1) + f(b - w_1)], \quad (29)$$

$$\hat{f}^1 = \frac{1}{2} [f(b + w_1) - f(b - w_1)], \quad (30)$$

and for $K = 2$,

$$\hat{f}^{00} = \frac{1}{4} [f(b + w_1 + w_2) + f(b - w_1 + w_2) + f(b + w_1 - w_2) + f(b - w_1 - w_2)], \quad (31)$$

$$\hat{f}^{10} = \frac{1}{4} [f(b + w_1 + w_2) - f(b - w_1 + w_2) + f(b + w_1 - w_2) - f(b - w_1 - w_2)], \quad (32)$$

$$\hat{f}^{01} = \frac{1}{4} [f(b + w_1 + w_2) + f(b - w_1 + w_2) - f(b + w_1 - w_2) - f(b - w_1 - w_2)], \quad (33)$$

$$\hat{f}^{11} = \frac{1}{4} [f(b + w_1 + w_2) - f(b - w_1 + w_2) - f(b + w_1 - w_2) + f(b - w_1 - w_2)]. \quad (34)$$

In the later part, we will use the asymptotic behavior of $\hat{f}$ for small $\vec{\omega}$. To analyze this, we first use the Taylor expansion

$$f(b + \vec{\omega}^T \vec{z}) = \sum_{m_k \geq 0} \frac{1}{m_1! \cdots m_K!} f^{\sum_k m_k}(b) \prod_k \omega_k^{m_k} z_k^{m_k} \quad (35)$$

$$\begin{aligned} \hat{f}^{x_1 x_2 \cdots x_K} &= \frac{1}{2^K} \sum_{z_1 z_2 \cdots z_K} z_1^{x_1} \cdots z_K^{x_K} \sum_{m_k \geq 0} \frac{1}{m_1! \cdots m_K!} f^{\sum_k m_k}(b) \prod_k \omega_k^{m_k} z_k^{m_k} \\ &= \frac{1}{2^K} \sum_{m_k \geq 0} \frac{1}{m_1! \cdots m_K!} f^{\sum_k m_k}(b) \prod_k \omega_k^{m_k} \sum_{z_k} z_k^{x_k + m_k} \\ &= \frac{1}{2^K} \sum_{m_k \geq 0} \frac{1}{m_1! \cdots m_K!} f^{\sum_k m_k}(b) \prod_k \omega_k^{m_k} 2\delta_{x_k + m_k, e} \\ &= \sum_{m_k \geq 0} \frac{1}{(x_1 + 2m_1)! \cdots (x_K + 2m_K)!} f^{\sum_k x_k + \sum_k 2m_k}(b) \prod_k \omega_k^{x_k + 2m_k}, \end{aligned} \quad (36)$$

by using $\sum_z z^n = 1 + (-1)^n = 2\delta_{n,e}$, where we introduce a shorthand notation $\delta_{n,e}$ to represent that the value is 1 if and only if n is an even integer. Introducing a parameter ϵ for each ω_k to count the order, the asymptotic behavior of $\hat{f}$ for small ϵ goes as

$$\hat{f}^{x_1 x_2 \cdots x_K} = f^{\sum_k x_k}(b) \epsilon^{\sum_k x_k} \prod_k \omega_k^{x_k} + O(\epsilon^{\sum_k x_k + 2}). \quad (37)$$

Using Eq. (36), we can also find the parity of $\hat{f}^{n_1 n_2 \cdots n_K}$ with respect to the sign change of $\vec{\omega}$. If one of its component ω_k changes sign, with the new vector denoted by $\vec{\omega}'$, the corresponding $f(b + \vec{\omega}'^T \vec{z})$ has an expansion with coefficients $(-1)^{x_k} \hat{f}^{x_1 x_2 \cdots x_K}$.

C. Introduction of intermediate functions $\tilde{\Psi}(\vec{n})$ and $\tilde{\Psi}_+(\vec{n})$

While Eq. (22) adopts distinct notations (Ψ and Φ) to emphasize the mathematical difference between functions with different types of arguments ($\vec{n}$ or $\vec{z}$), in subsequent sections we will use the same symbol for the wavefunction for the sake of brevity. Whether the representation $\Phi(\vec{z})$ or $\Psi(\vec{n})$ is intended will be clear from the context.

