Physics-inspired transformer quantum states via latent imaginary-time evolution

Kimihiro Yamazaki [1,2,*] Itsushi Sakata [3] Takuya Konishi [1,3] and Yoshinobu Kawahara [1,3]
[1]*Graduate School of Information Science and Technology, The University of Osaka, 1-5 Yamadaoka, Suita, Osaka, Japan*
[2]*Fujitsu Research, Fujitsu Limited, 4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki, Kanagawa, Japan*
[3]*Center for Advanced Intelligence Project, RIKEN, 1-4-1 Nihonbashi, Chuo-ku, Tokyo, Japan*

(Received 4 February 2026; accepted 10 June 2026; published 9 July 2026)

Neural quantum states (NQS) are powerful ansätze in the variational Monte Carlo framework, yet their architectures are often treated as black boxes. We propose a physically transparent framework in which NQS are treated as neural approximations to latent imaginary-time evolution. This viewpoint suggests that standard transformer-based NQS (TQS) architectures correspond to physically unmotivated effective Hamiltonians dependent on imaginary time in a latent space. Building on this interpretation, we introduce physics-inspired transformer quantum states, which enforce a static effective Hamiltonian by sharing weights across layers and improve propagation accuracy via Trotter-Suzuki decompositions without increasing the number of variational parameters. For the frustrated J_1-J_2 Heisenberg model, our ansätze achieve accuracies comparable to or exceeding state-of-the-art TQS while using substantially fewer variational parameters. This study demonstrates that reinterpreting the deep network structure as a latent cooling process enables a more physically grounded, systematic, and compact design, thereby bridging the gap between black-box expressivity and physically transparent construction.

DOI: 10.1103/bjxb-8tsk

I. INTRODUCTION

Accurately obtaining ground-state wave functions of quantum many-body systems, especially strongly correlated systems [1], is a grand challenge in modern physics because of the complexity of many-body correlations. In recent years, neural quantum states (NQS) [2] have emerged as a powerful class of variational ansätze within variational Monte Carlo (VMC) [3]. In particular, transformer-based NQS (TQS) [4–7] have achieved state-of-the-art accuracy on some of the most challenging benchmark problems, notably the frustrated J_1-J_2 Heisenberg model [8].

Despite these successes, existing NQS architectures, particularly TQS [4–7], largely repurpose generic machine-learning designs and thus remain black-box ansätze. As a result, improved accuracy is often achieved through substantial overparametrization, while providing limited physical insight into how and why the ansatz attains its performance. This black-box nature also makes systematic architectural refinement difficult, since there is no unified principle linking architectural changes to controlled improvements in accuracy.

In parallel to the development of variational ansätze, a long-standing and physically transparent route to ground states is provided by imaginary-time evolution (ITE). The fundamental principle of ITE relies on cooling an arbitrary initial state $|\Psi_0\rangle$ toward the ground state $|\Psi\rangle$ as

$$|\Psi\rangle \propto e^{-\beta\hat{\mathcal{H}}}|\Psi_0\rangle, \tag{1}$$

where $\hat{\mathcal{H}}$ is the Hamiltonian and β is a sufficiently large imaginary time [9]. The auxiliary-field quantum Monte Carlo (AF-QMC) method [10,11] realizes ITE by employing the Hubbard-Stratonovich transformation. Although this approach yields exact estimates for specific Hamiltonians, the explicit stochastic sampling suffers from a severe sign problem, particularly in frustrated spin and fermion systems, which strictly limits its applicability. Even recent deep Boltzmann machine constructions of exact ITE [12] essentially inherit the sign problem in frustrated systems. Thus, a robust, sign-problem-free integration of NQS and ITE for generic frustrated settings remains a significant challenge.

In this work, we bridge the gap between black-box NQS and physically transparent construction by introducing a framework based on latent imaginary-time evolution (LITE). We represent the NQS ansatz through a cooling process within a latent space, inspired by the imaginary-time picture in Eq. (1) within the VMC scheme. In this scheme, instead of directly using the bare physical Hamiltonian $\hat{\mathcal{H}}$, we model the LITE with a learnable effective Hamiltonian in the latent space, which is parametrized by a neural network. Crucially, this variational formulation makes our approach inherently free from the sign problem.

From the perspective of LITE, it becomes clear that existing TQS architectures [4–7] effectively imply that the cooling process is driven by an effective Hamiltonian with imaginary-time dependence. This variation at every evolution

*Contact author: k-yamazaki@ist.osaka-u.ac.jp

step represents an unphysical redundancy for ground-state cooling and leads to significant overparametrization. To address this, we propose physics-inspired transformer quantum states (PITQS). Since the cooling process is naturally governed by a single Hamiltonian independent of imaginary time, we enforce a static effective Hamiltonian by sharing weights across all layers.

This framework provides immediate physical insight into the network architecture and highlights a fundamental limitation of existing TQS. In particular, once each layer of the TQS is identified as an approximation to the latent imaginary-time propagator generated by noncommuting terms in the effective Hamiltonian, the standard TQS update is naturally interpreted as a first-order Lie-Trotter decomposition [13,14]. As in any first-order decomposition, finite per-layer imaginary-time steps induce systematic Trotter errors. Motivated by this observation, we systematically implement higher-order Trotter-Suzuki decompositions [15–17] of the static effective Hamiltonian, thereby improving the propagator accuracy without increasing the number of variational parameters.

We demonstrate the capabilities of our framework on the frustrated J_1-J_2 Heisenberg model, which serves as an established benchmark in the NQS domain [8,18]. Our results show that a static effective Hamiltonian is sufficiently expressive to capture intricate quantum correlations, indicating that much of the parameter budget in conventional TQS, which implicitly employ effective Hamiltonians with imaginary-time dependence, is redundant. Guided by this physical perspective, we explicitly design both the effective Hamiltonian and the Trotter-Suzuki scheme, achieving accuracies that are comparable to, and in some cases exceed, state-of-the-art benchmarks while using substantially fewer variational parameters. Overall, our approach elevates the design of the NQS architecture from an empirical heuristic to a systematic, physically grounded construction.

II. LATENT IMAGINARY-TIME EVOLUTION

Here, we formulate a framework to construct an NQS ansatz via LITE and optimize it within the VMC scheme. Working in a discrete configuration basis $\{|n\rangle\}$ (for an N spin-1/2 system, n labels spin configurations of the Hilbert space), we parametrize $\Psi_\theta(n) = \langle n|\Psi_\theta\rangle$ by multiple deep neural networks with variational parameters θ, composed of three logical blocks: a latent encoder $\hat{\mathcal{E}}_\theta$, a LITE operator $\hat{\mathcal{U}}_\theta(\beta)$, and a wave-function decoder $\hat{\mathcal{D}}_\theta$. The total variational ansatz is defined as the composition

$$\log \Psi_\theta(n) \;=\; \hat{\mathcal{D}}_\theta[\hat{\mathcal{U}}_\theta(\beta)[\hat{\mathcal{E}}_\theta[n]]] \in \mathbb{C}, \tag{2}$$

Here and throughout, log denotes the natural logarithm. Where β denotes the total imaginary time in the latent space [Fig. 1(a)].

First, the latent encoder operator $\hat{\mathcal{E}}_\theta$ maps a discrete configuration n to a latent state $z(0) \in \mathbb{R}^{N_{\text{tok}}\times d}$. Here, $z(0)$ is represented by N_{tok} latent tokens $z_i(0) \in \mathbb{R}^d$, where i is the latent-token index and d is the token dimension. Introducing latent tokens is analogous in spirit to introducing auxiliary fields in AF-QMC [10,11]; it provides extra degrees of freedom through which nontrivial correlations can be represented efficiently. However, unlike AF-QMC, our latent variables

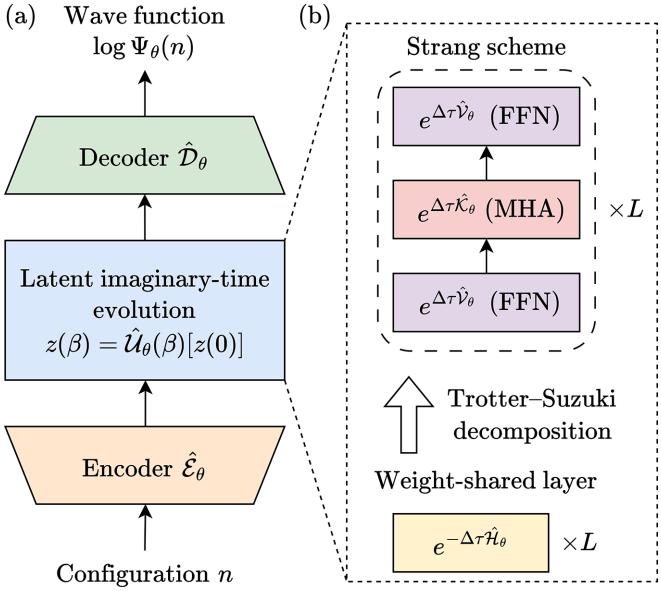

FIG. 1. PITQS overview. (a) A configuration n is mapped by the encoder $\hat{\mathcal{E}}_\theta$ to initial latent states $z(0)$, evolved by the LITE operator $\hat{\mathcal{U}}_\theta(\beta)$ to $z(\beta)$, and decoded by $\hat{\mathcal{D}}_\theta$ to the wave-function log-amplitude $\log \Psi_\theta(n)$. (b) In PITQS, $\hat{\mathcal{U}}_\theta(\beta)$ is implemented as L weight-shared layers approximating $e^{-\Delta\tau\hat{\mathcal{H}}_\theta}$ with a single effective Hamiltonian $\hat{\mathcal{H}}_\theta$; each step uses a Trotter-Suzuki decomposition (Strang decomposition scheme shown) that alternates the nonlocal operator $\hat{\mathcal{K}}_\theta$ and the local operator $\hat{\mathcal{V}}_\theta$.

are learnable and optimized end to end within VMC, rather than being stochastically sampled and integrated out. This latent-space viewpoint thus provides a physically motivated way to enlarge the variational manifold while retaining the sign-problem-free nature of the VMC formulation.

