

Supplementary Information: Solving the many-electron Schrödinger Equation with a Transformer-based framework

Honghui Shang,^{1†*} Chu Guo,^{1†} Yangjun Wu,¹ Zhenyu Li,^{1,2} Jinlong Yang^{1,2*}

¹ State Key Laboratory of Precision and Intelligent Chemistry,
University of Science and Technology of China, Hefei 230026, China

² Hefei National Laboratory,
University of Science and Technology of China, Hefei 230088, China

[†]These authors contribute equally to this work.

*To whom correspondence should be addressed: shanghai.ustc@gmail.com; jlyang@ustc.edu.cn.

S1 Variational Monte Carlo algorithm

The variational Monte Carlo (VMC) method has been adopted to optimize QiankunNet in an unsupervised fashion. Using the neural network ansatz, the ground state can be written as $|\psi_{\vec{\theta}}\rangle$, where $\vec{\theta}$ are the parameters to be optimized, and then the energy of the system can be written as a function of $\vec{\theta}$:

$$\begin{aligned} E(\vec{\theta}) &= \frac{\langle \psi_{\vec{\theta}} | H | \psi_{\vec{\theta}} \rangle}{\langle \psi_{\vec{\theta}} | \psi_{\vec{\theta}} \rangle} = \frac{\sum_{\mathbf{x}, \mathbf{x}'} \langle \psi_{\vec{\theta}} | \mathbf{x} \rangle \langle \mathbf{x} | H | \mathbf{x}' \rangle \langle \mathbf{x}' | \psi_{\vec{\theta}} \rangle}{\sum_{\mathbf{y}} \langle \psi_{\vec{\theta}} | \mathbf{y} \rangle \langle \mathbf{y} | \psi_{\vec{\theta}} \rangle} \\ &= \frac{\sum_{\mathbf{x}} E_{loc}(\mathbf{x}) p_{\vec{\theta}}(\mathbf{x})}{\sum_{\mathbf{y}} p_{\vec{\theta}}(\mathbf{y})} = \mathbb{E}_p[E_{loc}(\mathbf{x})], \end{aligned} \quad (1)$$

where $\mathbf{x}$, $\mathbf{x}'$, $\mathbf{y}$ denote specific bitstrings. The local energy $E_{loc}(\mathbf{x})$ in the second line of Eq.(1) is defined as

$$E_{loc}(\mathbf{x}) = \sum_{\mathbf{x}'} H_{\mathbf{x}\mathbf{x}'} \psi_{\vec{\theta}}(\mathbf{x}') / \psi_{\vec{\theta}}(\mathbf{x}), \quad (2)$$

where we have written $H_{\mathbf{x}\mathbf{x}'} = \langle \mathbf{x} | H | \mathbf{x}' \rangle$ and $\psi_{\vec{\theta}}(\mathbf{x}) = \langle \mathbf{x} | \psi_{\vec{\theta}} \rangle$ as the probability amplitude of the wave function ansatz $|\psi_{\vec{\theta}}\rangle$ in basis $|\mathbf{x}\rangle$. We have also written $p_{\vec{\theta}}(\mathbf{x}) = |\psi_{\vec{\theta}}(\mathbf{x})|^2$. It is impossible to exactly evaluate Eq.(1) in general as there exists an exponential number of different bitstrings. However, one could approximately evaluate Eq.(1) by sampling from the probability distribution $p_{\vec{\theta}}(\mathbf{x})$ to obtain a set of N_s samples, denoted as $\{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^{N_s}\}$, and then averaging over them

$$\tilde{E}(\vec{\theta}) = \frac{1}{N_s} \sum_{i=1}^{N_s} E_{loc}(\mathbf{x}^i). \quad (3)$$

Here we have used $\tilde{E}(\vec{\theta})$ instead of $E(\vec{\theta})$ to stress that the former is only an approximation of the latter. To this end, we can see that as long as one can efficiently sample from $p_{\vec{\theta}}(\mathbf{x})$ (which is the case if we can evaluate $\psi_{\vec{\theta}}(\mathbf{x})$ efficiently for each $\mathbf{x}$), and efficiently find those nonzero entry $H_{\mathbf{x}\mathbf{x}'}$ together with $\mathbf{x}'$, then Eq.(1) can be efficiently evaluated. To speed up the calculation, it is often advantageous to use a gradient-based optimizer instead of gradient-free ones. With the obtained samples, one could compute the approximate gradient of Eq.(1) using automatic differentiation (I):

$$\nabla_{\vec{\theta}} \tilde{E} = 2\text{Re} \left(\mathbb{E}_p \left[(E_{loc}(\mathbf{x}) - \mathbb{E}_p [E_{loc}(\mathbf{x})]) \nabla_{\vec{\theta}} \ln (\Psi_{\vec{\theta}}^*(\mathbf{x})) \right] \right). \quad (4)$$

Similarly, $\nabla_{\vec{\theta}} \tilde{E}$ is used to approximate the exact gradient $\nabla_{\vec{\theta}} E$. After that, the parameters $\vec{\theta}$ can be updated based on $\nabla_{\vec{\theta}} \tilde{E}$ and the optimizer, finishing one iteration of the VMC algorithm. In this way, the iterative methods can then optimize the network parameters to yield the lowest energy expectation value.