The wavefunction $\Psi(\vec{n})$ can contain zeros and negative values. We first show that we can construct an intermediate state $\tilde{\Psi}$, which is arbitrarily close to Ψ but with nonzero entries. Suppose the number of zeros in Ψ is M_0 , if $M_0 = 0$, then we simply choose $\tilde{\Psi} \triangleq \Psi$. Otherwise, if $M_0 \geq 1$, we define

$$\tilde{\Psi}(\vec{n}) \triangleq \begin{cases} \epsilon/\sqrt{M_0}, & \Psi(\vec{n}) = 0 \\ \Psi(\vec{n})\sqrt{1-\epsilon^2}, & \Psi(\vec{n}) \neq 0 \end{cases}, \quad (38)$$

which is normalized $\langle \tilde{\Psi} | \tilde{\Psi} \rangle = 1$ and satisfies

$$|\Psi(\vec{n}) - \tilde{\Psi}(\vec{n})| = \begin{cases} \epsilon/\sqrt{M_0}, & \Psi(\vec{n}) = 0 \\ |\Psi(\vec{n})|(1-\sqrt{1-\epsilon^2}), & \Psi(\vec{n}) \neq 0 \end{cases} \quad (39)$$

where $f^{\sum_k m_k}(b)$ represents the $\sum_k m_k$ -th order derivative of $f(x)$ taken at $x = b$. It deserves to point that although Eq. (35) looks quite similar to Eq. (23), they are different. To be more precise, we can find $\hat{f}$ as

In both case, we have $|\Psi(\vec{n}) - \tilde{\Psi}(\vec{n})| < \epsilon$ using $|\Psi(\vec{n})| \leq 1$ and $1 - \sqrt{1 - \epsilon^2} < \epsilon$ for $\epsilon < 1$.

Next, we show the following theorem holds.

Theorem 5. $\tilde{\Psi}(\vec{n})$ can be written as

$$\tilde{\Psi}(\vec{n}) = s(\vec{n}) \tilde{\Psi}_+(\vec{n}), \quad s(\vec{n}) = \prod_{\alpha=1}^{M_-} \phi(b_\alpha + \vec{w}_\alpha^T \vec{n}) \quad (40)$$

where $s(\vec{n})$ has the same sign structure as $\tilde{\Psi}(\vec{n})$, and hence $\tilde{\Psi}_+(\vec{n})$ is a function with all positive values.

Proof. Let the index set $I_- = \{i : \tilde{\Psi}(\vec{n}_i) < 0\}$ contains all the indices where $\tilde{\Psi}(\vec{n}_i)$ is negative and M_- be the number of negative values in $\tilde{\Psi}(\vec{n})$, that is, $M_- = |I_-|$. We show that for each $i \in I_-$, we can construct a factor $\phi(b_\alpha + \vec{w}_\alpha^T \vec{n})$ such that $\phi(b_\alpha + \vec{w}_\alpha^T \vec{n}_i) < 0$ and $\phi(b_\alpha + \vec{w}_\alpha^T \vec{n}_j) > 0$ for $j \neq i$. Without loss of generality, we can assume $\phi(x) > 0$ for $x \in (x_0 - \epsilon, x_0)$ and $\phi(x) < 0$ for $x \in (x_0, x_0 + \epsilon)$ based on the first condition in Theorem 4. The case with reversed signs follows similarly.

By the hyperplane separation theorem⁵³ applied to the hypercube vertices $\{\vec{n}_j\}_{j=0}^{D_K-1}$, there exists $\vec{u} \in \mathbb{R}^K$ and $c \in \mathbb{R}$ such that:

$$\vec{u}^T \vec{n}_i + c > 0, \quad (41)$$

$$\vec{u}^T \vec{n}_j + c < 0, \quad \text{for all } j \neq i. \quad (42)$$

Define the affine function

$$L(\vec{n}) = \theta(\vec{u}^T \vec{n} + c) + x_0, \quad (43)$$

where $\theta > 0$ is a scaling parameter to be determined. Let

$$M = \vec{u}^T \vec{n}_i + c > 0, \quad (44)$$

$$m = \max_{j \neq i} |\vec{u}^T \vec{n}_j + c| > 0. \quad (45)$$

Choose θ such that:

$$\theta < \frac{\epsilon}{\max(M, m)}, \quad (46)$$

then for all configurations,

$$|L(\vec{n}_j) - x_0| = \theta |\vec{u}^T \vec{n}_j + c| < \epsilon \quad \text{for all } j, \quad (47)$$

that is,

$$L(\vec{n}_i) \in (x_0, x_0 + \epsilon), \quad (48)$$

$$L(\vec{n}_j) \in (x_0 - \epsilon, x_0), \quad \text{for all } j \neq i. \quad (49)$$

By the sign properties of ϕ around x_0 , we obtain

$$\phi(L(\vec{n}_i)) < 0, \quad (50)$$

$$\phi(L(\vec{n}_j)) > 0 \quad \text{for all } j \neq i. \quad (51)$$

Setting $\vec{w} = \theta \vec{u}$ and $b = \theta c + x_0$ completes the proof. $\square$

This proof demonstrates that we can construct individual factors that control the sign pattern of NPS, which is a crucial step in establishing the universal approximation capability of NPS. In the following, we prove that $\tilde{\Psi}_+(\vec{n})$ can also be approximate arbitrarily well using NPS with only positive factors.

