


# Variational Adiabatic Transport of Tensor Networks

Hyeongjin Kim<sup>1,2,\*</sup>, Matthew Fishman<sup>1b,2</sup> and Dries Sels<sup>2,3</sup>

<sup>1</sup>*Department of Physics, Boston University, Boston, Massachusetts 02215, USA*

<sup>2</sup>*Center for Computational Quantum Physics, Flatiron Institute, New York, New York 10010, USA*

<sup>3</sup>*Department of Physics, New York University, New York, New York 10003, USA*

 (Received 28 November 2023; revised 14 March 2024; accepted 23 April 2024; published 14 June 2024)

We discuss a tensor network method for constructing the adiabatic gauge potential—the generator of adiabatic transformations—as a matrix product operator, which allows us to adiabatically transport matrix product states. Adiabatic evolution of tensor networks offers a wide range of applications, of which two are explored in this paper: improving tensor network optimization and scanning phase diagrams. By efficiently transporting eigenstates to quantum criticality and performing intermediary density-matrix renormalization group (DMRG) optimizations along the way, we demonstrate that we can compute ground and low-lying excited states faster and more reliably than a standard DMRG method at or near quantum criticality. We demonstrate a simple automated step size adjustment and detection of the critical point based on the norm of the adiabatic gauge potential. Remarkably, we are able to reliably transport states through the critical point of the models we study.

DOI: [10.1103/PRXQuantum.5.020361](https://doi.org/10.1103/PRXQuantum.5.020361)

## I. INTRODUCTION

In the past few decades, the density-matrix renormalization group (DMRG) [1] has established itself as one of most prominent numerical algorithms for simulating one-dimensional quantum lattice systems (see Refs. [2–4] for reviews). It is commonly realized as a variational tensor network method that uses a matrix product state (MPS) as an ansatz for quantum wave functions and has been extensively used to compute low-energy states and dynamics of various low-dimensional quantum systems. Determining excited states provides important physical information of the system: for example, excited energy spectra can provide information on quantum phase transitions and thus help determine quantum critical points (the critical points) [5,6] or can be used to construct the low-temperature spectral function. While ground-state methods such as the DMRG are well established, excited-state methods are still in development. Even so, there have been numerous extensions of the DMRG method that find excited states accurately and efficiently. One natural extension is to compute the lowest-energy states of the Hamiltonian in multiple symmetry sectors, provided that they exist [2,7].

Another, more general, method involves adding a penalty term in the Hamiltonian to penalize, already found, lower-energy states [4]. There is a plethora of other extensions, including but not limited to Refs. [7–16]. However, even with these efforts, computing excited states remains a computationally challenging task, especially near the critical points where the spectrum is dense.

Adiabaticity plays a fundamental role in physics with numerous applications in quantum state preparation, quantum computation, and even heat engines in thermodynamics. Specifically, the adiabatic theorem states that a quantum system, prepared in an eigenstate of a Hamiltonian, stays an eigenstate of the Hamiltonian as long as the Hamiltonian is varied slowly enough. One way to formalize this adiabatic transport is through the adiabatic gauge potential (AGP) [17–19]. In short, the AGP is the generator of adiabatic eigenstate deformations and has been used for various applications, ranging from counterdiabatic driving for optimal control [20–24] and adiabatic quantum computation [25–27] to studying quantum chaos of many-body systems [28]. Here, we demonstrate that the AGP can be efficiently and reliably represented numerically as a tensor network, specifically a matrix product operator (MPO), and introduce a simple numerical optimization method for obtaining the MPO representation of the AGP. This provides a numerically efficient and general way of performing adiabatic evolution of an MPS, with straightforward generalizations to other tensor networks, by evolving with the AGP (in contrast to previous work on adiabatic evolution of tensor networks [29–33] based on

\*Corresponding author: [hkim12@bu.edu](mailto:hkim12@bu.edu)

*Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.*

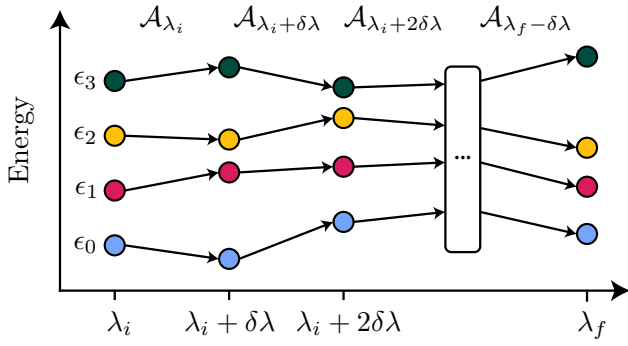


FIG. 1. Adiabatic evolution of energy eigenstates using the adiabatic gauge potential. As an illustrative example, we begin with four lowest-lying states at  $\lambda_i$ . Then, we compute the AGP  $\mathcal{A}_{\lambda_i}$  at  $\lambda_i$  to propagate these states to the next parameter point  $\lambda_i + \delta\lambda$  and continue this procedure iteratively until we reach  $\lambda_f$ . In practice, the spacing  $\delta\lambda$  can be automatically adjusted based on the norm of the AGP to add more steps where necessary, for example near quantum critical points, as discussed in Sec. III below. Optimization algorithms like the DMRG can be used to find more accurate states during the course of the adiabatic evolution.

parent Hamiltonian constructions), and thereby opens up numerous applications in tensor networks.

We demonstrate that AGP evolution of an MPS can be utilized to significantly improve the reliability and efficiency of computing low-energy excited states represented as MPSs at or near criticality, focusing in this work on a particular flavor of the excited-state DMRG method. Our approach involves computing MPS excitations far from the critical point and adiabatically evolving these states towards the critical point, as shown schematically in Fig. 1, where calculations become significantly more difficult. We show that this method finds a set of low-energy eigenstates more reliably, accurately, and efficiently at the critical point than a standard excited-state DMRG method. Relatedly, we demonstrate how adiabatic evolution of tensor networks with the AGP can be used to automatically scan the entire phase diagram of quantum many-body systems by evolving low-lying energy eigenstates through different phases. We demonstrate that the norm of the AGP can be used to automatically adjust the step size of the adiabatic evolution, with a higher norm and therefore more steps needed at or near the critical point, giving a way to automatically detect different phases and phase transitions.

## II. TENSOR NETWORK REPRESENTATION OF THE ADIABATIC GAUGE POTENTIAL

In this section, we give a brief introduction to the AGP with emphasis on its role as a generator of adiabatic eigenstate deformations and the minimization approach for its construction. While much of this has been discussed in earlier papers [17,23,34], it is provided here as context for

when we derive a method for constructing the AGP as an MPO. The latter was recently also discussed in detail in Ref. [34].

### A. Background on the adiabatic gauge potential

Suppose that we have a Hamiltonian  $\mathcal{H}(\lambda)$  with a tunable parameter  $\lambda$ . The AGP, denoted  $\mathcal{A}_\lambda$ , is a generator of adiabatic eigenstate deformations from changing  $\lambda$  via

$$i\partial_\lambda |n(\lambda)\rangle = \mathcal{A}_\lambda |n(\lambda)\rangle, \quad (1)$$

where  $|n(\lambda)\rangle$  is an energy eigenstate of  $\mathcal{H}(\lambda)$  and, in general, the AGP depends on  $\lambda$ . Using this, we can adiabatically transform the eigenstates of  $\mathcal{H}(\lambda)$  to those of  $\mathcal{H}(\lambda + d\lambda)$  with

$$|n(\lambda + d\lambda)\rangle = e^{-i\mathcal{A}_\lambda d\lambda} |n(\lambda)\rangle, \quad (2)$$

where  $d\lambda$  is an infinitesimal perturbation on  $\lambda$ .