S2 Quantum Chemistry Hamiltonians

The central task for *ab initio* quantum chemistry calculation is to solve the static Schrödinger equation $H|\Psi\rangle = E|\Psi\rangle$ to get the ground state $|\Psi\rangle$ and the ground-state energy E of the many-body interacting Hamiltonian:

$$\hat{H} = -\sum_{i=1}^N \frac{1}{2} \nabla_i^2 - \sum_{i=1}^N \sum_{A=1}^M \frac{Z_A}{|\mathbf{r}_i - \mathbf{R}_A|} + \sum_{i=1}^N \sum_{j>i}^N \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \quad (5)$$

where N , M denote the total number of electrons and nuclei, ∇_i is the single particle kinetic operator of the i -th electron, $\mathbf{r}_i$ indicates the electronic coordinates, $\mathbf{R}_A$ and Z_A indicate the coordinates and charges of the A -th nucleus in the molecule. The Hamiltonian in Eq.(5) is referred to as the first quantized form, which needs to be discretized into grid points in real space if it is solved directly. An alternative representation is the second quantized form, in which the wave function is expressed with a finite basis set,

$$H^e = \sum_{p,q} h_q^p \hat{a}_p^\dagger \hat{a}_q + \frac{1}{2} \sum_{p,q,r,s} g_{r,s}^{p,q} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_r \hat{a}_s, \quad (6)$$

with h_q^p and $g_{r,s}^{p,q}$ the one- and two-electron integrals, $\hat{a}_p^\dagger$ and $\hat{a}_q$ the creation and annihilation operators. For convenience of using arbitrary neural networks, the JW transformation (2) can be used to convert H^e into a many-spin Hamiltonian, which generally takes the form

$$H = \sum_{i=1}^{N_h} w_i P_i, \quad (7)$$

where each P_i is the tensor product of Pauli spin operators $\{I, X, Y, Z\}$ of length N , referred to as a Pauli string, and w_i is a real coefficient. N_h denotes the total number of Pauli strings. For quantum chemistry Hamiltonians, N_h often scales as $O(N^4)$, which means that for each input bitstring $\mathbf{x}$, there could exist $O(N^4)$ $\mathbf{x}'$ s with nonzero $H_{\mathbf{x}\mathbf{x}'}$.

S3 Technical Implementation Details of QiankunNet

QiankunNet enforces electron conservation during the autoregressive sampling process through a principled masking mechanism that maintains physical constraints throughout wavefunction generation. We implement two complementary approaches to electron conservation: total electron conservation and spin-resolved electron conservation. In the total electron conservation approach, we track the running count of electrons placed in the partially constructed quantum state and dynamically modify the sampling probabilities to ensure the final state contains exactly the desired total number of electrons. This is achieved by calculating, at each sampling step, how many more electrons must be placed in the remaining positions to reach the target count. We then adjust the sampling probabilities accordingly—forcing occupation (probability of 1) when all remaining positions must be filled to meet the target, preventing occupation (probability of 0) when no more electrons can be added without exceeding the target, and otherwise allowing the model’s predicted probabilities to determine the outcome. For more stringent physical constraints, we can also employ spin-resolved electron conservation, where we separately track spin-up (α) and spin-down (β) electrons. This approach ensures that both the total electron count and the spin projection (S_z) are conserved. By maintaining fixed numbers of spin-up and spin-down electrons, the total spin S_z is implicitly conserved, while other symmetries beyond electron number and S_z conservation are learned during the optimization process rather than being explicitly enforced.

The molecular orbitals in QiankunNet are sampled in order of energy, following the natural energetic hierarchy of molecular orbitals. This energy-based ordering is physically motivated, as lower-energy orbitals typically contribute more significantly to the ground state wavefunction and aligns with physical intuition about electron occupation patterns in quantum systems.

QiankunNet employs a decoder-based architecture with configurable hyperparameters op-

timized for quantum chemistry applications. For the amplitude component, we implemented decoder layers with an embedding dimension of $d_{\text{model}} = 32$. The specific architecture configurations, including the number of layers and attention heads employed for various molecular systems, are detailed in Table S1. A key advantage of QiankunNet is its parameter efficiency and scalability to larger systems. For the amplitude network, most parameters are system-size independent, with only the positional embedding layer scaling with the number of spin orbitals—a negligible contribution to the overall parameter count due to its linear scaling and small embedding dimension. In the phase network, only the first layer necessarily depends on the number of qubits (spin-orbitals), while the parameter count in all subsequent layers remains fixed regardless of system size. When comparing systems with different numbers of spin-orbitals, the difference in their total parameter counts arises primarily from these two components, as all other network parameters remain constant. This architectural design ensures that as the system size grows, the parameter count increases only marginally, making QiankunNet highly scalable to complex molecular systems while maintaining computational efficiency.