Next, $\hat{\mathcal{U}}_\theta(\beta)$ performs the LITE cooling over a total imaginary time β and realizes a low-energy projection in the latent space. In our approach, the LITE operator $\hat{\mathcal{U}}_\theta(\beta)$ is implemented by a neural network, and we write

$$z(\beta) = \hat{\mathcal{U}}_\theta(\beta)[z(0)]. \tag{3}$$

It is convenient to interpret this action in terms of a latent state $z(\tau) \in \mathbb{R}^{N_{\text{tok}}\times d}$, $0 \leqslant \tau \leqslant \beta$, with $z(0) = \hat{\mathcal{E}}_\theta[n]$ and $z(\beta) = \hat{\mathcal{U}}_\theta(\beta)[z(0)]$. In the continuum limit, one may regard $z(\tau)$ as obeying the latent imaginary-time evolution equation $dz(\tau)/d\tau = -\hat{\mathcal{H}}_\theta[z(\tau)]$, whose solution yields

$$z(\beta) = e^{-\beta\hat{\mathcal{H}}_\theta}[z(0)]. \tag{4}$$

Here, $\hat{\mathcal{H}}_\theta$ denotes an effective Hamiltonian parametrized by a neural network in the latent space. Although the latent dynamics considered here is generally nonlinear and is, more precisely, a nonlinear flow implemented by neural networks on the latent state, we use the exponential notation as a formal shorthand to emphasize its close analogy to imaginary-time evolution. In this sense, Eq. (4) is directly analogous to the ITE principle in Eq. (1), except that the latent state evolves under the effective Hamiltonian $\hat{\mathcal{H}}_\theta$ rather than the physical Hamiltonian $\hat{\mathcal{H}}$.

Finally, the wave-function decoder $\hat{\mathcal{D}}_\theta$ maps the evolved latent state $z(\beta)$ to the wave-function log-amplitude $\log \Psi_\theta(n)$.

III. RELATION TO EXISTING TQS

We reexamine the existing TQS architecture [4–8] through the lens of LITE. In this viewpoint, existing TQS can be organized into an embedding that maps physical degrees of freedom to an initial latent representation, a wave-function decoder that maps the final latent representation to the variational many-body wave function, and a depthwise latent update implemented by repeated multihead attention (MHA) and feedforward network (FFN) transformations acting on latent tokens.

First, we identify our latent encoder and decoder. The embedding used in standard TQS [4,5,8] corresponds to the latent encoder $\hat{\mathcal{E}}_\theta$: It partitions each configuration n into local patches and maps them to $z(0) \in \mathbb{R}^{N_{\rm tok}\times d}$. For a two-dimensional $M \times M$ lattice with square patches of size $b \times b$, this yields $N_{\rm tok} = (M/b)^2$ tokens. On the decoder side, the standard TQS formulation [4,5,8] is as follows:

$$\hat{\mathcal{D}}_\theta[z(\beta)] = \sum_{a=1}^{d} \log\cosh\left(w_a^\top \zeta + b_a\right), \tag{5}$$

where $\zeta = \sum_{i=1}^{N_{\rm tok}} z_i(\beta) \in \mathbb{R}^d$ and $\{w_a, b_a\}_{a=1}^d$ are complex-valued variational parameters.

Next, focusing on the stacked TQS layers, we can interpret the updates driven by MHA and FFN as the LITE. We discretize Eq. (4) by partitioning $\tau \in [0,\ \beta]$ into L slices with $\Delta\tau = \beta/L$, introducing $\tau_\ell = \ell\Delta\tau$, and defining the discrete latent state $z^{(\ell)} \equiv z(\tau_\ell)$ ($\ell = 0, \ldots, L$), so that $z^{(0)} = z(0)$ and $z^{(L)} = z(\beta)$. Within this notation, the ℓth layer updates the latent state from $z^{(\ell-1)}$ to $z^{(\ell)}$ and can be viewed as a neural approximation to a short imaginary-time propagator acting in the latent space.

In general, a quantum many-body Hamiltonian $\hat{\mathcal{H}}$ can be written as the sum of a nonlocal term $\hat{\mathcal{K}}$ and a local (on-site) term $\hat{\mathcal{V}}$. Motivated by this structure, we interpret the two submaps inside a TQS layer as providing an analogous decomposition in the latent space: The MHA sublayer updates each latent token through an aggregation over all other tokens, thereby enabling long-range token-token couplings and playing the role of an effective nonlocal interaction term, whereas the FFN sublayer applies the same nonlinear map independently to each token and thus plays the role of an effective on-site term. Concretely, denoting the MHA and FFN residual updates in layer ℓ by the maps $\hat{\mathcal{K}}_\theta^{(\ell)}$ and $\hat{\mathcal{V}}_\theta^{(\ell)}$, respectively, we define the layer-ℓ effective Hamiltonian in the latent space as

$$\hat{\mathcal{H}}_\theta^{(\ell)} \equiv -\left(\hat{\mathcal{V}}_\theta^{(\ell)} + \hat{\mathcal{K}}_\theta^{(\ell)}\right), \tag{6}$$

where the minus sign is chosen so that the ITE cooling corresponds to applying the LITE operator. For implementation details of $\hat{\mathcal{K}}_\theta^{(\ell)}$ and $\hat{\mathcal{V}}_\theta^{(\ell)}$, see Appendix A. With this definition, $\hat{\mathcal{H}}_\theta^{(\ell)}$ has the typical structure of a quantum many-body Hamiltonian as a sum of a local term and a nonlocal term, making it clear that the TQS naturally induces an effective Hamiltonian in the latent space in a form well suited to representing quantum many-body systems.

Each layer of the existing TQS applies MHA and FFN sequentially, which is naturally interpreted as a first-order Lie-Trotter decomposition [13,14] of the short imaginary-time propagator; for a single step at layer ℓ,

$$e^{-\Delta\tau\hat{\mathcal{H}}_\theta^{(\ell)}} = e^{\Delta\tau(\hat{\mathcal{V}}_\theta^{(\ell)}+\hat{\mathcal{K}}_\theta^{(\ell)})} \approx e^{\Delta\tau\hat{\mathcal{V}}_\theta^{(\ell)}} e^{\Delta\tau\hat{\mathcal{K}}_\theta^{(\ell)}}. \tag{7}$$

Because $[\hat{\mathcal{K}}_\theta^{(\ell)}, \hat{\mathcal{V}}_\theta^{(\ell)}] \neq 0$, this decomposition incurs Trotter errors scaling locally as $\mathcal{O}(\Delta\tau^2)$. Approximating each component by the Euler-type update yields the concrete layer update

$$\begin{aligned} \tilde{z}^{(\ell)} &= e^{\Delta\tau\hat{\mathcal{K}}_\theta^{(\ell)}}[z^{(\ell-1)}] \approx z^{(\ell-1)} + \Delta\tau\,\hat{\mathcal{K}}_\theta^{(\ell)}[z^{(\ell-1)}], \\ z^{(\ell)} &= e^{\Delta\tau\hat{\mathcal{V}}_\theta^{(\ell)}}[\tilde{z}^{(\ell)}] \approx \tilde{z}^{(\ell)} + \Delta\tau\,\hat{\mathcal{V}}_\theta^{(\ell)}[\tilde{z}^{(\ell)}], \end{aligned} \tag{8}$$

where $\tilde{z}^{(\ell)}$ denotes an intermediate latent state after applying the nonlocal update within the ℓth layer. This matches the standard TQS layer ordering (MHA → FFN) within each layer. Finally, stacking L layers implements the LITE operator as an ordered product of short imaginary-time propagators,

$$\hat{\mathcal{U}}_\theta(\beta) \approx T_\tau \prod_{\ell=1}^{L} e^{\Delta\tau\hat{\mathcal{V}}_\theta^{(\ell)}} e^{\Delta\tau\hat{\mathcal{K}}_\theta^{(\ell)}}, \tag{9}$$

where T_τ denotes the imaginary-time-ordered product in the latent space.

Importantly, in conventional TQS architectures [4,5,8], the weights are not shared across layers, so the induced latent-space operators vary with ℓ: For $\ell \neq \ell'$, one generally has $\hat{\mathcal{H}}_\theta^{(\ell)} \neq \hat{\mathcal{H}}_\theta^{(\ell')}$. Equivalently, the standard depth-L TQS realizes effective Hamiltonians dependent on latent imaginary time. However, from a physical standpoint, the VMC target problem is specified by a single, imaginary-time independent Hamiltonian $\hat{\mathcal{H}}$, and ITE cooling is an autonomous process. Therefore, once the stacked TQS layers are interpreted as implementing LITE, the layerwise variation of $\hat{\mathcal{H}}_\theta^{(\ell)}$ constitutes a physically redundant freedom, leading to substantial overparametrization: The number of variational parameters scales linearly with L even though the underlying mechanism is governed by a single Hamiltonian.

IV. PHYSICS-INSPIRED TRANSFORMER QUANTUM STATES

Guided by this viewpoint, we introduce PITQS, a physics-inspired construction of $\hat{\mathcal{U}}_\theta(\beta)$ that enforces a single effective Hamiltonian independent of imaginary time and systematically improves the propagation accuracy by higher-order Trotter-Suzuki decompositions [Fig. 1(b)]. Concretely, instead of the layer-dependent effective Hamiltonians $\{\hat{\mathcal{H}}_\theta^{(\ell)}\}_{\ell=1}^L$ implicit in standard TQS, we impose a single weight-shared effective Hamiltonian

$$\hat{\mathcal{H}}_\theta = -(\hat{\mathcal{V}}_\theta + \hat{\mathcal{K}}_\theta) \tag{10}$$

and approximate the LITE operator

$$\hat{\mathcal{U}}_\theta(\beta) = (e^{-\Delta\tau\,\hat{\mathcal{H}}_\theta})^L = e^{-\beta\hat{\mathcal{H}}_\theta}. \tag{11}$$

This corresponds to realizing the ITE [Eq. (1)] in the latent space. Crucially, the weight-sharing constraint across layers removes the physically redundant layerwise variation present in standard TQS [Eq. (9)] and reduces the number of variational parameters by a factor of order L.