While the AGP can be numerically computed exactly, it typically requires exact diagonalization, which is computationally infeasible for large systems. To combat this, variational approaches for approximately computing the AGP have been explored in earlier work [21–24,35], which usually involve minimizing the action  $S_\lambda$ :

$$\mathcal{A}_\lambda = \arg \min_{\mathcal{X}} S_\lambda(\mathcal{X}) \quad (3)$$

with

$$S_\lambda(\mathcal{X}) = \|\mathcal{G}_\lambda(\mathcal{X})\|^2 = \|\partial_\lambda \mathcal{H} + i[\mathcal{X}, \mathcal{H}]\|^2 \quad (4)$$

and  $\|\cdot\|^2$  the Frobenius norm. To compute  $\mathcal{A}_\lambda$ , we take the gradient of  $S_\lambda$ , i.e.,

$$\begin{aligned} \nabla_{\mathcal{X}} S_\lambda(\mathcal{X}) &= 2(\mathcal{X}\mathcal{H}^2 + \mathcal{H}^2\mathcal{X} - 2\mathcal{H}\mathcal{X}\mathcal{H}) \\ &\quad - 2i(\partial_\lambda \mathcal{H}\mathcal{H} - \mathcal{H}\partial_\lambda \mathcal{H}), \end{aligned} \quad (5)$$

and set it equal to zero. After setting  $\nabla_{\mathcal{X}} S_\lambda(\mathcal{X}) = 0$  in Eq. (5), we solve for  $\mathcal{X}$  in the linear equation

$$\mathcal{X}\mathcal{H}^2 + \mathcal{H}^2\mathcal{X} - 2\mathcal{H}\mathcal{X}\mathcal{H} = i(\partial_\lambda \mathcal{H}\mathcal{H} - \mathcal{H}\partial_\lambda \mathcal{H}), \quad (6)$$

where we have used the fact that  $\mathcal{H}^\dagger = \mathcal{H}$ . One method of solving this linear equation is to use a variational ansatz involving the commutator expansion of the form [23]

$$\mathcal{A}_\lambda^{(\ell)} = i \sum_{k=1}^{\ell} \alpha_k \underbrace{[\mathcal{H}, [\mathcal{H}, \dots [\mathcal{H}, \partial_\lambda \mathcal{H}]]]}_{2k-1}, \quad (7)$$

where the coefficients  $\{\alpha_1, \alpha_2, \dots, \alpha_k\}$  can be obtained from solving Eq. (6). Notably, the exact AGP is recovered

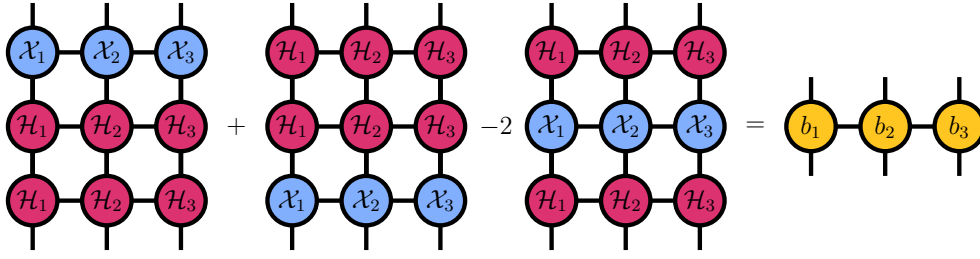


FIG. 2. Tensor network representation of Eq. (6) for solving for an approximate representation of the AGP as an MPO. For simplicity, we show an example with three sites, but the method generalizes to any number of sites (even an infinite number of sites). The tensors  $b_i$  represent the right-hand side term of Eq. (6). This diagram represents the form of the linear equation that we can solve to obtain an approximation to the AGP parametrized by  $\mathcal{X}_i$  (given  $\mathcal{H}_i$  and  $b_i$ ) using standard tensor network methods.

when  $\ell \rightarrow \infty$ . Because of the growing terms in the summation, the calculation quickly becomes inefficient for sufficiently large  $\ell$  and the convergence can be slow. Instead, we propose a tensor network method for directly computing the approximate AGP as an MPO whose local support depends on its bond dimension  $D$ . This allows us to construct the AGP for large system sizes in a well-controlled manner.

### B. Adiabatic gauge potential as a matrix product operator

To compute the approximate AGP as an MPO, we first express  $\mathcal{H}$  and  $\partial_\lambda \mathcal{H}$  as MPOs. These MPOs can be efficiently constructed with low bond dimension (that is, either constant or only grows polynomially with system size) for Hamiltonians with quasilocal interactions [36–38]. Then, we solve linear equation (6) in terms of tensor networks (see Fig. 2) to obtain  $\mathcal{X}$  that approximates  $\mathcal{A}_\lambda$ .

We recast the linear equation to the form  $Ax = b$ , where  $A$  is an MPO and  $x, b$  are MPSs acting in a doubled Hilbert space. We convert  $\mathcal{X}$  and the right-hand side of Eq. (6) to MPSs  $|\mathcal{X}\rangle$  and  $|b\rangle$ , respectively, and recast the operators acting on  $\mathcal{X}$  on the left-hand side of the equation as a superoperator acting on  $|\mathcal{X}\rangle$ :

$$(I \otimes \mathcal{H}^2 + \mathcal{H}^2 \otimes I - 2\mathcal{H} \otimes \mathcal{H}) |\mathcal{X}\rangle = |b\rangle \rightarrow A |\mathcal{X}\rangle = |b\rangle \quad (8)$$

with  $A$  an MPO acting on the doubled Hilbert space. We numerically solve this equation by using  $\mathcal{A}_\lambda^{(1)} = i\alpha_1[\mathcal{H}, \partial_\lambda \mathcal{H}]$  from Eq. (7) as an initial guess to obtain the MPS  $|\mathcal{X}\rangle \approx |\mathcal{A}_\lambda\rangle$  that approximately satisfies Eq. (8). Specifically, we use a modification of the DMRG algorithm for solving linear equations of an MPS [39,40] in order to approximately solve Eq. (8). Finally, we recast  $|\mathcal{A}_\lambda\rangle$  back as an MPO. The details of this method have been summarized in Algorithm 1.

Now, we numerically construct the AGP as an MPO and study its properties as we vary the bond dimension  $D$  of the AGP. To test the properties of Algorithm 1, we consider the transverse-field Ising model with open boundary

conditions:

$$\mathcal{H}_{\text{TFIM}} = \sum_{i=1}^{L-1} \sigma_i^x \sigma_{i+1}^x + g \sum_{i=1}^L \sigma_i^z, \quad (9)$$

$$\partial_g \mathcal{H}_{\text{TFIM}} = \sum_{i=1}^L \sigma_i^z \quad (10)$$

with  $g$  the transverse-field strength and  $L$  the system size. Notably, a phase transition occurs at  $g = 1$  ( $g \leq 1$  when  $L$  is finite): the eigenstates transition from a paramagnetic phase when  $g > 1$  to an antiferromagnetic phase when  $g < 1$ . In Fig. 3, we construct  $\mathcal{A}_g$  as an MPO using Algorithm 1 for  $L = 10, 120$  at  $g = 0.5, 1.0, 2.0$  with various bond dimensions  $D = 5, 10, 20, 40, 80, 160$  of  $\mathcal{A}_g$ . As shown in Fig. 3(a), the relative error  $|Ax - b|/|b|$  generally decreases as  $D$  increases for different realizations of  $g$  and  $L$  used. Furthermore, Fig. 3(b) shows similar convergence behaviors in  $\|\mathcal{A}_g\|$  as  $D$  increases, even though we do not directly solve to minimize the relative error of  $\|\mathcal{A}_g\|$ . Note that, even near the critical point, where the bond dimension of the AGP would formally diverge in the thermodynamic limit, we can efficiently find an approximate AGP with reasonable bond dimensions for this model. For example, on a personal computer, computing the AGP with bond dimension  $D = 40$  at  $L = 120$  took a couple of minutes. Finally, in Appendix A, we compare our variational method against the nested commutator approach in Eq. (7) for finding the AGPs as MPOs.