For parameter optimization, we utilized the AdamW optimizer in conjunction with a theoretically-motivated learning rate schedule: $\alpha_i = d_{\text{model}}^{-0.5} \times \min(i^{-0.5}, i \times S_{\text{warmup}}^{-1.5})$, where α_i represents the learning rate at the i -th training epoch, and $S_{\text{warmup}} = 4000$ denotes the number of warm-up steps. This schedule integrates both warm-up and decay phases—the former facilitating stable initialization of training dynamics and the latter ensuring proper convergence behavior. The model dimension scaling factor ($d_{\text{model}}^{-0.5}$) adjusts the learning rate proportionally to the network capacity, following established principles from transformer-based architectures.

Our empirical evaluations demonstrate that this combination of physically-informed architectural choices and optimization protocols effectively navigates the complex energy landscapes characteristic of quantum chemical systems, enabling QiankunNet to achieve chemical accuracy across a diverse range of molecular structures.

For large molecular systems, storing the Hamiltonian as a list of symbolic Pauli strings would require substantial memory and be computationally inefficient. We employ a compressed data structure where each Pauli string is represented by three components: two boolean tuples of length N (the first storing occurrences of X or Y gates for state modification, the second storing occurrences of Y or Z gates for coefficient calculation) and an integer for sign computation. Our improved compression scheme achieves approximately 40% memory reduction by storing only the unique boolean tuples that determine state couplings and reorganizing the remaining data accordingly. The coefficients are precomputed using the occurrence of Y operators and updated in-place. This compression not only reduces memory requirements but also improves computational efficiency by naturally grouping Pauli strings that couple to the same output state, allowing each unique output state to be calculated only once with coefficients summed over duplicates, as illustrated in Algo. 1.

S4 Pre-training with configuration interaction method

The configuration interaction method solves the Schrödinger equation for a fixed basis set by expanding the wavefunction in the following way:

$$|\Psi\rangle = c_0|\Phi_0\rangle + \sum_{ra} c_a^r |\Phi_a^r\rangle + \sum_{\substack{a<b \\ r<s}} c_{ab}^{rs} |\Phi_{ab}^{rs}\rangle + \sum_{\substack{a<b<c \\ r<s<t}} c_{abc}^{rst} |\Phi_{abc}^{rst}\rangle + \dots \quad (8)$$

where $|\Phi_0\rangle$ is the Hatree-Fock ground state, $|\Phi_a^r\rangle$ is a singly excited Slater determinant from occupied spin orbital a to unoccupied spin orbital r , and all the other terms correspond to multiple excitations; while $|\Psi\rangle$ could denote any excited state of a given system here the discussion will be focused on the ground-state. When all the possible excited determinants are included in Eq. (8) the approach is called full configuration interaction (FCI) and the solution of the Schrödinger equation becomes exact for a given basis set. Since the total number of determinants grows as $\binom{N_o}{N_e}$, the FCI approach becomes quickly unpractical for most applications. It is

Algorithm 1 Hamiltonian data structure compression

Require: pauliStrings, n_qubits**Ensure:** idxs, XY_buf, YZ_buf, coeffs_buf

```
1: N, K  $\leftarrow$  n_qubits, length(pauliStrings)
2: XY, YZ  $\leftarrow$  zeors(N, K), zeors(N, K)
3: for (i,(coeff,pauliStr)) in enumerate(pauliStrings) do
4:   Yocc  $\leftarrow$  0 // occurrence for Y
5:   /* Construct Pauli mat XY and YZ */
6:   for (k, ps) in pauliStr do
7:     if ps == "X" then
8:       XY[k, i]  $\leftarrow$  1
9:     else if ps == "Y" then
10:      XY[k, i]  $\leftarrow$  1, YZ[k, i]  $\leftarrow$  1
11:      Yocc  $\leftarrow$  Yocc + 1
12:     else
13:       YZ[k, i]  $\leftarrow$  1
14:     end if
15:   end for
16:   /* New coefficients: fuce const calculation */
17:   coeff  $\leftarrow$  real(coeff)  $\times$  real((-i)Yocc)
18:   /* Compressed Pauli mat XY using Dict */
19:   sid = _encode2id(XY[:, i])
20:   if not haskey(coeffs_dict, sid) then
21:     XY_dict[sid]  $\leftarrow$  XY[:, i]
22:   end if
23:   coeffs_dict[sid].append(coeff)
24:   YZ_dict[sid].append([YZ[:, i]])
25: end for
26: /* Compacting data into new continuous buffer */
27: for (i,(sid, pm)) in enumerate(YZ_dict) do
28:   l  $\leftarrow$  length(pm)
29:   YZ_buf[:, j : j + l]  $\leftarrow$  pm
30:   XY_buf[:, i]  $\leftarrow$  XY_dict[sid]
31:   coeffs_buf[j : j + l]  $\leftarrow$  coeffs_dict[sid]
32:   j  $\leftarrow$  j + l; idxs[i]  $\leftarrow$  j
33: end for
```