Furthermore, the LITE viewpoint suggests a systematic route beyond the first-order Lie-Trotter decomposition. We replace the single-step propagator with an mth order Trotter-Suzuki decomposition,

$$e^{-\Delta\tau\hat{\mathcal{H}}_\theta} \approx \prod_{i=1}^{k} \exp\left(a_i^{(m)}\Delta\tau\,\hat{\mathcal{V}}_\theta\right)\exp\left(b_i^{(m)}\Delta\tau\,\hat{\mathcal{K}}_\theta\right), \tag{12}$$

where k is the number of stages and $\{a_i^{(m)}, b_i^{(m)}\}$ are fixed coefficients defining an mth order decomposition with local Trotter error $\mathcal{O}(\Delta\tau^{m+1})$ and satisfying $\sum_i a_i^{(m)} = \sum_i b_i^{(m)} = 1$. In the present nonlinear setting, this construction is more precisely a general splitting scheme [19] for the nonlinear flow generated by $\hat{\mathcal{H}}_\theta = -(\hat{\mathcal{V}}_\theta + \hat{\mathcal{K}}_\theta)$. We nevertheless retain the Trotter-Suzuki terminology because it provides a compact representation and makes the close analogy to imaginary-time propagation explicit. For example, the second-order Strang ($m = 2$) decomposition [15] takes the form

$$e^{-\Delta\tau\hat{\mathcal{H}}_\theta} \approx e^{\frac{\Delta\tau}{2}\hat{\mathcal{V}}_\theta} e^{\Delta\tau\hat{\mathcal{K}}_\theta} e^{\frac{\Delta\tau}{2}\hat{\mathcal{V}}_\theta}, \tag{13}$$

with local error $\mathcal{O}(\Delta\tau^3)$. Furthermore, in our experiments, we employ two fourth-order schemes, due to Suzuki [16] and Blanes-Moan [17]. Importantly, increasing the decomposition order only changes these fixed coefficients and introduces no additional variational parameters, thereby improving the accuracy of the latent imaginary-time propagation without enlarging the parameter budget.

V. NUMERICAL RESULTS

Next, we validate our framework using the square lattice J_1-J_2 Heisenberg model, whose Hamiltonian is

$$\hat{\mathcal{H}} = J_1 \sum_{\langle i,j\rangle} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_j + J_2 \sum_{\langle\langle i,j\rangle\rangle} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_j, \tag{14}$$

where $\hat{\mathbf{S}}_i = (\hat{S}_i^x, \hat{S}_i^y, \hat{S}_i^z)$ are spin-1/2 operators, $\langle i,\ j\rangle$ denotes nearest neighbors, and $\langle\langle i, j\rangle\rangle$ next-nearest neighbors. All numerical experiments are performed on the 10×10 lattice with $J_2/J_1 = 0.5$ and periodic boundary conditions. We adopt the standard TQS encoder-decoder architecture [4,5,8] and perform all calculations within the VMC framework; unless otherwise stated, we report the best energy among five random seeds. For all implementation details, including the precise network specification, optimization settings, and computing environment, as well as seed-averaged results and the corresponding standard error of the mean across seeds (see Appendix C). We also present additional results beyond the J_1-J_2 benchmark in Appendix I.

A. PITQS ansatz

Table I compares the energies obtained with the PITQS ansatz [Eq. (11)] and the standard TQS ansatz [Eq. (9)]. In both cases, we use the first-order Lie-Trotter scheme for each layer. We fix $\Delta\tau = 0.5$ and compare PITQS and TQS at matched total imaginary times $\beta \in \{1.0, 2.0, 3.0, 4.0\}$, i.e., $L \in \{2, 4, 6, 8\}$ layers. With a constant number of variational parameters N_p, the PITQS ansatz achieves energies comparable to the TQS for each β. For larger β, the PITQS further achieves lower energies. In contrast, N_p for the standard TQS grows approximately linearly with β; nevertheless, increasing N_p by taking larger β does not lead to a systematic improvement in accuracy. These results indicate that conventional TQS with effective Hamiltonians dependent on imaginary time are heavily overparametrized: The essential representational power resides in the structure of the effective Hamiltonian itself.

TABLE I. Energies per site E obtained with the PITQS and standard TQS using the Lie-Trotter scheme. Parentheses indicate statistical (Monte Carlo) errors. We also report the corresponding total imaginary time β and the number of variational parameters N_p for each setting.

β	E (TQS)	N_p (TQS)	E (PITQS)	N_p (PITQS)
1.0	−0.495 93(11)	81 800	−0.495 96(14)	44 890
2.0	−0.496 52(9)	155 620	−0.496 69(9)	44 890
3.0	−0.496 56(11)	229 440	−0.496 76(10)	44 890
4.0	−0.496 51(10)	303 260	−0.496 96(9)	44 890

B. Higher-order decomposition schemes

Furthermore, at a fixed total imaginary time $\beta = 2.0$, we benchmark several update strategies to assess how the decomposition schemes affect the accuracy. Table II shows that increasing the decomposition order consistently tends to improve the energy relative to the first-order Lie-Trotter scheme, indicating a systematic reduction of Trotter errors within the sequential family. When we match the parameter budget to the TQS baseline with $\beta = 0.5$, $N_p = 44\,890$, our PITQS ansätze show a substantial performance advantage. Moreover, compared to standard TQS baselines, the fourth-order Suzuki and Blanes-Moan schemes achieve energies that are better than TQS despite using substantially fewer variational parameters than the deeper TQS model with $\beta = 2.0$, $N_p = 155\,620$. However, we note that higher-order decompositions come with trade-offs in computational cost and numerical stability, which can limit their practical advantages depending on the setting.

C. Computational cost

We evaluate the computational cost of PITQS and the standard TQS in terms of two practical metrics: the total peak usage of graphics processing unit (GPU) memory and the computational time per optimization iteration. All

TABLE II. Energies per site at $\beta = 2.0$ for several PITQS schemes and standard TQS baselines ($\beta = 0.5$ and 2.0). N_p denotes the number of variational parameters.

Scheme	Energy per site	N_p
TQS ($\beta = 0.5$)	−0.493 18(15)	44 890
TQS ($\beta = 2.0$)	−0.496 52(9)	155 620
PITQS (Lie-Trotter)	−0.496 69(9)	44 890
PITQS (Strang)	−0.496 71(9)	44 890
PITQS (Suzuki)	−0.496 97(9)	44 890
PITQS (Blanes-Moan)	−0.496 83(7)	44 890

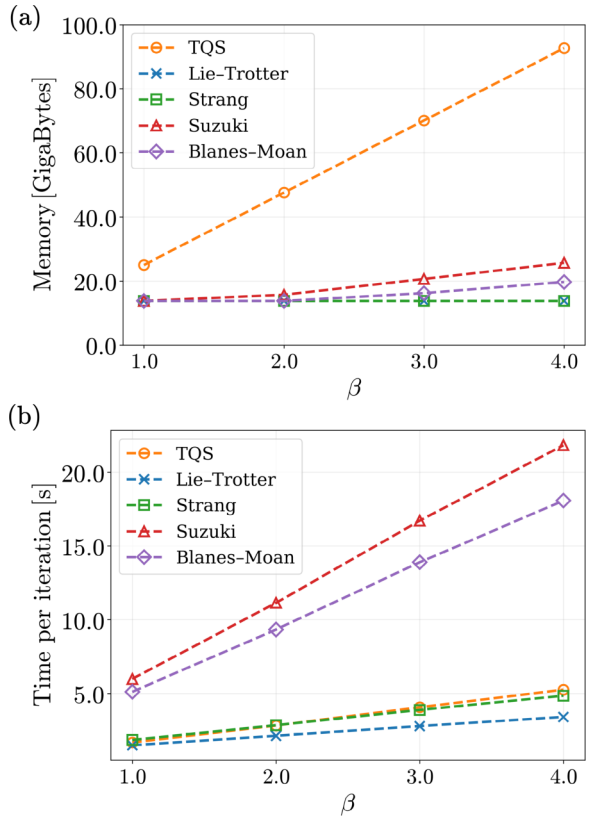

FIG. 2. Computational costs of TQS and PITQS as a function of total imaginary time $\beta \in \{1.0, 2.0, 3.0, 4.0\}$. (a) Peak GPU memory usage. (b) Computational time per optimization iteration in seconds.

benchmarks are performed at fixed step size $\Delta\tau = 0.5$ for total imaginary times $\beta \in \{1.0, 2.0, 3.0, 4.0\}$, and all computational costs are measured in separate profiling runs on two NVIDIA H100 GPUs using identical hyperparameters.

Figure 2(a) compares the peak GPU memory usage of PITQS and the standard TQS. For PITQS, the low-order Lie-Trotter and Strang schemes exhibit an essentially β-independent memory usage, consistent with weight sharing in the LITE blocks, which keeps the number of variational parameters independent of the unrolling depth. Even the fourth-order Suzuki and Blanes-Moan decompositions show only a moderate increase at larger β, owing to the intermediate buffers required for higher-order stages. In contrast, the memory usage of TQS increases strongly with β, reflecting that increasing β requires increasing the depth $L = \beta/\Delta\tau$ and thus the number of distinct layer parameters. Overall, PITQS remains substantially more memory efficient than TQS at larger β.