---



---

#### ALGORITHM 1. Computing the AGP.

---



---

**Input:**  $\mathcal{H}(\lambda), \partial \mathcal{H}(\lambda)$   
 $A \leftarrow I \otimes \mathcal{H}^2 + \mathcal{H}^2 \otimes I - 2\mathcal{H} \otimes \mathcal{H}$   
 $|b\rangle \leftarrow i[\partial_\lambda \mathcal{H} \mathcal{H} - \mathcal{H} \partial_\lambda \mathcal{H}]$   
 $|\mathcal{X}_0\rangle \leftarrow i\alpha_1 [|\mathcal{H}, \partial_\lambda \mathcal{H}\rangle] \quad \triangleright$  Use  $\ell = 1$  in Eq. (7)  
 $|\mathcal{A}_\lambda\rangle \leftarrow$  Solve for  $|\mathcal{X}\rangle$  in  $A |\mathcal{X}\rangle = |b\rangle \quad \triangleright$  Initialize with  $|\mathcal{X}_0\rangle$   
 $\mathcal{A}_\lambda \leftarrow |\mathcal{A}_\lambda\rangle \quad \triangleright$  Recast MPS as an MPO

---



---

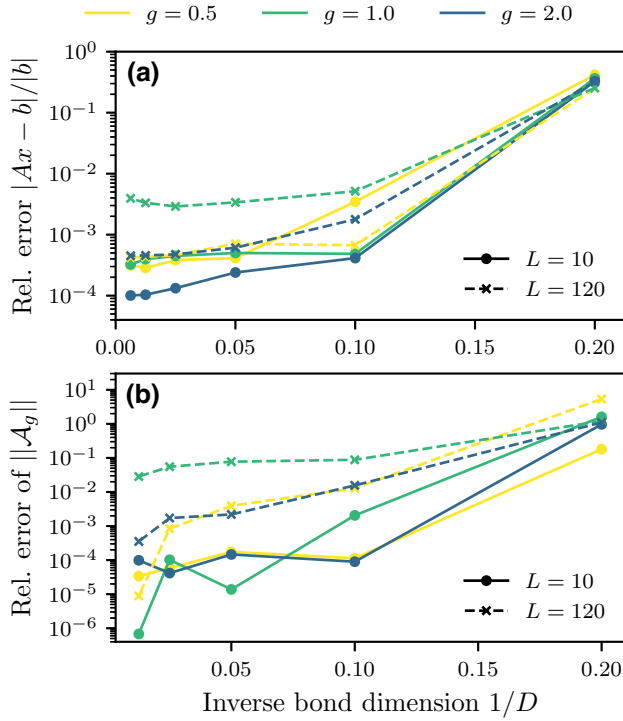


FIG. 3. Convergence properties of solving for the MPO representation of the AGP with Algorithm 1. We compute  $\mathcal{A}_g$  as an MPO for  $L = 10$  (solid line with circles) and  $L = 120$  (dashed line with crosses). Panel (a) shows the relative error  $|Ax - b|/|b|$  determined by Eq. (8) for various bond dimensions  $D$  of  $\mathcal{A}_g$ . Panel (b) shows the relative error of the norm of the MPO approximation of the AGP  $\|\mathcal{A}_g\|$  at bond dimension  $D$  compared to the MPO approximation of  $\|\mathcal{A}_g\|$  computed at  $D = 160$ .

### III. ADIABATIC EVOLUTION OF TENSOR NETWORKS WITH THE AGP

In this section, we briefly describe how adiabatic evolution of an MPS can be performed efficiently using an MPO representation of the AGP, and describe its application to preparing states for iterative optimizations as well as scanning phase diagrams.

Following Algorithm 1, we can construct the approximate AGP as an MPO for any quantum system with quasilocals Hamiltonian. In principle, we can use this AGP to adiabatically evolve any quantum state represented as an MPS to any parameter regime of the system. For example, as shown schematically in Fig. 1, we can evolve MPS excitations from  $\mathcal{H}(\lambda_i)$  to  $\mathcal{H}(\lambda_f)$  by iteratively applying the AGPs on the states throughout the trajectory. Specifically, we use the time-dependent variational principle (TDVP) method [41,42] to adiabatically evolve the matrix product states, which can be formulated as a natural generalization of the DMRG method to performing evolution of an MPS and is well suited for evolving an MPS with evolution generated by MPOs, unlike alternative MPS evolution methods like the time-evolving block decimation [43] or

time-dependent DMRG [44] methods. In practice, we find that the TDVP method works very reliably for evolving MPS eigenstates with the AGP to perform adiabatic evolution.

Now, we present two applications of using the AGP to perform adiabatic evolution of tensor networks. Firstly, we show how it can be used to improve the calculation of low-energy states with the DMRG method by combining AGP evolution with the DMRG method. We also describe how AGP evolution of an MPS can be used to automatically scan phase diagrams of low-dimensional quantum systems.

Applying the technique introduced at the start of this section, we combine our method for computing the MPO representation of the AGP (summarized in Algorithm 1) with the excited-state DMRG method with the goal of improving the computation time, reliability, and accuracy. Given that finding excited states (and even ground states) with the DMRG method can be computationally challenging around the critical point, where the states become nearly degenerate, we instead compute low-energy MPS eigenstates sufficiently far away from the critical point, for example in a gapped phase where the states are less entangled and easier to find with the DMRG method or at a point in the phase diagram where a tensor network state can be efficiently constructed from another ansatz, such as a free fermion state [45–50], and adiabatically evolve these states towards the critical point.

To be specific, suppose that the critical point is located at  $\lambda_f$  in Fig. 1. We start by computing the low-energy MPS at  $\lambda_i$  with the DMRG method, where we choose  $\lambda_i$  to be a point in the phase diagram far from the critical point where the DMRG method is reliable and fast. Then, we compute  $\mathcal{A}_{\lambda_i}$  as an MPO using Algorithm 1 and evolve the states to  $\lambda_i + \delta\lambda$  using the TDVP method. We compute a new AGP MPO at  $\lambda_i + \delta\lambda$  (perhaps using the AGP we previously found at  $\lambda_i$  as an initial guess for the optimization), and continue this procedure until we reach  $\lambda_f$ , at which point we perform the excited-state DMRG method using the propagated MPS excitations as an ansatz to find excited states at  $\lambda_f$ . This serves as a controlled method for preparing good initial states for the DMRG optimization that are already close to the true low-energy eigenstates, and therefore the DMRG method requires fewer iterations to converge and more reliably converges to states close to the true eigenstates. It is clear that the computational advantage of this method relies on the ability to (1) quickly find MPO approximations of AGPs, (2) efficiently evolve states using the MPO AGPs (for which we employ the TDVP method), and (3) maintaining accuracy of states during the evolution. As shown later in Sec. IV, we find that (1) and (2) can be easily achieved, while (3) can be done by incorporating the inexpensive excited-state DMRG method in the intermediate steps. The steps of this method combining the DMRG method with adiabatic evolution with the AGP are summarized in Algorithm 2.

## ALGORITHM 2. AGP evolution of an MPS.

---



---

**Require:**  $\lambda_f > \lambda_i$   
 $\{\epsilon_n(\lambda_i), |n(\lambda_i)\rangle\} \leftarrow$  DMRG  
 $\mathcal{A}_{\lambda_i} \leftarrow$  AGP( $\mathcal{A}_{\lambda_i}^{(1)}, \mathcal{H}(\lambda_i), \partial\mathcal{H}(\lambda_i)$ ) ▷ Refer to Alg. 1  
 $\lambda \leftarrow \lambda_i$   
**while**  $\lambda < \lambda_f$  **do**  
      $\delta\lambda \leftarrow \delta\lambda \propto 1/\|\mathcal{A}_\lambda\|$   
      $\{\epsilon_n(\lambda), |n(\lambda)\rangle\} \leftarrow \{\epsilon_n(\lambda + \delta\lambda), |n(\lambda + \delta\lambda)\rangle\}$  using TDVP  
     Optional:  $\{\epsilon_n(\lambda), |n(\lambda)\rangle\} \leftarrow$  DMRG on  $\{\epsilon_n(\lambda), |n(\lambda)\rangle\}$   
      $\mathcal{A}_\lambda \leftarrow$  AGP( $\mathcal{A}_\lambda, \mathcal{H}(\lambda + \delta\lambda), \partial\mathcal{H}(\lambda + \delta\lambda)$ )  
      $\lambda \leftarrow \lambda + \delta\lambda$   
**end while**

---



---

Notably, the AGP calculations can be performed independently from the DMRG and TDVP computations given that the AGP evolution step sizes ( $\delta\lambda$ ) can be determined by examining  $\|\mathcal{A}_\lambda\|$ : as shown in Eqs. (1) and (2), the eigenstates change the most via adiabatic evolution when  $\|\mathcal{A}_\lambda\|$  is the largest. Interestingly, this usually coincides with the critical point as the norm of the AGP is equivalent to the fidelity susceptibility, which has been extensively used to probe quantum phase transitions and peaks or even diverges at the critical point [51–54]. Hence, we use smaller step sizes near the critical point by varying  $\delta\lambda$  to be inversely proportional to  $\|\mathcal{A}_\lambda\|$ , as shown in Algorithm 2 with an illustrative example shown in Fig. 4. While variational AGP results are shown in this paper, the exact AGP results for the transverse-field Ising model have been previously shown in Refs. [55,56].