Algorithm 2 MCTS based sampling method for QiankunNet

Require: Root, unique samples N_s , number of orbitals N , number of electrons N_e , BFS depth D , width W

Ensure: N_s unique configurations set $\mathcal{S}$

```
1: Queue=[root],  $\mathcal{S} = \{\}$ 
2: /* Phase 1 Breadth-First Search for first D orbitals */
3: for  $i = 1$  to  $D$  do
4:   new_queue = []
5:   for each config in queue do
6:      $p \leftarrow \text{QiankunNet.get\_probability}(\text{sample})$ 
7:     for state in {00, 01, 10, 11} do
8:       if  $\text{electron\_count\_valid}(\text{config}+\text{state}, i, N_e)$  then
9:          $\text{weight} \leftarrow p[\text{state}]$ 
10:        new_queue.append((config+state,weight))
11:      end if
12:    end for
13:  end for
14:  queue  $\leftarrow \text{top\_k}(\text{new\_queue}, W)$  // Keep W highest weights
15: end for
16: /* Phase 2: Depth-First Search for remaining orbitals */
17: for each (config, weight) in queue do
18:   current  $\leftarrow \text{config}$ 
19:   for  $j \leftarrow D + 1$  to  $N$  do
20:      $p \leftarrow \text{QiankunNet.get\_probability}(\text{current})$ 
21:     // Sample with electron conservation constraint
22:     valids_states  $\leftarrow \text{get\_valid\_states}(\text{current}, j, N_e)$ 
23:     state  $\leftarrow \text{sample\_categorical}(p[\text{valid\_states}])$ 
24:     current  $\leftarrow \text{current} + \text{state}$ 
25:   end for
26:    $\mathcal{S} \leftarrow \mathcal{S} \cup \text{current}$ 
27: end for
```

System	N _o	Layers	Heads	Amplitude Params	Phase Params	Total Params
H ₂	4	4	4	51264	791041	842305
F ₂	20	4	4	51520	799233	850753
HCl	20	4	4	51520	799233	850753
LiH	12	4	4	51392	795137	846529
H ₂ O	14	4	4	51424	796161	847585
CH ₂	14	4	4	51424	796161	847585
O ₂	20	4	4	51520	799233	850753
BeH ₂	14	4	4	51424	796161	847585
H ₂ S	22	4	4	51552	800257	851809
NH ₃	16	4	4	51456	797185	848641
N ₂	20	4	4	51520	799233	850753
CH ₄	18	4	4	51488	798209	849697
C ₂	20	4	4	51520	799233	850753
LiF	20	4	4	51520	799233	850753
PH ₃	24	4	4	51584	801281	852865
LiCl	28	4	4	51648	803329	854977
Li ₂ O	30	4	4	51680	804353	856033
Fenton	52	4	4	52032	815617	867649

Table S1: Network configurations and parameter counts for different systems. Amplitude and phase parameters are separated, with total parameters reflecting their sum.

widely recognized that the wave function can often be accurately represented by a limited number of excited determinants. In this context, a pre-trained model is proposed to directly generate these important contributions. Encoding the electronic wave function into the quantum states of qubits requires the so-called Fermion-to-qubit mapping (2,3). And we have the Jordan-Wigner mapping (2) defined as:

$$\begin{aligned}
\hat{a}_j^\dagger &\rightarrow \hat{Q}_j^\dagger \hat{\sigma}_z \cdots \hat{\sigma}_z \\
\hat{a}_j &\rightarrow \hat{Q}_j \hat{\sigma}_z \cdots \hat{\sigma}_z \\
\hat{Q}^\dagger &= \frac{(\hat{\sigma}_x - i\hat{\sigma}_y)}{2} \\
\hat{Q} &= \frac{(\hat{\sigma}_x + i\hat{\sigma}_y)}{2},
\end{aligned} \tag{9}$$

Under this mapping strategy, spin orbitals find their counterparts in qubits, and the occupation of orbitals is directly mapped to the $|0\rangle$ or $|1\rangle$ states of qubits. Consequently, Eq. (8) can also

be regarded as the quantum state of qubits, where the coefficients of computational basis can be simply calculated by transforming the Fermion excitation operators using Eq. (9). Such an approach enables the employment of a pre-converged wave function, derived from low-level configuration interaction (CI)-based methods, as an initial guess for the model. This subsequently leads to significant enhancements in computational efficiency.