D. Recursive approximation of $\tilde{\Psi}_+(\vec{n})$ by NPS

To approximate $\tilde{\Psi}_+(\vec{n})$ using NPS, the second condition in Theorem 4 is important. Its necessity can be seen easily as follows: If $\kappa(x) = \ln \phi(x)$ is a polynomial with degree less than K , which implies that $\phi(x) = e^{P_n(x)}$ with $P_n(x) = \sum_{i=0}^n p_i x^i$ being a polynomial in x with degree $n < K$, then the product of two neurons becomes

$$\begin{aligned} \phi(b_1 + \vec{w}_1^T \vec{z}) \phi(b_2 + \vec{w}_2^T \vec{z}) &= e^{P_n(b_1 + \vec{w}_1^T \vec{z}) + P_n(b_2 + \vec{w}_2^T \vec{z})} \\ &= e^{Q_n(z_1, z_2, \dots, z_K)}, \end{aligned} \quad (52)$$

where $Q_n(z_1, z_2, \dots, z_K)$ is a multilinear polynomial in $\{z_k\}$ with degree $n < K$. This reveals that the product of ϕ does not have the representational power to approximate terms with degree higher than n , e.g., $e^{g z_1 z_2 \dots z_K}$.

The sufficiency can be proved as follows. Specifically, we will show that if $\kappa(x) = \ln \phi(x)$ is not a polynomial in x with degree less than K , then there exists a set of parameters $\{N_i, b_i, \vec{w}_i\}_{i=1}^{D_K-1}$ such that $\tilde{\Psi}(\vec{z})$ can be approximated arbitrarily well by NPS, viz.,

$$|\tilde{\Psi}_+(\vec{z}) - \Theta(\vec{z})| < \epsilon, \quad (53)$$

with

$$\Theta(\vec{z}) = \mathcal{N} \prod_{\alpha=1}^{D_K-1} \phi^{N_\alpha}(b_\alpha + \vec{w}_\alpha^T \vec{z}), \quad (54)$$

for any given ϵ , where $\mathcal{N} > 0$ is a normalization constant and $\phi(b_\alpha + \vec{w}_\alpha^T \vec{z}) > 0$. In the following, we show that

$$|\ln \tilde{\Psi}_+(\vec{z}) - \ln \Theta(\vec{z})| < \epsilon, \quad (55)$$

by matching the multilinear coefficient of $\ln \tilde{\Psi}_+(\vec{z})$ using neurons in $\ln \Theta(\vec{z})$ in a recursive way. Then, by the continuity of e^x , Eq. (53) holds.

To this end, we order the power set of $I = \{1, 2, \dots, K\}$ into $K+1$ tiers

$$\begin{cases} \text{tier } K & : \{1, \dots, K\} \\ \text{tier } K-1 & : \{1, \dots, K-1\}, \dots, \{2, \dots, K\} \\ & \dots \\ \text{tier } 2 & : \{1, 2\}, \dots, \{K-1, K\} \\ \text{tier } 1 & : \{1\}, \{2\}, \dots, \{K\} \\ \text{tier } 0 & : \emptyset. \end{cases}$$

For brevity, we define the symbol $I_{t,k}$ to represent the k -th subset at the tier t . Let $g(\vec{z}) = \ln \tilde{\Psi}_+(\vec{z})$, then using the multilinear expansion (23) for $g(\vec{z})$ gives

$$g(\vec{z}) = \hat{g}_{I_0} + \sum_{k=1}^{C_K^1} \hat{g}_{I_{1,k}} \mathcal{Z}_{I_{1,k}} + \dots + \hat{g}_{I_K} \mathcal{Z}_{I_K}, \quad (56)$$

or equivalently

$$\tilde{\Psi}_+(\vec{z}) = \exp \left(\hat{g}_{I_0} + \sum_{k=1}^{C_K^1} \hat{g}_{I_{1,k}} \mathcal{Z}_{I_{1,k}} + \dots + \hat{g}_{I_K} \mathcal{Z}_{I_K} \right) \quad (57)$$

where we have introduced an abbreviation $\mathcal{Z}_{I_{t,k}}$ for the factor $z_{p_1} z_{p_2} \dots z_{p_t}$ with $p_1 < p_2 < \dots < p_t$ and $p_i \in I_{t,k}$, and $\hat{g}_{I_{t,k}}$ to represent the Fourier coefficient $\hat{g}^{x_1, \dots, x_K}$ with $x_k = 1$ for $k \in I_{t,k}$.