A modification of the method described above is to perform DMRG calculations at each intermediate value of the  $\lambda$  between  $\lambda_i$  and  $\lambda_f$  and thereby compute accurate

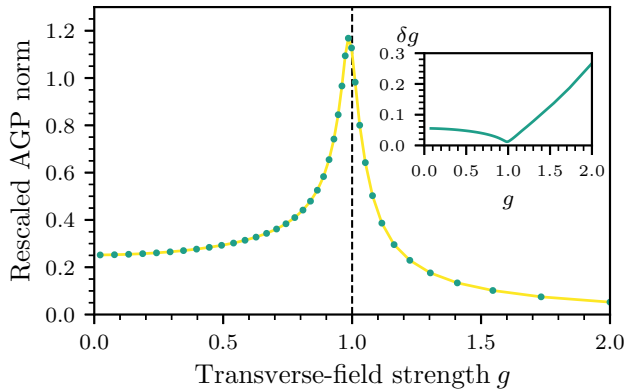


FIG. 4. Norm of the AGP for the transverse-field Ising model at system size  $L = 120$ . We scaled the norm of the AGP ( $\|\mathcal{A}_g\|$ ) by  $1/\sqrt{L2^L}$  to remove extensiveness due to  $\partial_g \mathcal{H}$ . The vertical line at  $g = 1$  signifies the critical point and the inset shows the spacing  $\delta g$  against  $g$ . The points on the plot illustrate an example of using the norm of the AGP to automatically determine an adjustable set of steps through phase space that can be used for tensor network adiabatic evolution; see the text for more details.

low-lying MPS excitations at each parameter point, where neither  $\lambda_i$  nor  $\lambda_f$  has to be at a critical point. By varying  $\delta\lambda$  to be inversely proportional to  $\|\mathcal{A}_\lambda\|$ , the algorithm can naturally focus on potential regimes of criticality and so better understand the low-energy dynamics of the system. Therefore, we can efficiently scan phase diagrams between any parameter regimes, even while crossing phase transitions or critical points. Additionally, we believe that this method for adiabatic evolution of tensor networks with the AGP can find many applications beyond those described here, some of which we discuss in the conclusion.

#### IV. APPLICATIONS OF AGP EVOLUTION OF TENSOR NETWORKS

In Sec. IV A, we demonstrate the utility of this approach for improving the reliability and performance of optimizing sets of low-energy eigenstates with the DMRG method, focusing on the transverse-field Ising model so that we can carefully benchmark our results. Then, in Sec. IV B, we use AGP evolution of tensor networks, in combination with the DMRG method, to scan the phase diagram of the transverse-field Ising model, with or without an integrability-breaking longitudinal field, by adiabatically evolving a set of approximate low-energy MPS eigenstates across the phase transition.

##### A. Improving tensor network optimization with AGP evolution

In this section, we demonstrate the utility of adiabatically evolving tensor networks with the AGP for improving the optimization of tensor networks. Many tensor network optimization methods, like the DMRG method, are iterative methods. The number of iterations needed to reach convergence, and whether or not the method converges to the global optimum, can depend on the initial state of the optimization. The convergence can be particularly sensitive to the initialization when there are other states very close to the globally optimal state, for example at or around quantum critical points, in which case the optimization can get stuck in a nearby state (i.e., a local minimum). Adiabatic evolution can provide a controlled way to prepare high-quality initial states for iterative methods like the DMRG. For example, if one can find high-quality low-energy states in a phase of the model that is relatively easy (for example, deep in a gapped phase, where the states generally have lower entanglement and the optimization is more reliable), one can then adiabatically evolve those states to a parameter regime that is more challenging and use them as initial states for an iterative optimization algorithm like the DMRG.

Hence, we focus on the problem of computing a set of low-lying energy eigenstates near a critical point. For simplicity, we focus on a particular flavor of the excited-state DMRG method that finds higher excited states by

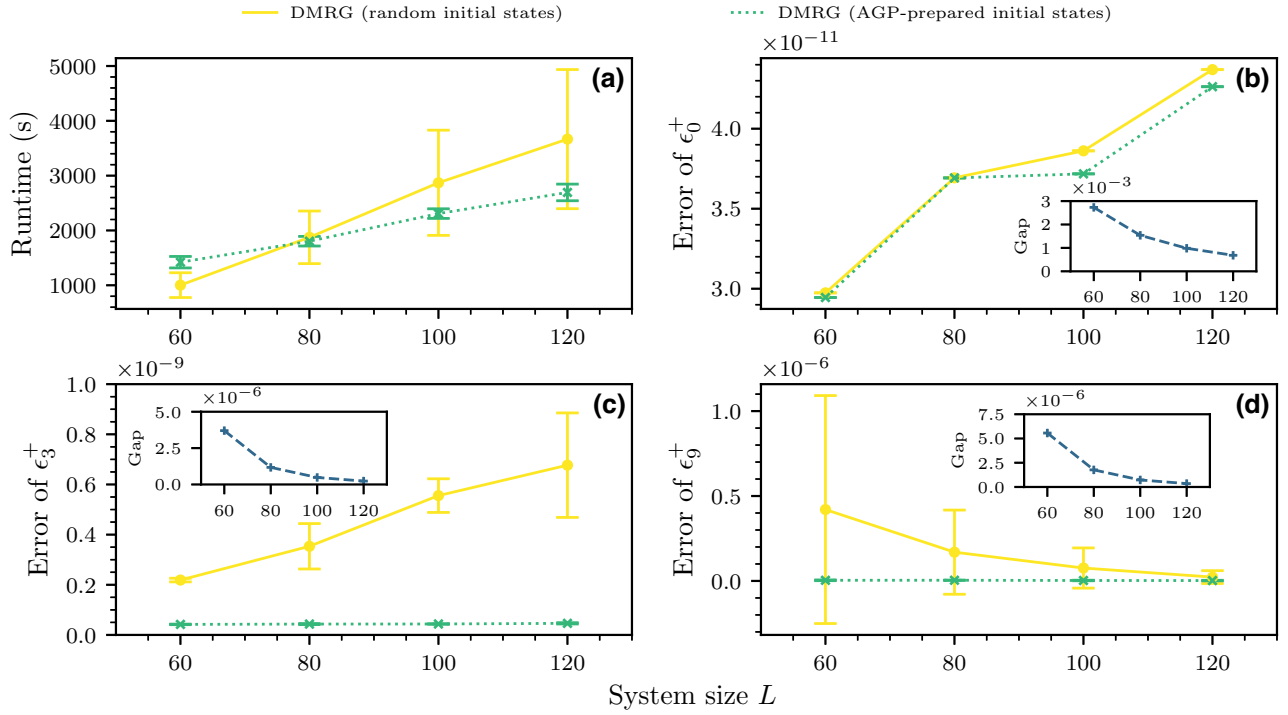


FIG. 5. Transverse-field Ising model at  $L = 120$ . We present the accuracy of our method by comparing the ten lowest-lying states (see the color bar for ordering) from transverse-field strength  $g = 2$  to  $g = 0$ , while passing the critical point at  $g = 1$ . We do this separately for even- and odd-parity sectors. In (a), we plot the relative energies in the even and odd sectors (denoted by dashed and solid lines, respectively) with respect to the ground state of the even sector ( $\epsilon_0^+$ ) against  $g$  obtained by using AGP+DMRG. In (b) and (c), we plot the error of the energy as a function of the transverse field  $g$ , for even and odd sectors, respectively, where the dashed and solid lines represent data from using AGP evolution only and AGP+DMRG, respectively.

adding weighted projectors of previously found states. One can then apply the DMRG method on the modified Hamiltonian of the form

$$\mathcal{H}' = \mathcal{H} + w \sum_{m < n} |m\rangle \langle m|, \quad (11)$$

where  $w > 0$  is the penalty weight and the  $n$ th state is the desired excited state [4]. We emphasize that there are many other excited-state DMRG techniques [7–16] and, in principle, incorporating adiabatic evolution via the AGP to any iterative ground- or excited-state DMRG method to prepare initial states could improve the convergence, including the computational time and/or precision.