S5 Comparison with leading NNQS methods

To put the NNQS method into a broader perspective we provide a comparison of methods in Table S2. Generally, they can be classified into two categories: first and second quantization methods. The computational complexity of first quantization methods exhibits a scaling behavior of $O(N_{sr}N_e^3)$, where N_{sr} represents the number of samples used to get the electron coordinates in real space, and N_e^3 pertains to the evaluation of determinants. In second quantization methods, the scaling is either $O(N_sN_o^4)$ or more accurately $O(N_uN_o^4)$, where N_o^4 accounts for the computation involved in assessing the local energy. N_s and N_u are number of samples and unique samples ($N_u \ll N_s$). Although first quantization methods has a lower computational scaling in calculating the local energy per sample, second quantization methods have the advantage of sampling efficiency and thus a higher overall performance for larger systems. Working in continuum space, first quantization methods cannot be adopted with the highly efficient AS method. Instead, only the less efficient MCMC method can be used. Therefore, QiankunNet emerges as a potentially effective methodology for quantum chemistry, that can be scaled to much larger systems while maintaining a high degree of accuracy.

S5.1 Comparison Between QiankunNet and PsiFormer

QiankunNet and PsiFormer share a fundamental similarity in their application of attention mechanisms to characterize electron-electron interactions, which has proven crucial for achiev-

ing high accuracy in quantum chemistry calculations. However, these approaches differ significantly in their implementation and technical foundations.

PsiFormer operates within a first-quantized formalism, employing neural networks to encode real-space wavefunctions in continuous space with complete variational treatment of basis sets. This approach has demonstrated impressive absolute energies for various systems. In contrast, QiankunNet utilizes a second-quantized formalism with a discrete basis set, which offers distinct advantages for enforcing boundary conditions and fermionic symmetry while facilitating the incorporation of explicit physical knowledge from configuration interaction theory. By implementing active space methods, QiankunNet can focus computational resources on chemically relevant orbitals and directly apply physical and chemical insights, which proves particularly advantageous when dealing with complex electronic structures and weak interactions.

The architectural differences between these approaches are equally significant. PsiFormer employs an encoder-style neural network that processes the complete input sequence simultaneously, lacking autoregressive properties. This design necessitates reliance on Markov Chain Monte Carlo (MCMC) sampling methods, which face challenges when scaling to larger systems due to sample rejection and correlation issues. QiankunNet, however, utilizes a decoder-based architecture that generates output tokens sequentially in a left-to-right manner, with previously generated tokens serving as context. This autoregressive property enables the implementation of autoregressive sampling (AS), which generates independent samples directly without rejection steps.

QiankunNet’s distinctive combination of a decoder-only transformer architecture with Monte Carlo tree search and batched autoregressive sampling creates a system that learns during both sampling and inference phases. The network enhances efficiency through parallel batch processing, making it particularly well-suited for GPU acceleration and large-scale applications.

This design philosophy complements the pioneering work of PsiFormer while offering a different approach to addressing the computational challenges of quantum chemistry calculations.

S5.2 Comparison Between QiankunNet and NAQS

QiankunNet and NAQS (Neural Autoregressive Quantum States) share fundamental similarities as both employ second-quantized formalism and autoregressive neural networks for quantum state representation. Both methods also utilize Autoregressive Sampling (AS) for efficient sampling and share similar theoretical scaling behavior with respect to system size.

From a computational complexity perspective, both approaches exhibit comparable scaling characteristics. At the k -th step of autoregressive sampling, the computational cost scales as $O(k^2)$ due to either the attention mechanism in transformers (QiankunNet) or the growing input dimension in MLPs (NAQS). For sampling a complete state with N orbitals, the total computational cost can be expressed as $\sum_{k=1}^N k^2 = \frac{N(N+1)(2N+1)}{6} \approx O(\frac{N^3}{3})$. For generating N_u unique samples, both approaches have a sampling complexity of $O(N_u N^3/3)$. In the local energy calculation step, since the Hamiltonian scales as $O(N^4)$ with the number of orbitals, the computational cost for energy evaluation is $O(N_u N^4)$ for both methods.

Despite these similarities, QiankunNet offers several significant advantages over NAQS. A fundamental architectural difference lies in how the two methods handle sampling: QiankunNet processes samples through the entire transformer-based amplitude network, while NAQS utilizes a specialized subnetwork for sampling. This architectural distinction allows QiankunNet to leverage the full representational power of its transformer model during the sampling process, contributing to both accuracy and efficiency. NAQS employs a multilevel perceptron (MLP) with hard-coded pre- and postprocessing to ensure the autoregressive property, and the size of its neural network grows with the number of qubits. This architecture results in a significant escalation in computational requirements and memory utilization as the system size increases.

For instance, when modeling the $\text{C}_2\text{H}_4\text{O}$ molecule comprising 38 qubits, NAQS requires a staggering 454 GB of GPU VRAM, exceeding the memory limits of even high-end GPUs like the A100 (80GB). In contrast, QiankunNet’s GPU VRAM requirement does not explicitly depend on the number of qubits, requiring only 14 GB for the same system. This memory efficiency translates to practical advantages in computational speed, particularly for larger systems. While NAQS calculations are faster for small systems with fewer than 24 qubits, QiankunNet demonstrates superior performance for larger molecules, achieving computational speeds up to five times faster than NAQS for a 28-qubit system like LiCl .