Eq. (54) gives

$$\begin{aligned} \ln \Theta(\vec{z}) &= \sum_{\alpha=1}^{D_K-1} N_\alpha \ln \phi(b_\alpha + \vec{w}_\alpha^T \vec{z}) + \ln \mathcal{N} \\ &= \sum_{\alpha=1}^{D_K-1} N_\alpha \kappa(b_\alpha + \vec{w}_\alpha^T \vec{z}) + \ln \mathcal{N}. \end{aligned} \quad (58)$$

The basic idea for proving Eq. (55) is to use one term in this expansion to match the term with the highest power of multilinear polynomials, such as Eq. (56), at each time in a recursive way starting from the highest tier in Eq. (56). Specifically, we use the following form

$$\ln \Theta(\vec{z}) = \sum_{I_{t,k}, t > 0} N_{I_{t,k}} \kappa(b_{I_{t,k}} + \vec{w}_{I_{t,k}}^T \vec{z}_{I_{t,k}}) + \ln \mathcal{N}, \quad (59)$$

where the sum is replaced by the sum over subset from tier K to tier 1, and $\vec{z}_{I_{t,k}}$ represents the vector consisted of z_{p_i} with $p_i \in I_{t,k}$. This structure is illustrated in a schematic way in Fig. 2. We will show how to choose $\{N_{I_{t,k}}, b_{I_{t,k}}, \omega_{I_{t,k}}\}$ such that Eq. (55) holds.

We start by giving a recipe to approximate the higher degree term. We rewrite Eq. (56) as

$$g(\vec{z}) \triangleq g_K^{(0)}(z_{I_K}) = g_{K-1}^{(0)}(z_{I_K}) + \hat{g}_{I_K}^{(0)} \mathcal{Z}_{I_K}, \quad (60)$$

where $g_{K-1}^{(0)}(z_{I_K})$ collects the remaining $2^K - 1$ terms is the multilinear polynomial in variables $z_{I_K} =$

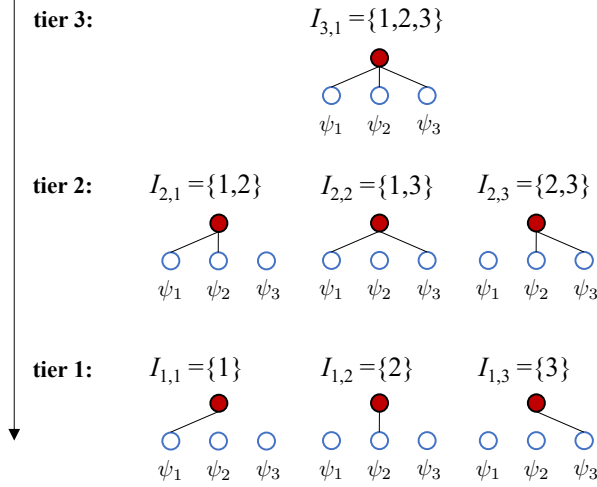


FIG. 2. Hierarchical structure of neurons $\kappa(b_{I_{t,k}} + \vec{w}_{I_{t,k}}^T \vec{z}_{I_{t,k}})$ used to eliminate the leading terms in $\mathcal{Z}_{I_{t,k}}$ recursively. Specifically, we can match the coefficients in Eq. (56) from the highest tier to the lowest tier recursively using neurons with supports shown in the figure.