Figure 2(b) shows the computational time per iteration for optimizing TQS and PITQS. We find that the computational times of the Lie-Trotter and Strang schemes of PITQS are comparable to those of TQS, although TQS exhibits a mild upward trend at larger β, consistent with its substantially larger parameter count relative to PITQS. In contrast, the fourth-order Suzuki and Blanes-Moan decompositions incur a significantly higher cost, typically by a factor of ∼5–7 compared with the lower-order schemes. This increase reflects the larger number of stages k in fourth-order decompositions and

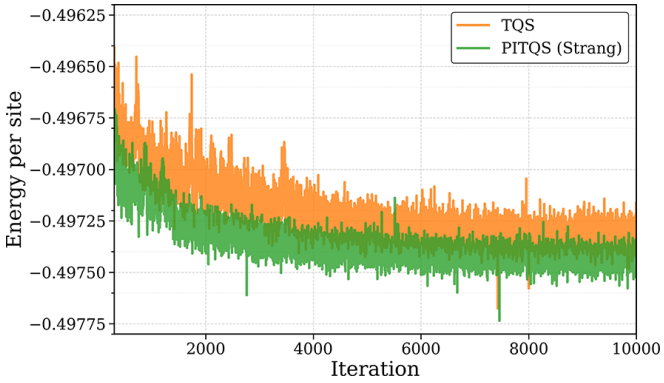

FIG. 3. Optimization of the energy per site (the first 300 iterations are omitted for readability). We compare the PITQS (Strang) ansatz with $N_p = 143\,010$ against a standard TQS with $N_p = 155\,620$.

highlights a practical trade-off between accuracy and computational cost for higher-order schemes. Taken together, these results suggest that the Lie-Trotter or Strang schemes provide a favorable balance of computational cost, memory efficiency, and accuracy in many practical settings.

D. Scalability

We demonstrate the scalability of our framework by targeting high accuracy. We trained the larger PITQS model, a second-order Strang ansatz with $d = 108$, $n_h = 18$, $L = 8$, $\Delta\tau = 0.5$, $N_p = 143\,010$, optimizing it over 10 000 iterations with 8192 Monte Carlo samples per iteration. To ensure a fair comparison, we benchmark it against a standard TQS of comparable parameter count ($d = 60$, $L = 4$, $\Delta\tau = 1.0$, $N_p = 155\,620$). We adopt this baseline because reproducing large-scale TQS [8] ($N_p \approx$ 268k–995k) is computationally prohibitive on our setup.

As illustrated in Fig. 3, our PITQS ansatz substantially outperforms the comparable standard TQS throughout the optimization process, achieving superior accuracy despite using fewer variational parameters ($N_p = 143\,010$ versus $155\,620$). Most notably, the Strang ansatz achieves an energy of $E \approx -0.497\,41(3)$, which is competitive with the values reported in Ref. [8] of $E \approx -0.497\,25$ ($N_p = 267\,720$) and $E \approx -0.497\,30$ ($N_p = 994\,700$), while using only 143 010 parameters. This highlights that a physics-inspired inductive bias in the ansatz, rather than sheer parameter scale, plays a key role in accuracy.

To complement the parameter-matched comparison in Fig. 3, we also assess the two runs at fixed wall-clock cost. Let E_n denote the energy per site measured at the nth optimization iteration. For the resulting sequence $\{E_n\}$, we define the trailing moving average with window w as $\bar{E}_n^{(w)} = (1/w)\sum_{k=1}^{w} E_{n-w+k}$. Using the measured per-iteration times of 22.83 s for the PITQS ansatz and 7.58 s for the TQS baseline, we compare the energies reached within the same wall-clock budget of 21.06 h, corresponding to the full 10 000-iteration TQS run. Within this budget, the PITQS ansatz completes 3320 iterations and attains lower moving-average energies per site than the TQS baseline for all tested window sizes: $-0.497\,350$ versus $-0.497\,307$ for $w = 50$,

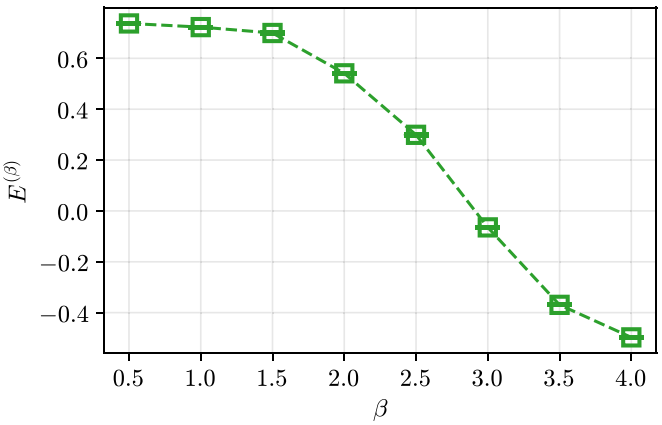

FIG. 4. Energies per site $E^{(\beta)}$ evaluated using 8192 Monte Carlo samples by truncating the PITQS ansatz using the Strang scheme trained at $\beta = 4.0$. Error bars show the Monte Carlo error.

$-0.497\,353$ versus $-0.497\,307$ for $w = 100$, and $-0.497\,351$ versus $-0.497\,310$ for $w = 200$. Therefore, the advantage of the PITQS ansatz is not based solely on parameter count: It remains visible under a fixed-compute comparison as well.

E. Cooling profile in latent space

To directly probe whether the LITE acts as a cooling procedure, we evaluate the energy of a pretrained PITQS at intermediate imaginary-time extents. Specifically, we take the PITQS ansatz with the Strang scheme ($d = 108$, $n_h = 18$, $L = 8$, $\Delta\tau = 0.5$, $N_p = 143\,010$) used in Fig. 3 and evaluate the energy by truncating the unrolling length to $\beta = 0.5, \ldots, 4.0$ while keeping the variational parameters θ^* pretrained at $\beta_{\text{train}} = 4.0$ fixed. In the present benchmark, we fix $\Delta\tau = 0.5$, which yields eight evaluation points $\beta \in \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0\}$. This procedure defines a family of wave functions $\Psi^{(\beta)}_{\theta^*}$ obtained by applying $L = \beta/\Delta\tau$ shared evolution blocks followed by the same decoder $\hat{\mathcal{D}}_{\theta^*}$.

For each β, we estimate the energy per site $E^{(\beta)}$ with fixed trained variational parameters θ^*. In particular, we use 8192 Monte Carlo samples for the evaluation at each β.

Figure 4 shows the resulting estimates of the energy per site $E^{(\beta)}$ for eight evaluation points. Within the training horizon, increasing β systematically lowers the variational energy. This behavior indicates that iterating the LITE steps in the latent space acts as an effective cooling procedure, consistent with the LITE interpretation.

The evaluation is restricted to $\beta \leqslant \beta_{\text{train}} = 8\Delta\tau$, that is, within the training horizon, and does not claim reliable extrapolation beyond it. In practice, evaluating at $\beta > \beta_{\text{train}}$ corresponds to extrapolating the latent evolution beyond the training horizon, where the decoder may experience a distribution shift. Therefore, obtaining reliable predictions at larger β requires fine-tuning with the extended unrolling length.

VI. MACHINE-LEARNING PERSPECTIVE

From a machine-learning perspective, the PITQS ansatz with a single effective Hamiltonian is naturally related to universal transformers [20]: A single transformer layer is iterated over depth with *weight sharing*, analogous to recurrence in the broad sense used for recurrent neural networks [21]. Although weight sharing in transformers is often introduced for parameter efficiency (e.g., ALBERT [22]) and can raise concerns about expressivity, in our setting it corresponds to evolution independent of imaginary time and therefore serves as a constructive inductive bias, enhancing physical transparency without sacrificing accuracy and, in some regimes, even enhancing it.

VII. SUMMARY

In this work, we introduced the physics-inspired framework that reinterprets deep TQS as LITE driven by a single effective Hamiltonian. By enforcing weight sharing across layers and employing higher-order Trotter-Suzuki decompositions, we showed that PITQS architectures can match state-of-the-art TQS models on the J_1-J_2 Heisenberg model using substantially fewer variational parameters than standard TQS architectures. This work bridges the gap between machine-learning expressivity and physical transparency, turning NQS architecture design from heuristic trial and error into a systematic and physically grounded construction.

ACKNOWLEDGMENTS

This work was supported by the Core Research for Evolutional Science and Technology (CREST) program of the Japan Science and Technology Agency (JST), Grant No. JPMJCR1913, and by the Japan Society for the Promotion of Science (JSPS) KAKENHI Grants No. JP22H00516, No. JP22H05106 (Y.K.), No. JP25K15230 (T.K.), and No. JP24K20864 (I.S.).

DATA AVAILABILITY

The implementation source code associated with this work is openly available in Zenodo [23].

APPENDIX A: IMPLEMENTATION OF $\hat{\mathcal{K}}_\theta$ AND $\hat{\mathcal{V}}_\theta$

We describe the implementation of the nonlocal operator $\hat{\mathcal{K}}_\theta$ and the local operator $\hat{\mathcal{V}}_\theta$ used in our numerical experiments for PITQS. As an illustrative example, we present the layer update in the form of the Lie-Trotter scheme. A discrete configuration n is first mapped to a latent-token state $z(0) \in \mathbb{R}^{N_{\text{tok}} \times d}$ consisting of N_{tok} tokens. The ith row $z_i(0) \in \mathbb{R}^d$ is a d-dimensional vector for the token at index $i \in \{1, \ldots, N_{\text{tok}}\}$. We then apply a transformation composed of L layers to $z(0)$ to obtain $z(\beta)$. Following the notation in the main text, we denote the state in the ℓth layer by $z^{(\ell)} \in \mathbb{R}^{N_{\text{tok}} \times d}$. We write the update from $z^{(\ell-1)}$ to $z^{(\ell)}$ as

$$\tilde{z}^{(\ell)} = z^{(\ell-1)} + \Delta\tau\, \hat{\mathcal{K}}_\theta\, [z^{(\ell-1)}], \tag{A1}$$

$$z^{(\ell)} = \tilde{z}^{(\ell)} + \Delta\tau\, \hat{\mathcal{V}}_\theta\, [\tilde{z}^{(\ell)}], \tag{A2}$$

where both operators $\hat{\mathcal{K}}_\theta[\cdot]$ and $\hat{\mathcal{V}}_\theta[\cdot]$ internally apply a prelayer normalization (pre-LN) to their inputs.