QiankunNet’s implementation of Autoregressive Sampling further enhances its computational efficiency. QiankunNet assuming that there are $N_{u,k}$ unique samples generated at the k -th step (corresponding to the width of the k -th layer in Fig. 3(b)), the total computational cost becomes $\sum_{k=1}^N k^2 N_{u,k}$. When $N_{u,k}$ remains approximately constant as N_u (typically less than 10^6 even for molecular systems with $N > 100$), the cost of sampling reduces to $O(N_u N^3/3)$, which is independent of the total number of samples N_s . This independence from the total sample count represents a significant practical advantage, allowing QiankunNet to efficiently generate large numbers of samples with minimal computational overhead. Additionally, our implementation leverages KV-cache technology and GPU-accelerated batch processing, further enhancing performance while maintaining the same theoretical scaling behavior.

Beyond computational efficiency, QiankunNet’s transformer-based architecture also delivers superior accuracy compared to NAQS. For instance, in the Li_2O case, while NAQS captures 98.1% of the electron correlation energy, QiankunNet recovers an impressive 99.9%. This enhanced accuracy stems from the attention mechanism’s ability to capture complex electron-electron interactions and long-range dependencies in the wavefunction, demonstrating that QiankunNet not only offers practical computational advantages but also achieves higher precision in quantum chemistry calculations.

S5.3 Comparison Between QiankunNet and TQS

We note the related work by Zhang et al. (9), who developed a decoder-only transformer (TQS) to model wavefunctions in second quantization. While both QiankunNet and TQS employ autoregressive sampling, they differ in several key aspects. In sampling implementation, QiankunNet utilizes an efficient batch processing mechanism with counts tracking, whereas TQS tracks each sample independently. In network architecture, QiankunNet employs separate amplitude and phase networks that learn distinct feature representations, while TQS uses a shared backbone network with independent output heads. Additionally, QiankunNet incorporates GPU optimizations and KV-cache technology that significantly accelerates autoregressive sampling by storing previously computed matrices. These architectural and implementation differences contribute to QiankunNet’s superior computational efficiency and accuracy in capturing quantum correlations, as demonstrated in our comparative benchmarks.

For the benchmark tests, we employed the Transverse Field Ising (TFI) model, a paradigmatic quantum spin system described by the Hamiltonian $H = -J \sum_{i=1}^{n-1} \sigma_i^z \sigma_{i+1}^z - h \sum_{i=1}^n \sigma_i^x$, where σ_i^z and σ_i^x are Pauli matrices acting on site i , J represents the strength of nearest-neighbor interactions, and h denotes the transverse field strength. This model serves as an ideal benchmark test due to its well-understood physics and exact solvability in certain regimes, while still capturing essential quantum many-body phenomena including quantum phase transitions. Our direct comparative experimental results as shown in Figure 3 clearly demonstrate these advantages: for a TFI system size of $n = 40$ without pretraining, we fix $J = 1$ and $h = 1$, TQS required 280 minutes to generate 2048 unique samples, while QiankunNet needed only 28.9 minutes to generate 1952 unique samples—reducing computation time by approximately 90%. Moreover, at the same number of iterations, the absolute error ($|E - E_{\text{dmrg}}|$) of NNQS is approximately one order of magnitude lower than that of TQS, and NNQS achieves in about 1000 iterations a level of precision that TQS struggles to reach even after 5500 iterations. In this

application, our method demonstrates superior accuracy and computational efficiency.

S6 The effect of parameters and pre-training

The effect of the number of sample size is systematically investigated. For H_2O , the chemical accuracy is reached only when the sampling numbers (N_s) is larger than 10^4 (with $N_u=244$) (Fig. S2a). Regarding the 30-qubit Li_2O system, an even larger sample size of 10^9 (with $N_u=46755$) is required to achieve the chemical accuracy. Because the variance of the local energy increases with the system size, so the number of MC samples must increase to reduce the error to the same tolerance. Although the scaling of how the sample size increases with the system size at a specific accuracy is unknown, the sample size is expected to be significantly smaller than the number of Slater determinants (N_f). At the same time, the computational cost only scales with the number of unique samples (N_u), which is generally much smaller compared with the number of the sample size ($N_u \ll N_s \ll N_f$).

We use H_2O as an example to demonstrate the power of the CISD pre-training model. As shown in Fig. S2(b), without pretraining, 3,000 iterations are required to converge to the chemical accuracy. However, by employing the CISD pre-train model, we manage to achieve the convergence to the chemical accuracy in less than 1,000 iterations. Such a benefit is even more pronounced for the Li_2O system, where we observe a fivefold increase in the convergence speed.