$(z_1, \dots, z_K)$, with the subscript $K-1$ indicating the highest degree. Now we choose one term in Eq. (59), and rewritten it as

$$\kappa(b_{I_K} + \vec{w}_{I_K}^T \vec{z}_{I_K}) = \kappa_{K-1}(z_{I_K}) + \hat{\kappa}_{I_K} \mathcal{Z}_{I_K}, \quad (61)$$

where $\kappa_{K-1}(z_{I_K})$ and $\hat{\kappa}_{I_K}$ are both functions of $(b_{I_K}, \vec{w}_{I_K})$, but for brevity we have omitted their function dependence. Our goal is to use it to match the term with the highest power $\hat{g}_{I_K}^{(0)} \mathcal{Z}_{I_K}$ in Eq. (60) via

$$\begin{aligned} & g_K^{(0)}(z_{I_K}) - N_{I_K} \kappa(b_{I_K} + \vec{w}_{I_K}^T \vec{z}_{I_K}) \\ &= [g_{K-1}^{(0)}(z_{I_K}) - N_{I_K} \kappa_{K-1}(z_{I_K})] + (\hat{g}_{I_K}^{(0)} - N_{I_K} \hat{\kappa}_{I_K}) \mathcal{Z}_{I_K} \\ &\triangleq g_{K-1}^{(1)}(z_{I_K}) + (\hat{g}_{I_K}^{(0)} - N_{I_K} \hat{\kappa}_{I_K}) \mathcal{Z}_{I_K}, \end{aligned} \quad (62)$$

where $g_{K-1}^{(1)}(z_{I_K})$ is a new polynomial with highest power equal to $K-1$. In a similar way, we can use C_K^{K-1} terms in Eq. (59) to match the terms with power equal to $K-1$ in $g_{K-1}^{(1)}(z_{I_K})$, viz.,

$$\begin{aligned} & g_{K-1}^{(1)}(z_{I_K}) - \sum_{k=1}^{C_K^{K-1}} N_{I_{K-1,k}} \kappa(b_{I_{K-1,k}} + \vec{w}_{I_{K-1,k}}^T \vec{z}_{I_{K-1,k}}) \\ &= [g_{K-2}^{(1)}(z_{I_K}) - \sum_{k=1}^{C_K^{K-1}} N_{I_{K-1,k}} \kappa_{K-2}(z_{I_{K-1,k}})] \\ &+ \sum_{k=1}^{C_K^{K-1}} (\hat{g}_{I_{K-1,k}}^{(1)} - N_{I_{K-1,k}} \hat{\kappa}_{I_{K-1,k}}) \mathcal{Z}_{I_{K-1,k}}, \\ &\triangleq g_{K-2}^{(2)}(z_{I_K}) + \sum_{k=1}^{C_K^{K-1}} (\hat{g}_{I_{K-1,k}}^{(1)} - N_{I_{K-1,k}} \hat{\kappa}_{I_{K-1,k}}) \mathcal{Z}_{I_{K-1,k}}, \end{aligned} \quad (63)$$

This procedure can be carried out recursively down to tier 1, such that

$$\begin{aligned} & g_1^{(K-1)}(z_{I_K}) - \sum_{k=1}^{C_K^1} N_{I_{1,k}} \kappa(b_{I_{1,k}} + \vec{w}_{I_{1,k}}^T \vec{z}_{I_{1,k}}) \\ &= [g_0^{(K-1)}(z_{I_K}) - \sum_{k=1}^{C_K^1} N_{I_{1,k}} \kappa_0(z_{I_{1,k}})] \\ &+ \sum_{k=1}^{C_K^1} (\hat{g}_{I_{1,k}}^{(K-1)} - N_{I_{1,k}} \hat{\kappa}_{I_{1,k}}) \mathcal{Z}_{I_{1,k}}, \\ &= g_0^{(K)}(z_{I_K}) + \sum_{k=1}^{C_K^1} (\hat{g}_{I_{1,k}}^{(K-1)} - N_{I_{1,k}} \hat{\kappa}_{I_{1,k}}) \mathcal{Z}_{I_{1,k}}. \end{aligned} \quad (64)$$

Now $g_0^{(K)}(z_{I_K})$ is just a constant and we can simply choose $\ln \mathcal{N} = g_0^{(K)}(z_{I_K})$ to exactly match it.

Summing up these equations from tier K to tier 1 leads to

$$\begin{aligned} & \ln \tilde{\Psi}_+(\vec{z}) - \ln \Theta(\vec{z}) \\ &= \sum_{t=1}^K \sum_{k=1}^{C_K^t} (\hat{g}_{I_{K-t,k}}^{(t)} - N_{I_{K-t,k}} \hat{\kappa}_{I_{K-t,k}}) \mathcal{Z}_{I_{K-t,k}} \end{aligned} \quad (65)$$

with $\ln \Theta(\vec{z}) = \sum_{t=1}^K \sum_{k=1}^{C_K^t} N_{I_{K-t,k}} \kappa(b_{I_{K-t,k}} + \vec{w}_{I_{K-t,k}}^T \vec{z}_{I_{K-t,k}}) + \ln \mathcal{N}$. Suppose by choosing $\{N_{I_{K-t,k}}, b_{I_{K-t,k}}, \vec{w}_{I_{K-t,k}}\}$, we can make