We demonstrate the utility of AGP evolution for computing a set of low-lying excited states of the transverse-field Ising model [Eq. (9)] at the critical point,  $g = 1$ . This model preserves  $\sigma^z$  parity, which separates the Hamiltonian into even- and odd-parity sectors. In the following, we denote the energy eigenvalues in the even sector as  $\epsilon^+$  and in the odd sector as  $\epsilon^-$ . This model is integrable and is numerically solvable by a mapping to a free fermion model via the Jordan-Wigner transformation [57,58], allowing us to exactly compute errors in the energies of MPSs.

For our approach that combines adiabatic evolution using the AGP and DMRG optimization, we start by running DMRG optimizations to find the ten lowest-lying states in even and odd sectors, respectively, at  $g = 2$ , until the relative energy difference between DMRG sweeps is less than  $10^{-5}$ . Then, we propagate those approximate eigenstates to  $g = 1$  using AGP evolution and DMRG calculations with only a few sweeps after each step adiabatic evolution. To ensure that these intermediate DMRG runs do not dominate the computation time, they are run until the relative energy difference between sweeps is less than  $10^{-6}$ , which we find is enough to prevent accumulating large errors from taking large steps for the adiabatic evolution. Finally, we perform the DMRG optimization at  $g = 1$  using the propagated states until the relative energy difference between sweeps for each eigenstate is less than  $10^{-11}$ , which is less than the relative difference between the exact energies of the eigenstates for the system sizes considered (see the insets of Fig. 6). This ensures that we have enough precision to properly find excited states at the critical point. In all, this procedure defines the AGP-initialized DMRG method.

For comparison, we run standard DMRG optimization at transverse field  $g = 1$  using random initial wave functions. For this comparison, we used MPSs of bond dimension 10

that are generated with random circuits, which we find is a relatively good unbiased method for initializing the starting states of the optimization. Then we perform DMRG optimization to compute the ten lowest-lying states in even and odd sectors, respectively, until the relative energy difference between sweeps for each eigenstate is less than  $10^{-11}$ .

For the DMRG method with AGP-prepared initial states, we start with ten different realizations of random trial states at  $g = 2$ . On the other hand, for the DMRG method with random initial states, we simply use 20 different realizations of random initial states at  $g = 1$ . Figure 6(a) shows the average runtimes of the DMRG method using random initial states compared to the DMRG method with initial states prepared using AGP evolution (with their sample standard deviations represented by error bars) for system sizes  $L = 60, 80, 100$ , and  $120$ . As shown, on average, the AGP-initialized DMRG method performs faster than the random-initialized DMRG method for larger system sizes, as the runtime of the former increases slower with system size. Notably, the standard deviation of runtimes for the standard DMRG method drastically increases with system size, which further suggests that the AGP-initialized DMRG method is a more reliable method. Next, we systematically compare the errors of the energies  $\epsilon_0^+$ ,  $\epsilon_3^+$ , and  $\epsilon_9^+$  in Figs. 6(b)–6(d), respectively, obtained using the two methods. As shown, on average, the AGP-initialized DMRG method outperforms the random-initialized DMRG method by achieving up to 2 orders of magnitude lower errors with much smaller fluctuations. As reference, the insets of Figs. 6(b)–6(d) show the relative exact energy differences  $|\epsilon_0^+ - \epsilon_1^+|/|\epsilon_0^+|$ ,  $|\epsilon_3^+ - \epsilon_4^+|/|\epsilon_3^+|$ , and  $|\epsilon_9^+ - \epsilon_8^+|/|\epsilon_9^+|$ , respectively, for varying system sizes. While not clearly shown, the DMRG method initialized with AGP-prepared states outperforms the DMRG method initialized with random states at  $L = 120$  for the highest excited state studied in the even sector, where the errors are  $2.2(16) \times 10^{-9}$  and  $2.3(38) \times 10^{-8}$ , respectively.

### B. Scanning phase diagrams with AGP evolution of tensor networks

Understanding the landscape of the system's phases, including the locations of quantum critical points, can provide critical information on numerous physical phenomena such as high-temperature superconductivity [59–61], quantum Hall states [62,63], magnetic insulators [64], and much more. However, finding low-lying energy eigenstates (and thereby studying the phases) becomes particularly challenging near phase transitions, where the correlation length generally diverges. Hence, scanning phase diagrams is generally difficult if passing through different phases. While the standard DMRG procedure can be employed, DMRG optimization can become expensive and difficult to converge near phase transitions. To