S7 Discussions about Fenton reaction systems

To validate the accuracy of QiankunNet for the Fenton reaction calculation and address potential concerns about error sources, we performed additional CASCI calculations using the same (28e, 26o) active space. As shown in Table S3, the differences between CASCI and QiankunNet energies remain consistently within chemical accuracy (< 1.6 mHa) across the entire reaction coordinate. This excellent agreement confirms that QiankunNet accurately reproduces the elec-

tronic structure within the chosen active space. The around 30 mHa difference between our results and prior literature (Sayfutyarova et al. 2017) values can therefore be attributed to our use of a larger active space (28e, 26o) compared to the smaller active spaces employed in previous studies. The apparent roughness of our energy profile compared to literature is simply due to our sparser sampling along the IRC (10 points versus finer grids in other works), not computational noise or accuracy issues.

References

1. T. D. Barrett, A. Malyshev, A. Lvovsky, Autoregressive neural-network wavefunctions for ab initio quantum chemistry. *Nat. Mach. Intelle.* **4**, 351 (2022).
2. P. Jordan, E. Wigner, Über das Paulische Äquivalenzverbot. *Zeitschrift für Physik* **47**, 631 (1928).
3. S. B. Bravyi, A. Y. Kitaev, Fermionic quantum computation. *Annals of Physics* **298**, 210 (2002).
4. D. Pfau, J. S. Spencer, A. G. D. G. Matthews, W. M. C. Foulkes, Ab initio solution of the many-electron schrödinger equation with deep neural networks. *Phys. Rev. Res.* **2**, 033429 (2020).
5. J. Hermann, Z. Schätzle, F. Noé, Deep-neural-network solution of the electronic schrödinger equation. *Nat. Chem.* **12**, 891 (2020).
6. I. von Glehn, J. S. Spencer, D. Pfau, A self-attention ansatz for ab-initio quantum chemistry. *arXiv:2211.13672* (2022).
7. K. Choo, A. Mezzacapo, G. Carleo, Fermionic neural-network states for ab-initio electronic structure. *Nat. Commun.* **11**, 2368 (2020).

8. T. Zhao, J. Stokes, S. Veerapaneni, Scalable neural quantum states architecture for quantum chemistry. *Machine Learning: Science and Technology* (2023).
9. Y.-H. Zhang, M. Di Ventura, Transformer quantum state: A multipurpose model for quantum many-body problems. *Phys. Rev. B* **107**, 075147 (2023).
10. M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, M. Troyer, Elucidating reaction mechanisms on quantum computers. *Proc. Natl. Acad. Sci. U.S.A.* **114**, 7555 (2017).
11. Z. Li, J. Li, N. S. Dattani, C. Umrigar, G. K.-L. Chan, The electronic complexity of the ground-state of the fmo cofactor of nitrogenase as relevant to quantum simulations. *J. Chem. Phys.* **150**, 024302 (2019).

Table S2: A comparison of leading NNQS methods. MLP means multilevel perceptron neural-network, RBM means restricted Boltzmann machines neural-network, MADE means Masked Autoencoder for Distribution Estimation. MCMC refers to Markov Chain Monte Carlo sampling method, AS refers to bathed autoregressive sampling method. MRHF refers to multi-reference HF method, CISD means configuration interaction methods. N_e represents the total number of electrons, N_o represents the total number of spin orbitals, N_{sr} indicates the number of sampled electron coordinates in real space, N_s indicates the number of samples in second quantization, N_u indicates the number of unique samples.

Method	First Quantization			Second Quantization			
Capability / Model	FermiNet	PauliNet	Psiformer	NC	NAQS	MADE	QiankunNet
References	(4)	(5)	(6)	(7)	(1)	(8)	This work
Neural Network	MLP	graph DNN and MLP	encoder-only Transformer	RBM	MLP	MADE	decoder-only Transformer
Self-Attention	NO	NO	YES	NO	NO	NO	YES
Autoregressive Network	NO	NO	NO	NO	YES	YES	YES
Sampling Method	MCMC	MCMC	MCMC	MCMC	AS	AS	AS
Physical Knowledge		encode MRHF					encode CI
Scaling	$O(N_{sr}N_e^3)$	$O(N_{sr}N_e^3)$	$O(N_{sr}N_e^3)$	$O(N_sN_o^4)$	$O(N_uN_o^4)$	$O(N_uN_o^4)$	$O(N_uN_o^4)$

Table S3: Comparison of CASCI and QiankunNet energies along the Fenton reaction IRC.