$$|\hat{g}_{I_{K-t,k}}^{(n)} - N_{I_{K-t,k}} \hat{\kappa}_{I_{K-t,k}}| \leq (\epsilon/K)^{K-t}, \quad (66)$$

for an arbitrarily small ϵ , then

$$\begin{aligned} & |\ln \tilde{\Psi}_+(\vec{z}) - \ln \Theta(\vec{z})| \\ &\leq \sum_{t=1}^K \sum_{k=1}^{C_K^t} |\hat{g}_{I_{K-t,k}}^{(n)} - N_{I_{K-t,k}} \hat{\kappa}_{I_{K-t,k}}| \cdot |\mathcal{Z}_{I_{K-t,k}}| \\ &\leq \sum_{t=1}^K C_K^t (\epsilon/K)^{K-t} = (1 + \epsilon/K)^K - 1 \leq 2\epsilon, \end{aligned} \quad (67)$$

for $0 \leq \epsilon < 1$ and using $|\mathcal{Z}_{I_{K-t,k}}| = 1$. This show that if Eq. (66) holds, then by an recursive procedure, we can construct an approximation to $\ln \tilde{\Psi}_+(\vec{z})$ arbitrarily well using NPS.

E. Proof of Eq. (66)

We show that for any tolerance $\delta > 0$, there exist integer N , and real parameters b and $\vec{w}$ such that

$$|\hat{g} - N \hat{\kappa}(b, \vec{w})| \leq \delta, \quad (68)$$

which proves Eq. (66). For simplicity, the subscripts for symbols in Eq. (66) have been omitted. The key observation is that the Fourier coefficients $\hat{\kappa}(b, \vec{w})$ can be arbitrarily small, see Eq. (37). Specifically, from the

small- ω expansion of $\hat{\kappa}(b, \vec{\omega})$, we know that for small $\vec{\omega}$, the Fourier coefficient behaves as:

$$\hat{\kappa}(b, \vec{\omega}) = A(b) \prod_k \omega_k^{x_k} + O\left(\|\omega\|^{\sum x_k + 2}\right), \quad (69)$$

where $A(b)$ is an analytic function of b . As $\vec{\omega} \rightarrow 0$, $\hat{\kappa}(b, \vec{\omega})$ goes to 0 continuously, which means that we can make $\hat{\kappa}(b, \omega)$ arbitrarily small by choosing $\vec{\omega}$ sufficiently small. [If $A(b) \equiv 0$, we can always use the higher order term whose coefficient is nonzero. This is possible as the assumption is that $\kappa(x)$ is not a polynomial of degree less than K .]

For simplicity, we assume that $\hat{g}$ and $\hat{\kappa}(b, \vec{\omega})$ are of the same sign, otherwise, we can flip the sign of $\hat{\kappa}(b, \vec{\omega})$ by making the sign of one ω_k with $x_k = 1$ in $\vec{\omega}$ flipped, see the discussion after Eq. (37). Now, we define the integer $N = \text{round}\left(\frac{\hat{g}}{\hat{\kappa}(b, \vec{\omega})}\right)$, then the error due to rounding is given by

$$\left| \frac{\hat{g}}{\hat{\kappa}(b, \vec{\omega})} - N \right| \leq \frac{1}{2}, \quad (70)$$

or equivalently,

$$|\hat{g} - N\hat{\kappa}(b, \vec{\omega})| \leq \frac{|\hat{\kappa}(b, \vec{\omega})|}{2}. \quad (71)$$

Since $\hat{\kappa}(b, \vec{\omega})$ can be made arbitrarily small by choosing $\vec{\omega}$ sufficiently small, we can ensure that the error is less than δ by choosing $|\hat{\kappa}(b, \vec{\omega})| \leq \delta$ using small enough $\vec{\omega}$, which completes the proof of Eq. (68).