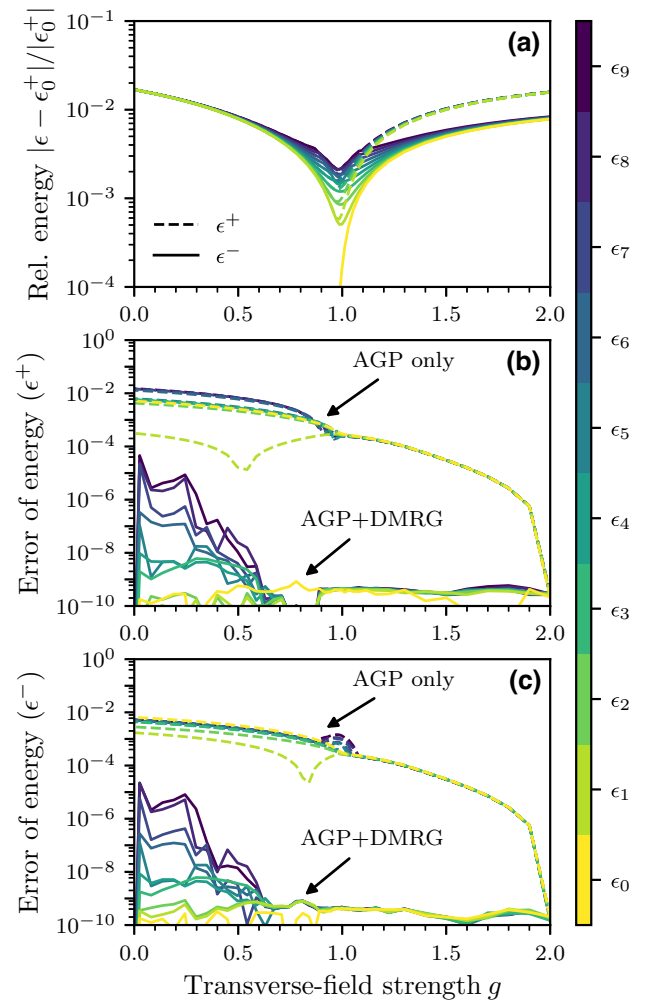


FIG. 6. Transverse-field Ising model at transverse-field strength  $g = 1$ . We compare the performance of DMRG optimization with AGP-prepared initial states (dotted lines) against DMRG optimization with random initial states (solid lines) at  $g = 1$  for varying system sizes  $L = 60, 80, 100, 120$ . Both methods find the ten lowest-lying states in even and odd sectors at  $g = 1$ , respectively, until the relative energy difference between each sweep is less than  $10^{-11}$ . The AGP-initialized DMRG method uses eight realizations of  $g$ , computed at  $g = [2.0, 1.8, 1.6, 1.4, 1.2, 1.08, 1.02, 1.0]$ . For the DMRG calculations, singular value decomposition (SVD) truncation cutoffs between  $10^{-11}$  and  $10^{-8}$  are used for the MPS truncation, while truncation cutoffs of  $10^{-6}$  to  $10^{-5}$  are used when solving for the AGP  $\mathcal{A}_\lambda$ . Panel (a) shows the average runtime in seconds against system size. For the AGP-initialized DMRG method, this includes the time taken to obtain the AGP, perform the AGP evolutions, and execute the intermediary DMRG calculations. Panels (b)–(d) show the errors of  $\epsilon_0^+$ ,  $\epsilon_3^+$ , and  $\epsilon_9^+$ , respectively. Insets in (b)–(d) show the relative exact energy differences between  $\epsilon_0^+$  and  $\epsilon_1^+$ ,  $\epsilon_3^+$  and  $\epsilon_4^+$ , and  $\epsilon_8^+$  and  $\epsilon_9^+$ , respectively, for all system sizes considered.

prevent this, we augment our AGP evolution procedure by adding accurate DMRG calculations along the trajectory (coined AGP+DMRG) to find low-lying energy

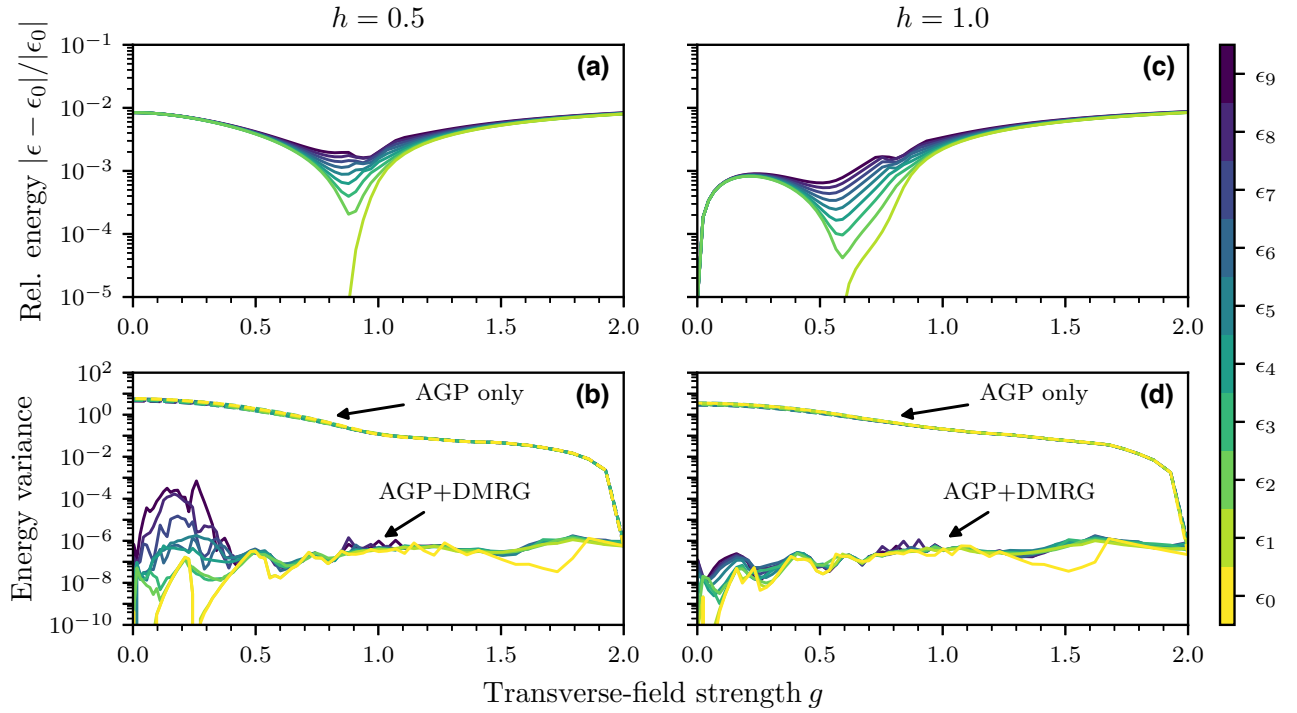


FIG. 7. Ising model with transverse and longitudinal fields at  $L = 120$  with longitudinal-field strengths  $h = 0.5, 1$ . We compute the ten lowest-lying states (see the color bar for ordering) from transverse-field strength  $g = 2$  to  $g = 0$  at (a),(b)  $h = 0.5$  and (c),(d)  $h = 1$ . (a),(c) The relative energy against the ground-state energy  $\epsilon_0$  for  $h = 0.5$  and  $h = 1$ , respectively. (b),(d) The energy variance of the states ( $\sigma_\epsilon^2 = \langle \mathcal{H}^2 \rangle - \langle \mathcal{H} \rangle^2$ ) for  $h = 0.5$  and  $h = 1.0$ , respectively, where the solid lines represent data from using AGP+DMRG, while dashed lines represent AGP evolution only.

eigenstates. This makes finding phase diagrams more computationally efficient and robust, for using AGP-initialized states as ansatzes is more reliable than random trial states. Finally, using the AGP provides automated scanning of the phase diagram by automatically using more steps near the phase transitions.

Using AGP+DMRG, we compute the low-lying energy spectra of the transverse-field Ising model and propagate them to scan its phase diagrams. In the transverse-field Ising model, we pass through the critical point at  $g = 1$ , starting at  $g = 2$  and terminating at  $g = 0$ , while propagating ten lowest-lying states in even and odd sectors, respectively, with system size  $L = 120$ . Specifically, 40  $g$  values are considered with similar spacing as shown in Fig. 4, but with  $\delta g$  capped at 0.1 instead (we find that allowing larger steps than that leads to large errors in the adiabatic evolution). In contrast to Sec. IV A, however, we perform more expensive intermediary DMRG calculations to ensure more accurate results for all values of  $g$  considered. The relative energies with respect to the ground-state energy ( $\epsilon_0^+$ ) are shown in Fig. 5. As shown, when  $g \lesssim 1$ ,  $\epsilon_i^+ \approx \epsilon_i^-$  for all integer  $i$  such that  $0 \leq i \leq 9$ . Furthermore, as shown in Figs. 5(b) and 5(c), incorporating intermediary DMRG calculations in between AGP evolutions drastically improves the results. Importantly, AGP+DMRG is able to correctly calculate the excited states even while

passing through the critical point. While not shown, AGP calculation and evolution times only contributed to roughly 10% of the total computation time with the remainder taken up by DMRG optimizations. On average, intermediary DMRG calculations at each value of  $g$  took roughly 88% less time than the DMRG calculations at starting point  $g = 2$ . It is clear that AGP-related computation is hardly the bottleneck of AGP+DMRG.