The nonlocal operator $\hat{\mathcal{K}}_\theta$ is implemented by factored MHA (FMHA) [8,24]. FMHA first normalizes the input

$z^{(\ell-1)}$ using pre-LN: $\bar{z}^{(\ell-1)} = \mathrm{LN}(z^{(\ell-1)})$. Then, for each head $h = 1, \ldots, n_h$, the value features are computed as

$$V^{(h)} = \bar{z}^{(\ell-1)} W_v^{(h)} \in \mathbb{R}^{N_{\mathrm{tok}} \times d_h}, \tag{A3}$$

where $d_h = d/n_h$ and $W_v^{(h)} \in \mathbb{R}^{d \times d_h}$. Next, FMHA mixes tokens for head h as

$$Y^{(h)} = A^{(h)} V^{(h)} \in \mathbb{R}^{N_{\mathrm{tok}} \times d_h}. \tag{A4}$$

Here, $A^{(h)}$ is a trainable, input-independent token-token kernel matrix given by

$$A^{(h)} = \left[\alpha_{ij}^{(h)}\right] \in \mathbb{R}^{N_{\mathrm{tok}} \times N_{\mathrm{tok}}}, \tag{A5}$$

where $(i,\ j)$-entry $\alpha_{i,j}^{(h)}$ is the attention weight that represents the effective interaction strength between tokens i and j. After concatenating the head outputs $Y^{(h)}$ along the feature dimension and applying an output projection $W_o \in \mathbb{R}^{d \times d}$, the map is written as

$$\hat{\mathcal{K}}_\theta[z^{(\ell-1)}] = \mathrm{Concat}\,(Y^{(1)}, \ldots, Y^{(n_h)})\, W_o \in \mathbb{R}^{N_{\mathrm{tok}} \times d}. \tag{A6}$$

This representation makes the nonlocality explicit: The kernel $A^{(h)}$ couples all token pairs $(i,\ j)$. For translationally invariant settings, one may further constrain $\alpha_{ij}^{(h)}$ to depend only on relative displacements.

The local operator $\hat{\mathcal{V}}_\theta$ is implemented by a positionwise FFN. As in $\hat{\mathcal{K}}_\theta$, we absorb pre-LN in the definition of the operator. Namely, given an input $\tilde{z}^{(\ell)}$, we first obtain the normalized tokens $\check{z}^{(\ell)} = \mathrm{LN}(\tilde{z}^{(\ell)})$ and then apply the same two-layer MLP independently to each token index $i = 1, \ldots, N_{\mathrm{tok}}$:

$$(\hat{\mathcal{V}}_\theta[\tilde{z}^{(\ell)}])_i = \mathrm{GELU}\big(\check{z}_i^{(\ell)} W_1 + b_1^\top\big) W_2 + b_2^\top. \tag{A7}$$

Here, $\check{z}_i^{(\ell)} \in \mathbb{R}^d$ denotes the ith row of $\check{z}^{(\ell)}$. $W_1 \in \mathbb{R}^{d \times d_{\mathrm{FFN}}}$ and $W_2 \in \mathbb{R}^{d_{\mathrm{FFN}} \times d}$ are weight matrices, $b_1 \in \mathbb{R}^{d_{\mathrm{FFN}}}$ and $b_2 \in \mathbb{R}^d$ are bias vectors, and the hidden dimension is chosen as $d_{\mathrm{FFN}} = 4d$. This map is explicitly local (on site): The output at index i depends only on the specific input token $\tilde{z}_i^{(\ell)}$ and is not mixed with the other tokens.

APPENDIX B: PITQS FOR FERMION SYSTEMS

We introduce PITQS tailored to fermionic many-body systems. Compared with the spin-system PITQS, there are three essential differences: (1) the encoder that maps an occupation configuration to tokens, (2) the implementation of the LITE block, and (3) the incorporation of antisymmetry in the decoder. Following the construction of fermionic TQS in, e.g., Refs. [6,7], we build fermionic PITQS as follows.

We represent a fermionic occupation configuration by $n \in \{0, 1\}^{2N}$. For each spatial site $i = 1, \ldots, N$, we consider a local class label c_i that takes one of four values and encodes the on-site occupancy:

$$c_i \equiv n_{i,\uparrow} + 2\, n_{i,\downarrow} \in \{0, 1, 2, 3\}, \tag{B1}$$

where $n_{i\sigma}$ denotes the occupation number of spin σ at site i. The encoder embeds these discrete labels with a learned continuous embedding matrix $E \in \mathbb{R}^{4 \times d}$, producing token features

$$z_i(0) = E_{c_i} + p_i \in \mathbb{R}^d, \quad i = 1, \ldots, N, \tag{B2}$$

where $\{p_i\}_{i=1}^N$ are learned positional embeddings. Collecting tokens gives $z(0) \in \mathbb{R}^{N \times d}$, and hence we set $N_{\mathrm{tok}} = N$ in the fermionic setting.

Next, the nonlocal operator $\hat{\mathcal{K}}_\theta$ is implemented by standard MHA [25] with queries and keys, following standard fermionic TQS architectures [6,7]. As in FMHA, we first normalize the input $z^{(\ell-1)} \in \mathbb{R}^{N_{\mathrm{tok}} \times d}$ as $\bar{z}^{(\ell-1)} = \mathrm{LN}(z^{(\ell-1)})$. Then, for each head $h = 1, \ldots, n_h$ with $d_h = d/n_h$, we compute the query, key, and value matrices

$$Q^{(h)} = \bar{z}^{(\ell-1)} W_Q^{(h)}, \tag{B3}$$

$$K^{(h)} = \bar{z}^{(\ell-1)} W_K^{(h)}, \tag{B4}$$

$$V^{(h)} = \bar{z}^{(\ell-1)} W_V^{(h)}, \tag{B5}$$

where $W_Q^{(h)}, W_K^{(h)}, W_V^{(h)} \in \mathbb{R}^{d \times d_h}$. The (input-dependent) token-token kernel matrix is defined by the row-wise softmax function:

$$A^{(h)}(\bar{z}^{(\ell-1)}) = \mathrm{Softmax}\left(\frac{Q^{(h)} K^{(h)\top}}{\sqrt{d_h}}\right). \tag{B6}$$

The head output is then written in the same form as FMHA,

$$Y^{(h)} = A^{(h)}(\bar{z}^{(\ell-1)})\, V^{(h)} \in \mathbb{R}^{N_{\mathrm{tok}} \times d_h}, \tag{B7}$$

and after concatenation and an output projection $W_o \in \mathbb{R}^{d \times d}$, we obtain

$$\hat{\mathcal{K}}_\theta[z^{(\ell-1)}] = \mathrm{Concat}\,(Y^{(1)}, \ldots, Y^{(n_h)})\, W_o \in \mathbb{R}^{N_{\mathrm{tok}} \times d}. \tag{B8}$$

This representation makes the nonlocality explicit: $A^{(h)}(z^{(\ell-1)})$ couples all token pairs (i, j) and is state dependent through $Q^{(h)}$ and $K^{(h)}$. The local operator $\hat{\mathcal{V}}_\theta$ follows the same structure as Eq. (A7), absorbing the pre-LN step. Here, we employ SiLU as the nonlinearity instead of GELU and set the hidden dimension to $d_{\mathrm{FFN}} = 2d$:

$$(\hat{\mathcal{V}}_\theta[\tilde{z}^{(\ell)}])_i = \mathrm{SiLU}\big(\check{z}_i^{(\ell)} W_1 + b_1^\top\big) W_2 + b_2^\top, \tag{B9}$$

where $\check{z}_i^{(\ell)}$ denotes the ith row of the normalized tokens $\mathrm{LN}(\tilde{z}^{(\ell)})$. The parameter shapes are identical to those in Eq. (A7) except for the hidden dimension. Finally, we enforce weight sharing across layers; i.e., the PITQS has a single static effective Hamiltonian

$$\hat{\mathcal{H}}_\theta = -(\hat{\mathcal{V}}_\theta + \hat{\mathcal{K}}_\theta), \tag{B10}$$

and the LITE operator is $\hat{\mathcal{U}}_\theta(\beta) = e^{-\beta \hat{\mathcal{H}}_\theta}$. In practice, $\hat{\mathcal{U}}_\theta(\beta)$ is realized by applying a shared transformer layer with $\hat{\mathcal{V}}_\theta$ and $\hat{\mathcal{K}}_\theta$ L times. The layer is implemented as a Trotter-Suzuki decomposition of the exponential (with step size $\Delta\tau = \beta/L$).

For the decoder $\hat{\mathcal{D}}_\theta$, we employ a multi-Slater backflow [6,7] to enforce antisymmetry. From each decoded token $z_i(\beta)$, we generate configuration-dependent backflow orbitals by a shared, positionwise affine map,

$$x_i(\beta) = \mathrm{LN}(z_i(\beta))\, W_{\mathrm{out}} + \mathbf{1} b_{\mathrm{out}}^\top, \tag{B11}$$

with $W_{\mathrm{out}} \in \mathbb{R}^{d \times (2KN_e)}$ and $b_{\mathrm{out}} \in \mathbb{R}^{2KN_e}$. The vector x_i is interpreted as K sets of spin-resolved orbitals: For each determinant $k \in \{1, \ldots, K\}$, we extract two N_e-dimensional vectors $x_{i,\downarrow}^{(k)}$ and $x_{i,\uparrow}^{(k)}$, which we regard as the rows of a backflow orbital matrix $M^{(k)}(n) \in \mathbb{R}^{2N \times N_e}$. Concretely, the row of

$M^{(k)}(n)$ associated with the spin orbital (i, σ) is $x^{(k)}_{i,\sigma}$ in the chosen spin-orbital ordering. Given an occupation configuration $n \in \{0, 1\}^{2N}$, let $P(n) \in \{0, 1\}^{N_e \times 2N}$ be a row-selection matrix that picks the occupied spin orbitals. Then, the kth Slater matrix $\Phi^{(k)}_\theta(n)$ is represented as a matrix product:

$$\Phi^{(k)}_\theta(n) = P(n) M^{(k)}_\theta(n) \in \mathbb{R}^{N_e \times N_e}. \tag{B12}$$

Finally, the decoder outputs the complex log amplitude corresponding to a real multideterminant backflow form

$$\hat{\mathcal{D}}_\theta[z(\beta)] = \log \left| \sum_{k=1}^{K} \det \Phi^{(k)}_\theta(n) \right| \tag{B13}$$

$$+ i\pi \, \mathbb{I}\left[\sum_{k=1}^{K} \det \Phi^{(k)}_\theta(n) < 0 \right], \tag{B14}$$

where $\mathbb{I}[\cdot]$ is the indicator function.