- ¹I. Shavitt, "The history and evolution of configuration interaction," *Mol. Phys.* **94**, 3–17 (1998).
- ²I. Shavitt and R. J. Bartlett, *Many-body methods in chemistry and physics: MBPT and coupled-cluster theory* (Cambridge university press, 2009).
- ³R. Orús, "A practical introduction to tensor networks: Matrix product states and projected entangled pair states," *Ann. Phys.* **349**, 117–158 (2014).
- ⁴G. K.-L. Chan and S. Sharma, "The density matrix renormalization group in quantum chemistry," *Annu. Rev. Phys. Chem.* **62**, 465–481 (2011).
- ⁵G. Carleo and M. Troyer, "Solving the quantum many-body problem with artificial neural networks," *Science* **355**, 602–606 (2017).
- ⁶N. Le Roux and Y. Bengio, "Representational power of restricted boltzmann machines and deep belief networks," *Neural Comput.* **20**, 1631–1649 (2008).
- ⁷G. Montufar and N. Ay, "Refinements of universal approximation results for deep belief networks and restricted boltzmann machines," *Neural Comput.* **23**, 1306–1319 (2011).
- ⁸D.-L. Deng, X. Li, and S. D. Sarma, "Quantum entanglement in neural network states," *Phys. Rev. X* **7**, 021021 (2017).
- ⁹Z. Cai and J. Liu, "Approximating quantum many-body wave functions using artificial neural networks," *Phys. Rev. B* **97**, 035116 (2018).
- ¹⁰K. Choo, G. Carleo, N. Regnault, and T. Neupert, "Symmetries and many-body excitations with neural-network quantum states," *Phys. Rev. Lett.* **121**, 167204 (2018).
- ¹¹X. Liang, W.-Y. Liu, P.-Z. Lin, G.-C. Guo, Y.-S. Zhang, and L. He, "Solving frustrated quantum many-particle models with convolutional neural networks," *Phys. Rev. B* **98**, 104426 (2018).
- ¹²K. 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).
- ¹³M. Hibat-Allah, M. Ganahl, L. E. Hayward, R. G. Melko, and J. Carrasquilla, "Recurrent neural network wave functions," *Phys. Rev. Res.* **2**, 023358 (2020).
- ¹⁴M. Hibat-Allah, R. G. Melko, and J. Carrasquilla, "Supplementing recurrent neural network wave functions with symmetry and annealing to improve accuracy," *arXiv preprint arXiv:2207.14314* (2022).
- ¹⁵O. Sharir, Y. Levine, N. Wies, G. Carleo, and A. Shashua, "Deep autoregressive models for the efficient variational simulation of many-body quantum systems," *Phys. Rev. Lett.* **124**, 020503 (2020).
- ¹⁶T. D. Barrett, A. Malyshev, and A. Lvovsky, "Autoregressive neural-network wavefunctions for ab initio quantum chemistry," *Nat. Mach. Intell.* **4**, 351–358 (2022).
- ¹⁷D. Luo and B. K. Clark, "Backflow transformations via neural networks for quantum many-body wave functions," *Phys. Rev. Lett.* **122**, 226401 (2019).
- ¹⁸Z. Liu and B. K. Clark, "Unifying view of fermionic neural network quantum states: From neural network backflow to hidden fermion determinant states," *Phys. Rev. B* **110**, 115124 (2024).
- ¹⁹A.-J. Liu and B. K. Clark, "Neural network backflow for ab-initio quantum chemistry," *Phys. Rev. B* **110**, 115137 (2024).
- ²⁰A.-J. Liu and B. K. Clark, "Efficient optimization of neural network backflow for ab-initio quantum chemistry," *arXiv preprint arXiv:2502.18843* (2025).
- ²¹T. Westerhout, N. Astrakhantsev, K. S. Tikhonov, M. I. Katsnelson, and A. A. Bagrov, "Generalization properties of neural network approximations to frustrated magnet ground states," *Nat. Commun.* **11**, 1593 (2020).
- ²²M. Bukov, M. Schmitt, and M. Dupont, "Learning the ground state of a non-stoquastic quantum hamiltonian in a rugged neural network landscape," *SciPost Phys.* **10**, 147 (2021).
- ²³C.-Y. Park and M. J. Kastoryano, "Expressive power of complex-valued restricted boltzmann machines for solving nonstoquastic hamiltonians," *Phys. Rev. B* **106**, 134437 (2022).
- ²⁴L. L. Viteritti, F. Ferrari, and F. Becca, "Accuracy of restricted boltzmann machines for the one-dimensional $j=1-j=2$ heisenberg model," *SciPost Phys.* **12**, 166 (2022).
- ²⁵O. Sharir, A. Shashua, and G. Carleo, "Neural tensor contractions and the expressive power of deep neural quantum states," *Phys. Rev. B* **106**, 205136 (2022).
- ²⁶Y. Nomura, A. S. Darmawan, Y. Yamaji, and M. Imada, "Restricted boltzmann machine learning for solving strongly correlated quantum systems," *Phys. Rev. B* **96**, 205152 (2017).
- ²⁷P.-J. Yang, M. Sugiyama, K. Tsuda, and T. Yanai, "Artificial neural networks applied as molecular wave function solvers," *J. Chem. Theory Comput.* **16**, 3513–3529 (2020).
- ²⁸K. Choo, A. Mezzacapo, and G. Carleo, "Fermionic neural-network states for ab-initio electronic structure," *Nat. Commun.* **11**, 2368 (2020).
- ²⁹N. Yoshioka, W. Mizukami, and F. Nori, "Solving quasiparticle band spectra of real solids using neural-network quantum states," *Commun. Phys.* **4**, 106 (2021).
- ³⁰J. Robledo Moreno, G. Carleo, A. Georges, and J. Stokes, "Fermionic wave functions from neural-network constrained hidden states," *Proc. Natl. Acad. Sci. U.S.A.* **119**, e2122059119 (2022).
- ³¹Y. Wu, X. Xu, D. Poletti, Y. Fan, C. Guo, and H. Shang, "A real neural network state for quantum chemistry," *arXiv preprint arXiv:2301.03755* (2023).
- ³²J. Hermann, J. Spencer, K. Choo, A. Mezzacapo, W. M. C. Foulkes, D. Pfau, G. Carleo, and F. Noé, "Ab initio quantum chemistry with neural-network wavefunctions," *Na. Rev. Chem.* **7**, 692–709 (2023).
- ³³H. J. Changlani, J. M. Kinder, C. J. Umrigar, and G. K.-L. Chan, "Approximating strongly correlated wave functions with correlator product states," *Phys. Rev. B* **80**, 245116 (2009).