Now, we consider the nonintegrable Ising model by adding a longitudinal field with strength  $h$  to our Hamiltonian in Eq. (9):

$$\mathcal{H}_{\text{LTFIM}} = \sum_{i=1}^{L-1} \sigma_i^x \sigma_{i+1}^x + g \sum_{i=1}^L \sigma_i^z + h \sum_{i=1}^L \sigma_i^x. \quad (12)$$

Contrary to the previous model, this Hamiltonian is not exactly solvable, generically has no symmetry sectors (open boundary conditions are employed), and is generally chaotic [65]. This model has a more complicated phase structure and exhibits phase transitions at  $0 \leq g \leq 1$  when  $0 \leq h \leq 2$ . Here, we employ AGP+DMRG to study the low-lying spectrum of  $\mathcal{H}_{\text{LTFIM}}$  as we pass through phase transitions for system size  $L = 120$  at two fixed values of  $h = 0.5, 1$ . In Fig. 7, we propagate ten lowest-lying states from  $g = 2$  to  $g = 0$  at  $h = 0.5$  and  $h = 1$ . In Figs. 7(a) and 7(c), we plot the energy spectrum of



the states relative to the ground-state energy for  $h = 0.5$  and  $h = 1$ , respectively. Notably, the energy spectrum shown in Fig. 7(a) is similar to that shown in Fig. 5(c), with slight differences in the values of  $g$  at which the ground state becomes doubly degenerate. Similar to the integrable case, the error increases when  $g \lesssim 0.5$  as the excited states become highly degenerate. However, in Fig. 7(c), we have a noticeably different energy spectrum with all of the ten lowest-lying states becoming degenerate at  $g = 0$ . As we lack access to exact energies, we compute the energy variance  $\sigma_\epsilon^2 = \langle \mathcal{H}^2 \rangle - \langle \mathcal{H} \rangle^2$  for  $h = 0.5$  and  $h = 1$  in Figs. 7(b) and 7(d), respectively, where the expectation values are taken with respect to computed states  $|n\rangle$  such that  $\mathcal{H}|n\rangle \approx \epsilon_n|n\rangle$ . The variance provides a measure of how close  $|n\rangle$  is to being an eigenstate of  $\mathcal{H}$ . As shown, a variance of the order of  $10^{-6}$  is achievable using AGP+DMRG. Once again, the AGP calculation and evolution contributed to roughly 12% of the total runtime with DMRG calculations taking up the rest. Furthermore, on average, intermediary DMRG calculations at each value of  $g$  took roughly 80% less time than the DMRG calculations at  $g = 2$ .

## V. CONCLUSION

We presented a tensor network method for computing the adiabatic gauge potential, the generator of adiabatic eigenstate deformations, as a tensor network and tested the method on some paradigmatic models. For simplicity, an MPO ansatz was used, though generalizing to other tensor networks (or other ansatzes) is straightforward.

Future work and applications are abundant. Firstly, our choice of the particular excited-state DMRG method used is arbitrary and, in principle, adiabatically evolving matrix product states with the adiabatic gauge potential has the possibility to improve other excited-state (or ground-state) DMRG algorithms. More applications of the AGP in tensor networks are as follows.

- (1) Some recent excited-state DMRG methods could directly benefit from our results, such as for adiabatic transport of orbital wave functions in molecular crystals [11] and the multitarget matrix product state ansatz [16].
- (2) The AGP could be applied to other variational tensor network methods: other excited-state methods, two-dimensional tensor network state (TNS) or projected entangled pair state methods, preparation of starting states for dynamics, and more.
- (3) Continuum systems of weakly interacting electrons can be slow to converge with the DMRG method [66–70]. This convergence could be improved by preparing a free fermion state as a tensor network [45–50], evolving it with the AGP to an interacting regime, and then perform DMRG optimizations using the AGP-prepared state.

- (4) State preparation for quantum computers could be implemented using the AGP. Namely, one could prepare an MPS or TNS, adiabatically evolve it to the desired parameter regime using the AGP, and then convert it to a low-depth circuit [71,72]. This could help produce starting states for simulations with quantum chemistry [73,74], quantum field theory [75,76], variational quantum circuits [77–82], and more.
- (5) A natural extension of our work would be to consider tensor network simulations of adiabatic quantum algorithms: quantum approximate optimization algorithms [83], Grover’s algorithm [84,85], and adiabatic quantum computation [86].
- (6) The AGP as an MPO can be generalized to more Hamiltonian parameters and be used to study, for example, geodesics in parameter space [35,87] for larger system sizes.
- (7) Herbst *et al.* [88] used a sampling of states in phase space, determined by computing the residuals of high-dimensional eigenvalue problems, to map out the rest of the phase diagram. Instead, the AGP could be used to automate this sampling process to lower the computational cost.
- (8) Yarloo *et al.* [89] showed that adiabatic time evolution of highly excited states, particularly quantum scars, can be robustly performed. Using this, our work could be extended to adiabatically evolving highly excited states with the AGP.

## COMPUTING RESOURCES AND SOFTWARE PACKAGES

The code used to produce the numerical results in this paper was written using the ITensorTDVP.jl package [90]—a publicly available JULIA [91] package for solving a variety of equations in terms of MPSs, including finding eigenstates with DMRG and DMRG-X optimizations [12], performing time evolution with the TDVP method, and solving linear equations of an MPS. It is built on top of ITensors.jl [92], which provides the basic tensor operations and MPO and MPS types. For the MPS linear solver in ITensorTDVP.jl, we use the *linsolve* method from KrylovKit.jl [93] as the local solver, which implements the generalized minimal residual algorithm [94]. Calculations of the AGPs for all of the numerical computations were performed on a personal laptop, while the rest of the numerics were performed on Flatiron Institute’s Rusty computing cluster. The code to construct the AGP as an MPO is available in ITensorAGP.jl [95].

The tensor network diagrams in this paper were produced using the publicly available package GraphTikZ.jl [96], a general-purpose JULIA package for visualizing graphs, including tensor networks.

*Note added.* Recently, an article [97] appeared on arXiv that uses a similar construction to variationally prepare the AGP as an MPO. Complementary to our results, their work focuses on using the AGP to optimize quantum circuits for adiabatic quantum computing.

### ACKNOWLEDGMENTS

The authors thank Miles Stoudenmire and Anatoli Polkovnikov for helpful discussions. M.F. and D.S. are grateful for ongoing support through the Flatiron Institute, a division of the Simons Foundation. H.K. acknowledges partial support from the NSF (Grant No. DMR-2103658). D.S. is partially supported by the AFOSR (Grant No. FA9550-21-1-0236) and NSF (Grant No. OAC-2118310).

### APPENDIX A: COMPARISON WITH NESTED COMMUTATORS

In this section we compare our variational method, i.e., Algorithm 1, against the nested commutator approach from Eq. (7), for computing the AGPs as MPOs. We investigate the numerical performance and convergence of the latter method. To construct the AGP  $\mathcal{A}_\lambda^{(\ell)}$  using the nested commutator ansatz, for any  $\ell$ , we solve Eq. (6) and determine the  $\alpha_k$  by solving  $\ell$ -coupled equations. Using the nested commutator ansatz in Eq. (7), we find that

$$\frac{\partial S_\lambda}{\partial \alpha_k} = \|(\mathcal{L}^{2k}(\partial_\lambda \mathcal{H}))\partial_\lambda \mathcal{H}\|^2 + \sum_{m=1}^{\ell} \alpha_m \|(\mathcal{L}^{2k}(\partial_\lambda \mathcal{H}))(\mathcal{L}^{2m}(\partial_\lambda \mathcal{H}))\|^2, \quad (\text{A1})$$

where  $\mathcal{L}(\cdot) \equiv [\mathcal{H}, \cdot]$  is the Liouvillian superoperator and  $\mathcal{L}^k(X)$  should be interpreted as  $k$  consecutive applications of the Liouvillian on  $X$ . We set  $\partial S_\lambda / \partial \alpha_k = 0$  for all  $k$  and solve for the  $\alpha_k$ . Then, representing  $\mathcal{H}$  and  $\partial_\lambda \mathcal{H}$  as MPOs, we can construct  $\mathcal{A}_\lambda^{(\ell)}$  as an MPO. This construction elucidates the key difference between the two methods: our MPO variational method (denoted ‘‘AGP (MPO)’’ in Figs. 8 and 9) uses an MPO of finite bond dimension as the variational search space, while the nested commutator method (denoted ‘‘AGP (NC)’’ in Figs. 8 and 9) uses the Liouvillian Krylov space of  $\partial_\lambda \mathcal{H}$  up to some fixed  $\ell$  as its respective search space.