APPENDIX C: EXPERIMENTAL DETAILS

All numerical experiments were carried out in the VMC framework, where the variational energy is estimated from Monte Carlo samples $n \sim p_\theta(n)$ with $p_\theta(n) \propto |\Psi_\theta(n)|^2$ as $\mathbb{E}_{n\sim p_\theta}[(\hat{\mathcal{H}}\Psi_\theta)(n)/\Psi_\theta(n)]$. Our PITQS architectures were implemented in the NetKet library [26,27], which is built on JAX [28], Flax [29], and mpi4jax [30]. Unless otherwise stated, we report the best final energy obtained after 800 iterations among five independent runs with random initializations. For a detailed analysis of stability across random seeds, see Appendix D. Computations were performed on two NVIDIA A100 GPUs with 40 GB of memory each for models that fit in memory, while larger models were run on two NVIDIA H100 GPUs with 80 GB each.

For the J_1-J_2 Heisenberg model, the variational parameters were optimized using the minimum-step stochastic reconfiguration (MinSR) [8,18] with a learning rate of 0.0075 and a diagonal shift of 10^{-4}. Unless otherwise specified, we performed 800 optimization iterations with 4096 Monte Carlo samples per iteration and set the hyperparameters to model dimension $d = 60$, patch size 2×2, number of heads $n_h = 10$, and FFN hidden dimension $4d$. For spin systems, we used the translation-invariant FMHA setting, i.e., $\alpha^{(h)}_{ij} = \alpha^{(h)}_{i-j}$. The optimization budget and these hyperparameters follow the baseline setting of the NetKet tutorial on the vision transformer wave function; (see Ref. [31]).

For the Hubbard model, we optimized the variational parameters using MinSR with a learning rate of 0.01 and a diagonal shift of 10^{-4}. We performed 10 000 optimization iterations with 4096 Monte Carlo samples per iteration. The hyperparameters were set to model dimension $d = 16$, number of heads $n_h = 4$, depth $L = 4$, and imaginary-time step $\Delta\tau = 0.5$.

APPENDIX D: COMPUTATIONAL STABILITY OF PITQS

We examine how the computational stability of PITQS depends on the total imaginary time $\beta = L\Delta\tau$. While the main text reports the best energy obtained among independent runs, here we focus on the run-to-run variability and evaluate the summary statistics, namely, the mean and standard error. We set the step size to $\Delta\tau = 0.5$ and sweep β over $\{1.0, 2.0, 3.0, 4.0\}$ while keeping all other settings fixed. These values of β correspond to the depths $L \in \{2, 4, 6, 8\}$, respectively. To assess stability, we repeat each experiment with five independent random seeds and summarize the resulting run-to-run variability. The experiment is performed on two NVIDIA A100 GPUs, except for the standard TQS baseline at $\beta \in \{3.0, 4.0\}$, which is run on two NVIDIA H100 GPUs due to memory constraints.

TABLE III. Energies per site averaged over five random seeds, the corresponding SEM across seeds, and the number of variational parameters N_p for the PITQS and TQS ansätze at $\beta = 1.0, 2.0, 3.0, 4.0$.

$\beta = 1.0$			
Scheme	Mean	SEM ($\times 10^{-4}$)	N_p
TQS	−0.495 76	0.82	81 800
PITQS (Lie-Trotter)	−0.495 79	0.56	44 890
PITQS (Strang)	−0.495 63	2.3	44 890
PITQS (Suzuki)	−0.496 42	0.88	44 890
PITQS (Blanes-Moan)	−0.495 78	3.6	44 890
$\beta = 2.0$			
Scheme	Mean	SEM ($\times 10^{-4}$)	N_p
TQS	−0.495 83	4.8	155 620
PITQS (Lie-Trotter)	−0.496 26	1.2	44 890
PITQS (Strang)	−0.496 33	1.3	44 890
PITQS (Suzuki)	−0.496 53	2.7	44 890
PITQS (Blanes-Moan)	−0.496 72	0.32	44 890
$\beta = 3.0$			
Scheme	Mean	SEM ($\times 10^{-4}$)	N_p
TQS	−0.496 21	2.0	229 440
PITQS (Lie-Trotter)	−0.496 69	0.34	44 890
PITQS (Strang)	−0.496 65	0.35	44 890
PITQS (Suzuki)	−0.496 08	2.9	44 890
PITQS (Blanes-Moan)	−0.496 58	1.1	44 890
$\beta = 4.0$			
Scheme	Mean	SEM ($\times 10^{-4}$)	N_p
TQS	−0.496 45	0.27	303 260
PITQS (Lie-Trotter)	−0.496 54	2.5	44 890
PITQS (Strang)	−0.496 59	0.64	44 890
PITQS (Suzuki)	−0.496 30	2.8	44 890
PITQS (Blanes-Moan)	−0.496 13	3.6	44 890

Table III reports the energy per site averaged over five random seeds together with the corresponding standard error of the mean (SEM) across seeds. In the small-β regime ($\beta = 1.0, 2.0$), higher-order decompositions (Suzuki and Blanes-Moan) exhibit superior accuracy, yielding lower mean energies. However, in the large-β regime ($\beta = 3.0, 4.0$), the performance of these high-order methods deteriorates significantly; they suffer from both higher mean energies and increased SEMs, indicating optimization instability. Conversely, the low-order Lie-Trotter and Strang schemes maintain high robustness even at larger β and are more stable than the higher-order schemes while achieving competitive or better energies than TQS. These observations highlight a practical trade-off for higher-order decompositions: While

TABLE IV. Energy per site for $\Delta\tau \in \{0.1,\ 0.5,\ 1.0\}$ at fixed depth $L = 4$. Parentheses indicate Monte Carlo errors, and N_p denotes the number of variational parameters.

Scheme	$\Delta\tau = 0.1$	$\Delta\tau = 0.5$	$\Delta\tau = 1.0$	N_p
TQS	−0.496 20(11)	−0.496 52(9)	−0.496 66(9)	155 620
PITQS (Lie-Trotter)	−0.496 81(10)	−0.496 69(9)	−0.496 59(10)	44 890
PITQS (Strang)	−0.496 67(9)	−0.496 71(9)	−0.496 67(9)	44 890
PITQS (Suzuki)	−0.496 78(8)	−0.496 97(9)	−0.496 96(9)	44 890
PITQS (Blanes-Moan)	−0.496 79(10)	−0.496 83(7)	−0.497 01(11)	44 890

they can reduce Trotter error at moderate β, their increased optimization sensitivity at larger β warrants care when selecting $(L, \Delta\tau)$ in practice.

APPENDIX E: DEPENDENCE ON THE IMAGINARY-TIME STEP SIZE $\Delta\tau$

We examine how the energy per site depends on the imaginary-time step size $\Delta\tau$ used to discretize the LITE operator. We treat $\Delta\tau$ as a hyperparameter and evaluate the three values $\Delta\tau \in \{0.1,\ 0.5,\ 1.0\}$ with the depth fixed at $L = 4$, i.e., $\beta \in \{0.4,\ 2.0,\ 4.0\}$.

Table IV summarizes the resulting energies per site and lists N_p for reference. The column at $\Delta\tau = 0.5$ corresponds to the setting used in the main text. At fixed depth $L = 4$, the dependence on $\Delta\tau$ varies across the schemes, reflecting their distinct theoretical properties. For PITQS, the observed trends are consistent with the theoretical properties of the Trotter-Suzuki decompositions. The Lie-Trotter scheme yields slightly higher energies as $\Delta\tau$ increases, which is expected for a first-order approximation. The Strang decomposition shows no significant degradation within the statistical uncertainty, consistent with its second-order nature. Notably, the fourth-order Suzuki and Blanes-Moan decompositions maintain high accuracy even at $\Delta\tau = 1.0$, demonstrating the benefit of higher-order decompositions at larger $\Delta\tau$.

TQS exhibits a trend opposite to PITQS with the Lie-Trotter scheme. Although TQS effectively implements the first-order Lie-Trotter decomposition, we confirm that its accuracy improves with larger $\Delta\tau$. Since the depth L is fixed, increasing $\Delta\tau$ implies a larger total imaginary time β, which generally aids ground-state cooling. This behavior of TQS is likely attributable to its layer-dependent effective Hamiltonians, which lead to a larger variational freedom and allow TQS to absorb finite-$\Delta\tau$ Trotter errors.

APPENDIX F: EFFECTIVE COUPLING FROM ATTENTION MECHANISM

To further examine the physical content of the learned static effective Hamiltonian, we analyze the optimized headwise effective couplings associated with the attention mechanism in the FMHA component of PITQS. Recent TQS studies have shown that visualizing attention patterns provides useful insight into the learned correlation structure [7]. From the present LITE viewpoint, such attention analyses can be interpreted as a direct visualization of the learned effective couplings in latent space.

In this Appendix, we focus on the PITQS ansatz used in Fig. 3. For each head h, the nonlocal part is characterized by a coupling matrix $A^{(h)} = [\alpha_{ij}^{(h)}]$. In the translation-invariant setting used here, the coupling depends only on the relative displacement between patches, $\alpha_{ij}^{(h)} = \alpha_{i-j}^{(h)}$. Therefore, although the full coupling matrix is of size 25×25, it is completely determined by a single 5×5 map of relative-displacement couplings.