Order	IRC	CASCI (Ha)	QiankunNet (Ha)	Error (mHa)
1	0	-1802.5112	-1802.511067	0.15
2	2.371955	-1802.5103	-1802.510193	0.10
3	4.287537	-1802.5058	-1802.505676	0.17
4	6.366722	-1802.4817	-1802.481335	0.37
5	8.168359	-1802.4616	-1802.460962	0.67
6	10.176215	-1802.4426	-1802.441195	1.44
7	11.178753	-1802.4056	-1802.405322	0.27
8	14.192211	-1802.3796	-1802.378058	1.57
9	16.188559	-1802.3823	-1802.381089	1.16
10	17.283115	-1802.3816	-1802.380164	1.42

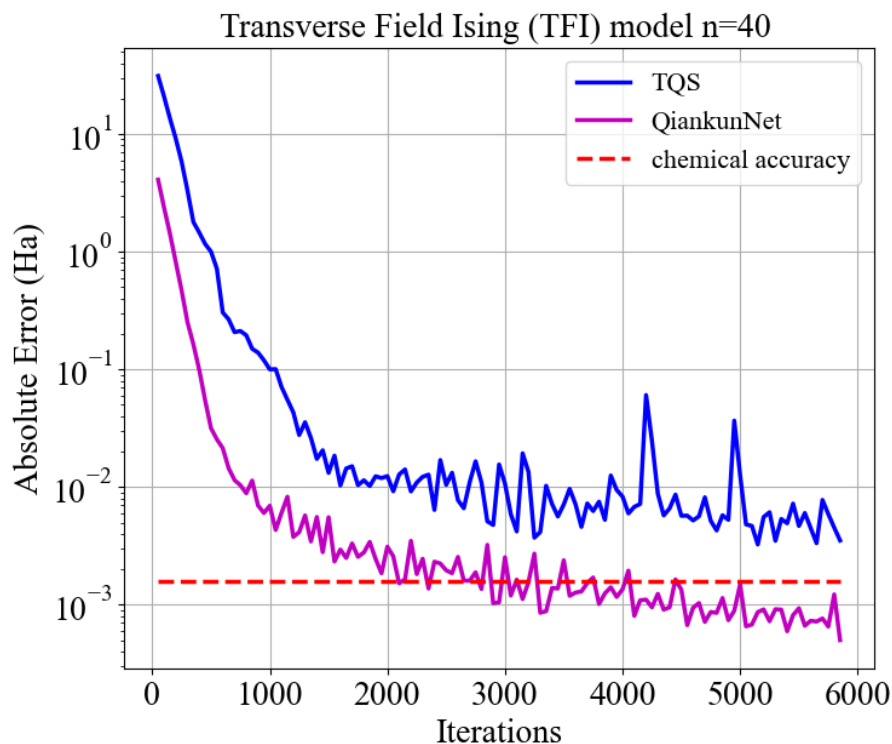


Figure S1: Convergence comparison between QiankunNet and TQS (9) for system sizes $n = 40$. The vertical axis shows absolute energy error ($|E - E_{\text{dmrg}}|$) on a logarithmic scale, while the horizontal axis shows iterations. The red dashed line indicates chemical accuracy ($\sim 10^{-3}$ Hartree). This graph compares two methods without pretraining: QiankunNet with 1952 unique samples (blue line, 28.9 minutes runtime) and TQS with 2048 unique samples (magenta line, 280 minutes runtime). QiankunNet demonstrates significantly faster convergence and lower final error, reaching chemical accuracy in fewer iterations while requiring approximately $10\times$ less computational time. Source data for the results are provided as a Source Data file.

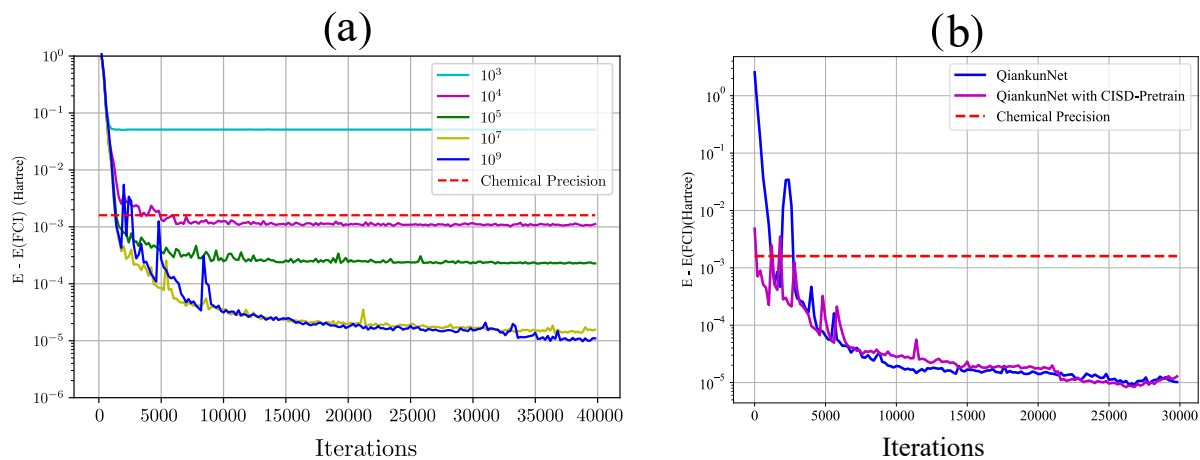


Figure S2: **Energy estimation as a function of the optimization step for the H_2O molecule** (a) with different sample sizes.(b) with or without CISD pre-training.