- ³⁴Z. Wu, B. Zhang, W.-H. Fang, and Z. Li, “Hybrid tensor network and neural network quantum states for quantum chemistry,” *J. Chem. Theory Comput.* (2025).
- ³⁵K.-I. Funahashi, “On the approximate realization of continuous mappings by neural networks,” *Neural Netw.* **2**, 183–192 (1989).
- ³⁶G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Math. Control Signals Syst.* **2**, 303–314 (1989).
- ³⁷K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Netw.* **2**, 359–366 (1989).
- ³⁸K. Hornik, “Approximation capabilities of multilayer feedforward networks,” *Neural Netw.* **4**, 251–257 (1991).
- ³⁹M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, “Multilayer feedforward networks with a nonpolynomial activation function can approximate any function,” *Neural Netw.* **6**, 861–867 (1993).
- ⁴⁰Y. Huang and J. E. Moore, “Neural network representation of tensor network and chiral states,” *Phys. Rev. Lett.* **127**, 170601 (2021).
- ⁴¹P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for activation functions,” arXiv preprint arXiv:1710.05941 (2017).
- ⁴²X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (JMLR Workshop and Conference Proceedings, 2011) pp. 315–323.
- ⁴³S. R. Clark, “Unifying neural-network quantum states and correlator product states via tensor networks,” *J. Phys. A: Math. Theor.* **51**, 135301 (2018).
- ⁴⁴F. Mezzacapo, N. Schuch, M. Boninsegni, and J. I. Cirac, “Ground-state properties of quantum many-body systems: entangled-plaquette states and variational monte carlo,” *New J. Phys.* **11**, 083026 (2009).
- ⁴⁵I. Glasser, N. Pancotti, M. August, I. D. Rodriguez, and J. I. Cirac, “Neural-network quantum states, string-bond states, and chiral topological states,” *Phys. Rev. X* **8**, 011006 (2018).
- ⁴⁶C.-T. Perng, “A note on gordan’s theorem,” *British Journal of Mathematics & Computer Science* **10**, 1–6 (2015).
- ⁴⁷W. L. McMillan, “Ground state of liquid he 4,” *Phys. Rev.* **138**, A442 (1965).
- ⁴⁸J. Gubernatis, N. Kawashima, and P. Werner, *Quantum Monte Carlo Methods* (Cambridge University Press, 2016).
- ⁴⁹F. Becca and S. Sorella, *Quantum Monte Carlo approaches for correlated systems* (Cambridge University Press, 2017).
- ⁵⁰Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature* **521**, 436–444 (2015).
- ⁵¹Y. Crama and P. L. Hammer, *Boolean functions: Theory, algorithms, and applications* (Cambridge University Press, 2011).
- ⁵²M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information* (Cambridge University Press, 2010).
- ⁵³S. P. Boyd and L. Vandenberghe, *Convex optimization* (Cambridge university press, 2004).