Figure 5 shows the absolute values of the relative-displacement couplings for all attention heads in the optimized PITQS. Several qualitative features are evident. First, the dominant weight of most heads is concentrated on a small number of short-range patch displacements, showing that the learned nonlocal coupling is highly structured rather than broadly distributed over all patch pairs. Second, different heads specialize in distinct displacement patterns: Some emphasize horizontal nearest-neighbor-like displacements, others emphasize vertical ones, and a few exhibit more localized or more diffuse short-range structures. In this sense, the optimized attention heads decompose the learned nonlocal coupling into a set of complementary geometric patterns.

This head specialization supports the LITE interpretation of PITQS. Since the nonlocal part of the effective Hamiltonian is represented by a single shared coupling matrix, the patterns in Fig. 5 can be interpreted as a structured decomposition of one static nonlocal coupling in latent space. We emphasize that these couplings should be interpreted as induced effective couplings in latent space, rather than as a direct reconstruction of the bare couplings of the original Hamiltonian. Some heads may reflect the interaction geometry, such as the nearest- and next-nearest-neighbor exchange structure of the underlying lattice Hamiltonian, while other heads may encode correlation structures emerging in the optimized low-energy wave function. This interpretation is consistent with recent attention analyses of TQS, where learned attention heads were shown to capture not only the lattice structure of the Hamiltonian but also correlations at longer length scales [7]. From the LITE viewpoint, such emergent structures can be regarded as physical information encoded in the effective Hamiltonian learned in latent space.

APPENDIX G: CONTINUOUS LIMIT AND RUNGE-KUTTA INTEGRATION

Our operator-centric LITE formulation admits a continuous latent imaginary-time limit, echoing the viewpoint of neural ordinary differential equations [32] and a related view of transformer architectures as discretized dynamical systems (e.g., Macaron Net [33]). In our setting, the effective

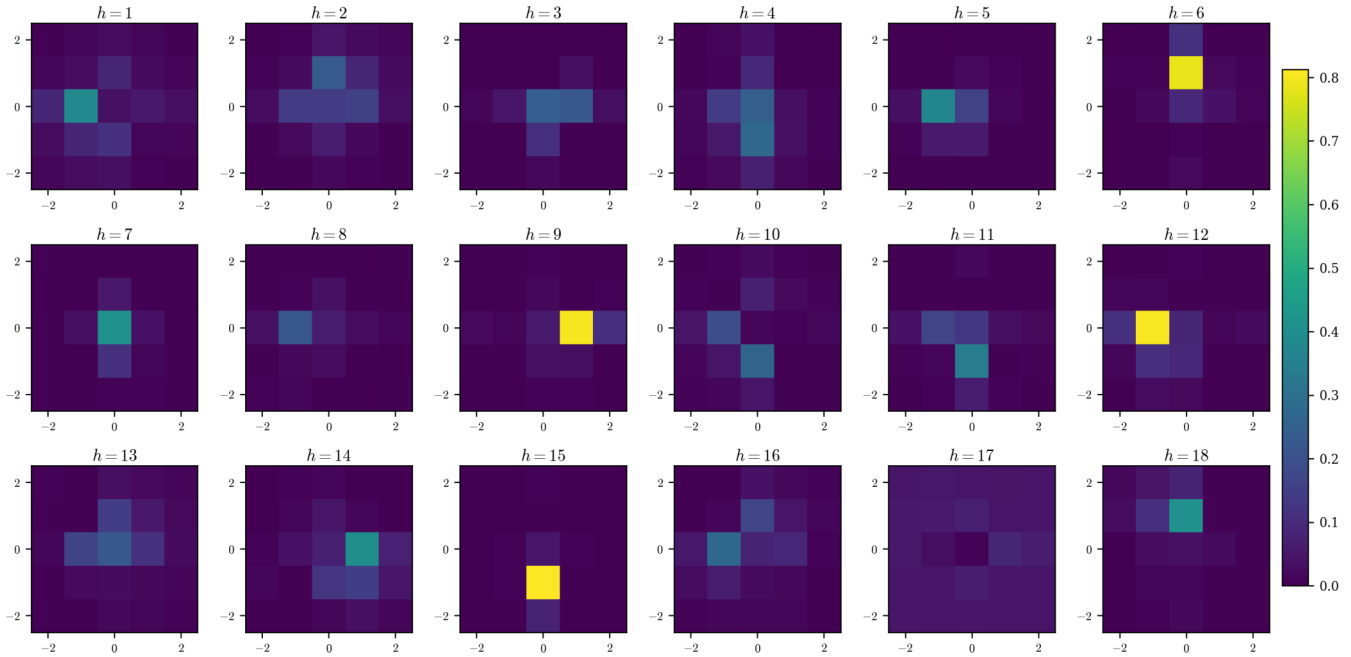

FIG. 5. Absolute values of the effective couplings encoded by the attention mechanism for all heads in the PITQS. Each panel shows the 5×5 map of absolute relative-displacement couplings of the corresponding head. The horizontal and vertical axes label the relative patch displacement along the two directions of the patch lattice.

Hamiltonian defines a latent-space dynamics whose solution is approximated by a depth-discretized update. This discretization introduces two distinct sources of approximation error.

First, when the effective Hamiltonian is decomposed into noncommuting components (i.e., $\hat{\mathcal{K}}_\theta$ and $\hat{\mathcal{V}}_\theta$), Trotter-Suzuki decompositions incur a Trotter error, which we can reduce by using higher-order decompositions. Second, each component causes numerical integration errors when we approximate its action with discrete updates. To mitigate this error, we can adopt higher-order numerical methods instead of the first-order Euler method presented in "latent imaginary-time evolution" and used elsewhere in our experiments. Specifically, we implement a second-order Runge-Kutta (RK2) method, i.e., Heun's method [34]. For each component $\hat{\mathcal{A}}_\theta \in \{\hat{\mathcal{K}}_\theta, \hat{\mathcal{V}}_\theta\}$, this method updates the latent state from z to z' using the following predictor-corrector step:

$$\begin{aligned} k_1 &= \hat{\mathcal{A}}_\theta[z], \\ \tilde{z} &= z + \Delta\tau k_1, \\ z' &= z + \frac{\Delta\tau}{2}(k_1 + \hat{\mathcal{A}}_\theta[\tilde{z}]). \end{aligned} \tag{G1}$$

This approach reduces the local integration error, providing a more accurate approximation of the exponential operators $e^{\Delta\tau\hat{\mathcal{K}}_\theta}$ and $e^{\Delta\tau\hat{\mathcal{V}}_\theta}$.

Table V reports the energies per site of the Euler and RK2 methods with the Lie-Trotter scheme. The results of the Euler method correspond to the ones of PITQS in Table I. At $\beta = 1.0$ and $\beta = 2.0$, the RK2 method yields a modest improvement over the Euler method within statistical uncertainty, whereas at $\beta = 3.0$ and $\beta = 4.0$ the two methods become statistically indistinguishable. These results suggest that at larger β, the potential accuracy gain from RK2 is masked by optimization instability, mirroring the behavior observed in the high-order Suzuki and Blanes-Moan decompositions.

APPENDIX H: PARALLEL FORMULATION

Throughout the main text and the preceding Appendixes, we primarily employ a *serialized* formulation in which the nonlocal and local components are applied sequentially within each layer. For the first-order Lie-Trotter scheme, a single step with $\Delta\tau = \beta/L$ can be written as

$$\tilde{z}^{(\ell)} = z^{(\ell-1)} + \Delta\tau\, \hat{\mathcal{K}}_\theta\, [z^{(\ell-1)}], \tag{H1}$$

$$z^{(\ell)} = \tilde{z}^{(\ell)} + \Delta\tau\, \hat{\mathcal{V}}_\theta\, [\tilde{z}^{(\ell)}]. \tag{H2}$$

Equivalently, using $\hat{\mathcal{H}}_\theta = -(\hat{\mathcal{V}}_\theta + \hat{\mathcal{K}}_\theta)$, this serialized formulation corresponds to approximating the full evolution operator $\hat{\mathcal{U}}_\theta(\Delta\tau) = e^{-\Delta\tau\hat{\mathcal{H}}_\theta} = e^{\Delta\tau(\hat{\mathcal{K}}_\theta+\hat{\mathcal{V}}_\theta)}$ by a product of exponentials of the components. When $\hat{\mathcal{K}}_\theta$ and $\hat{\mathcal{V}}_\theta$ do not commute, this decomposition incurs a Trotter error, which motivates the use of higher-order decompositions discussed elsewhere in this work.

Here, we consider an alternative *parallel* formulation, where the effective Hamiltonian is kept in the exact sum

TABLE V. Energies per site E obtained with the Euler and the RK2 substeps at identical total imaginary times.

β	E (Euler)	E (RK2)
1.0	−0.495 96(14)	−0.496 28(11)
2.0	−0.496 69(9)	−0.496 82(10)
3.0	−0.496 76(10)	−0.496 79(10)
4.0	−0.496 96(9)	−0.496 95(8)

TABLE VI. Energies per site E obtained with the parallel and serialized formulations at identical total imaginary times.

β	E (serialized)	N_p (serialized)	E (parallel)	N_p (parallel)
1.0	−0.495 96(14)	44 890	−0.492 85(16)	44 770
2.0	−0.496 69(9)	44 890	−0.495 89(11)	44 770
3.0	−0.496 76(10)	44 890	−0.496 47(11)	44 770
4.0	−0.496 96(9)	44 890	−0.496 65(10)	44 770

form $\hat{\mathcal{H}}_\theta = -(\hat{\mathcal{V}}_\theta + \hat{\mathcal{K}}_\theta)$ and the update is performed without operator decomposition. In this formulation, the update with the Euler method follows:

$$z^{(\ell)} = e^{-\Delta\tau\hat{\mathcal{H}}_\theta}[z^{(\ell-1)}] \approx z^{(\ell-1)} - \Delta\tau\hat{\mathcal{H}}_\theta\,[z^{(\ell-1)}]. \tag{H3}$$

Because the formulation uses the full effective Hamiltonian $\hat{\mathcal{H}}_\theta$ directly, it does not introduce a Trotter error associated with splitting $\hat{\mathcal{K}}_\theta$ and $\hat{\mathcal{V}}_\theta$. In recent machine learning, related parallel formulations have been adopted in large-scale models to reduce serialized computation within each layer [35,36]. However, in practice, the parallel structure can be more sensitive during optimization and may require stabilization.

Table VI compares the results of the parallel and serialized formulations under identical hyperparameters at fixed step size $\Delta\tau = 0.5$ and $\beta \in \{1.0, 2.0, 3.0, 4.0\}$. The serialized results correspond to the ones of PITQS in Table I. Across the tested range, the serialized formulation achieves slightly lower variational energies than the parallel scheme, while the parameter counts remain nearly identical. The performance gap is most pronounced at $\beta = 1.0$ and becomes smaller at larger β. This trend suggests that, for the present model sizes and optimization settings, the practical advantage of the serialized update is not limited by Trotter error alone; rather, the sequential application of $\hat{\mathcal{K}}_\theta$ and $\hat{\mathcal{V}}_\theta$ within each layer can provide a more favorable inductive bias and optimization behavior than the parallel coupling in Eq. (H3).

TABLE VII. Energy per site E for the 4×4 Hubbard model at half filling for $U/t = 4$ and 8 with periodic boundary conditions. N_p denotes the number of variational parameters. Reference values from exact diagonalization (ED) reported in Ref. [37] are also shown.

Method	$E\ (U/t = 4)$	$E\ (U/t = 8)$	N_p
ED	−0.851 37	−0.529 31	N/A
TQS	−0.851 14(13)	−0.529 09(25)	11 424
PITQS	−0.851 13(22)	−0.529 19(35)	4752

APPENDIX I: BENCHMARK FOR HUBBARD MODELS

We benchmark PITQS on the two-dimensional Hubbard model and compare with the standard fermionic TQS. The Hubbard Hamiltonian is given by

$$\hat{\mathcal{H}} = -t \sum_{\langle i,j\rangle,\sigma} (\hat{c}^\dagger_{i\sigma}\hat{c}_{j\sigma} + \text{H.c.}) + U \sum_i \hat{n}_{i\uparrow}\hat{n}_{i\downarrow}, \tag{I1}$$

where $\hat{c}^\dagger_{i\sigma}$ ($\hat{c}_{i\sigma}$) creates (annihilates) an electron with spin $\sigma \in \{\uparrow, \downarrow\}$ at site i and $\hat{n}_{i\sigma} = \hat{c}^\dagger_{i\sigma}\hat{c}_{i\sigma}$ is the number operator. The first term describes the hopping between nearest-neighbor sites $\langle i, j\rangle$ with amplitude t, and the second term represents the on-site Coulomb repulsion with strength U. The implementation and experimental details follow Appendixes B and C.

Table VII summarizes the energies for the 4×4 Hubbard model at half filling with periodic boundary conditions, comparing PITQS and TQS at interaction strengths $U/t = 4$ and $U/t = 8$ under identical hyperparameters. As in the J_1-J_2 Heisenberg benchmark in Table I, PITQS achieves an accuracy comparable to that of TQS while using substantially fewer variational parameters. These results indicate that enforcing a static effective Hamiltonian through weight sharing provides a parameter-efficient inductive bias that remains effective beyond spin systems, including interacting fermions.

[1] M. Imada, A. Fujimori, and Y. Tokura, Metal-insulator transitions, Rev. Mod. Phys. **70**, 1039 (1998).

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

[3] W. L. McMillan, Ground state of liquid He^4, Phys. Rev. **138**, A442 (1965).

[4] L. L. Viteritti, R. Rende, and F. Becca, Transformer variational wave functions for frustrated quantum spin systems, Phys. Rev. Lett. **130**, 236401 (2023).

[5] L. L. Viteritti, R. Rende, A. Parola, S. Goldt, and F. Becca, Transformer wave function for two dimensional frustrated magnets: Emergence of a spin-liquid phase in the Shastry-Sutherland model, Phys. Rev. B **111**, 134411 (2025).

[6] M. Geier, K. Nazaryan, T. Zaklama, and L. Fu, Self-attention neural network for solving correlated electron problems in solids, Phys. Rev. B **112**, 045119 (2025).

[7] Y. Gu, W. Li, H. Lin, B. Zhan, R. Li, Y. Huang, D. He, Y. Wu, T. Xiang, M. Qin, L. Wang, and D. Lv, Solving the Hubbard model with neural quantum states, Nat. Commun. (2026).

[8] R. Rende, L. L. Viteritti, L. Bardone, F. Becca, and S. Goldt, A simple linear algebra identity to optimize large-scale neural network quantum states, Commun. Phys. **7**, 260 (2024).

[9] M. Gell-Mann and F. Low, Bound states in quantum field theory, Phys. Rev. **84**, 350 (1951).

[10] R. Blankenbecler, D. J. Scalapino, and R. L. Sugar, Monte Carlo calculations of coupled boson-fermion systems. I, Phys. Rev. D **24**, 2278 (1981).

[11] G. Sugiyama and S. Koonin, Auxiliary field Monte-Carlo for quantum many-body ground states, Ann. Phys. **168**, 1 (1986).

[12] G. Carleo, Y. Nomura, and M. Imada, Constructing exact representations of quantum many-body systems with deep neural networks, Nat. Commun. **9**, 5322 (2018).

[13] S. Lie, *Theorie der Transformationsgruppen* (B. G. Teubner, Leipzig, 1888), Vol. 1.

[14] H. F. Trotter, On the product of semi-groups of operators, Proc. Am. Math. Soc. **10**, 545 (1959).

[15] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. **5**, 506 (1968).

[16] M. Suzuki, Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations, Phys. Lett. A **146**, 319 (1990).

[17] S. Blanes and P. Moan, Practical symplectic partitioned Runge–Kutta and Runge–Kutta–Nyström methods, J. Comput. Appl. Math. **142**, 313 (2002).

[18] A. Chen and M. Heyl, Empowering deep neural quantum states through efficient optimization, Nat. Phys. **20**, 1476 (2024).

[19] S. Blanes, F. Casas, and A. Murua, Splitting methods for differential equations, Acta Numer. **33**, 1 (2024).

[20] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and L. Kaiser, Universal transformers, in *Proceedings of the International Conference on Learning Representations* (ICLR, 2019).

[21] J. L. Elman, Finding structure in time, Cognit. Sci. **14**, 179 (1990).

[22] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, ALBERT: A lite BERT for self-supervised learning of language representations, in *Proceedings of the International Conference on Learning Representations* (ICLR, 2020).

[23] K. Yamazaki, I. Sakata, T. Konishi, and Y. Kawahara, Source Code for "Physics-inspired transformer quantum states via latent imaginary-time evolution", Zenodo, 2026, https://doi.org/10.5281/zenodo.18334851.

[24] R. Rende and L. L. Viteritti, Are queries and keys always relevant? A case study on transformer wave functions, Mach. Learn.: Sci. Technol. **6**, 010501 (2025).

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need, in *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Curran Associates, Inc., Red Hook, NY, 2017), pp. 6000–6010.

[26] G. Carleo, K. Choo, D. Hofmann, J. E. Smith, T. Westerhout, F. Alet, E. J. Davis, S. Efthymiou, I. Glasser, S.-H. Lin, M. Mauri, G. Mazzola, C. B. Mendl, E. van Nieuwenburg, O. O'Reilly, H. Théveniaut, G. Torlai, F. Vicentini, and A. Wietek, NetKet: A machine learning toolkit for many-body quantum systems, SoftwareX **10**, 100311 (2019).

[27] F. Vicentini, D. Hofmann, A. Szabó, D. Wu, C. Roth, C. Giuliani, G. Pescia, J. Nys, V. Vargas-Calderón, N. Astrakhantsev, and G. Carleo, NetKet 3: Machine learning toolbox for many-body quantum systems, SciPost Phys. Codebases 7 (2022).

[28] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, JAX: Composable transformations of Python+NumPy programs, 2018, https://github.com/jax-ml/jax.

[29] J. Heek, A. Levskaya, A. Oliver, M. Ritter, B. Rondepierre, A. Steiner, and M. van Zee, Flax: A neural network library and ecosystem for JAX, 2024, https://github.com/google/flax.

[30] D. Häfner and F. Vicentini, Mpi4jax: Zero-copy MPI communication of JAX arrays, J. Open Source Software **6**, 3419 (2021).

[31] https://netket.readthedocs.io/en/latest/tutorials/ViT-wave-function.html.

[32] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, Neural ordinary differential equations, in *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Curran Associates, Inc., Red Hook, NY, 2018), pp. 6572–6583.

[33] Y. Lu, Z. Li, D. He, Z. Sun, B. Dong, T. Qin, L. Wang, and T.-Y. Liu, Understanding and improving transformer from a multi-particle dynamic system point of view, in *Proceedings of the ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations* (ICLR, 2020).

[34] K. Heun, Neue Methode zur approximativen Integration der Differentialgleichungen einer unabhängigen Veränderlichen, Z. Math. Phys. **45**, 23 (1900).

[35] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, *et al.*, PaLM: Scaling language modeling with Pathways, J. Mach. Learn. Res. **24**, 240 (2023).

[36] M. Dehghani, J. Djolonga, B. Mustafa, P. Padlewski, J. Heek, J. Gilmer, A. P. Steiner, M. Caron, R. Geirhos, I. Alabdulmohsin, *et al.*, Scaling vision transformers to 22 billion parameters, in *Proceedings of the 40th International Conference on Machine Learning* (PMLR, 2023), Vol. 202, pp. 7480–7512.

[37] J. S. Anderson, M. Nakata, R. Igarashi, K. Fujisawa, and M. Yamashita, The second-order reduced density matrix method and the two-dimensional Hubbard model, Comput. Theor. Chem. **1003**, 22 (2013).