We numerically test the convergence of the nested commutator method by computing  $\mathcal{A}_\lambda^{(\ell)}$  at different values of the transverse-field strength  $g = 0.5, 1, 2$  in the transverse-field Ising model with  $L = 120$  by varying the ansatz length  $\ell = 1, 2, \dots, 8$ . In Fig. 8, we measure and plot the (a) relative errors  $|Ax - b|/|b|$  and (b) action  $S_\lambda$ . For comparison, we also show results obtained by our MPO variational method with bond dimension  $D = 20$ , as computed in Fig. 3 ( $D = 20$  was chosen for reference because

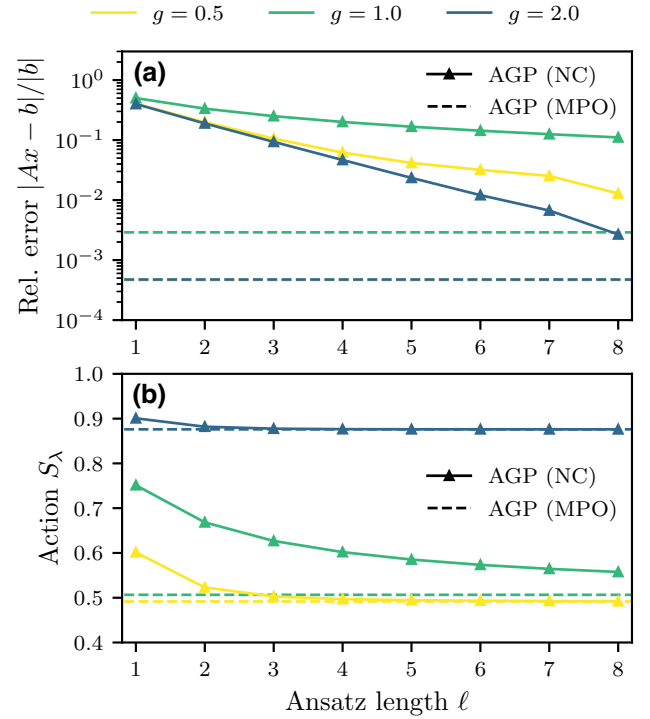


FIG. 8. Numerical convergence of relative errors and action  $S_\lambda$  of the AGP  $\mathcal{A}_\lambda^{(\ell)}$  as a function of  $\ell$ . We compute the AGP using the nested commutator method, denoted ‘‘AGP (NC)’’ and plotted with solid lines, for varying ansatz lengths  $\ell = 1, 2, \dots, 8$  and compare it to our new MPO variational method (Algorithm 1), denoted ‘‘AGP (MPO)’’ and plotted with dashed lines. We consider the transverse-field Ising model at different values of  $g$  with system size  $L = 120$ . Panel (a) shows the relative errors and (b) shows the action  $S_\lambda$ . In (a), the results for the MPO variational method at  $g = 0.5$  and  $g = 2.0$  are displayed on top of each other.

$\mathcal{A}_\lambda^{(\ell)}$  has a bond dimension of roughly 20 when  $\ell = 8$ ). We find that  $\mathcal{A}_\lambda^{(\ell)}$  and thus the nested commutators are contained in small bond dimension MPOs (between  $D = 8$  for  $\ell = 1$  and  $D = 19$  for  $\ell = 8$ ). As shown in Fig. 8, the relative error from  $\mathcal{A}_\lambda^{(\ell)}$  decreases as  $\ell$  increases, but does not decrease below that achieved using the AGP from Algorithm 1. A similar behavior is shown for  $S_\lambda$ , but we reach agreement between the two methods for large  $\ell$  when  $g = 0.5$  and  $g = 2$ . As expected, convergence is slower with  $\ell$  near criticality ( $g = 1$ ), where our new MPO variational method shows the biggest improvement over the nested commutator method. In terms of computational performance, it takes a similar amount of computational time to construct  $\mathcal{A}_\lambda^{(\ell)}$  when  $\ell = 8$  as the MPO variational method at  $D = 20$ , though the time will depend on implementation details and choices of truncation and convergence parameters. Notably, the computational time of finding  $\mathcal{A}_\lambda^{(\ell)}$  scales as  $\mathcal{O}(\ell^2)$ , which indicates that it will require much longer times to reach an  $\ell$  that gives an AGP with comparable accuracy to that from our MPO variational method, especially near criticality.

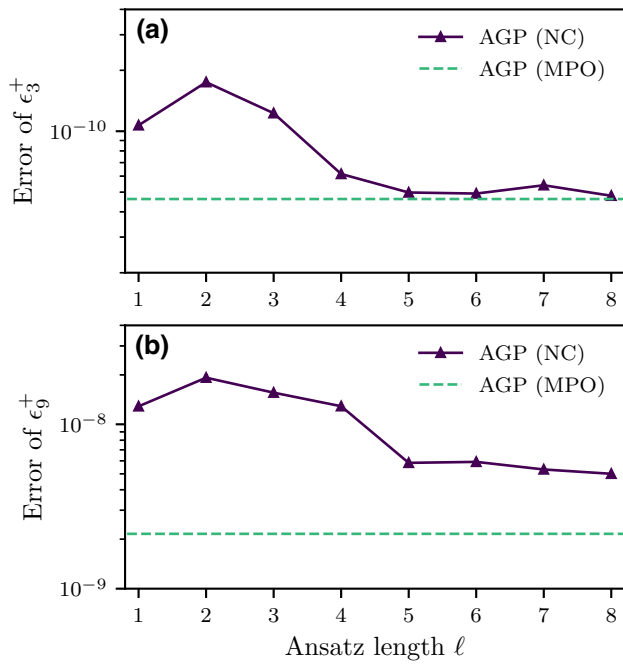


FIG. 9. Error in the energy of the excited-state DMRG method using states propagated with the AGP  $\mathcal{A}_\lambda^{(\ell)}$  as a function of  $\ell$ . We compute excited states of the transverse-field Ising model at  $g = 1$  for  $L = 120$  using the propagation method from  $g = 2$  with  $\mathcal{A}_\lambda^{(\ell)}$  for varying nested commutator ansatz lengths  $\ell = 1, 2, \dots, 8$ . We compute the errors of  $\epsilon_3^+$  in (a) and  $\epsilon_9^+$  in (b). Those obtained using the nested commutator method (NC) are shown with solid lines (with triangles for each  $\ell$ ), while those from our new method (MPO) are shown as horizontally dashed lines.

Our findings suggest that using the nested commutator approach to compute the AGP in Algorithm 2 will give less accurate energy eigenstates than using our new variational method. To show this, we perform the exact same procedure as in Fig. 6 using the nested commutator approach. We propagate the same initial states from  $g = 2$  to  $g = 1$  using  $\mathcal{A}_\lambda^{(\ell)}$  for different  $\ell$  values and perform DMRG optimizations at  $g = 1$  until the relative energy difference between each sweep is less than  $10^{-11}$  (the same criterion as before). The only difference is that we use an SVD truncation cutoff of  $10^{-8}$  (instead of the  $10^{-6}$  to  $10^{-5}$  used in our variational method) as a lower truncation cutoff is required to properly represent  $\mathcal{A}_\lambda^{(\ell)}$  as an MPO when  $\ell \geq 6$ . In Fig. 9, we consider the transverse-field Ising model with  $L = 120$  and plot the errors of (a)  $\epsilon_3^+$  and (b)  $\epsilon_9^+$ . As shown, while the two methods achieve similar errors for  $\epsilon_3^+$  at larger  $\ell$ , the nested commutator approach is unable to achieve similar errors of  $\epsilon_9^+$  compared to our new variational method.

[1] S. R. White, Density matrix formulation for quantum renormalization groups, *Phys. Rev. Lett.* **69**, 2863 (1992).

[2] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, *Ann. Phys. (N. Y.)* **326**, 96 (2011).

[3] F. Verstraete, V. Murg, and J. Cirac, Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems, *Adv. Phys.* **57**, 143 (2008).

[4] I. P. McCulloch, From density-matrix renormalization group to matrix product states, *Journal of Statistical Mechanics: Theory and Experiment* **2007**, P10014 (2007).

[5] L. Wang and A. W. Sandvik, Critical level crossings and gapless spin liquid in the square-lattice spin-1/2  $J_1 - J_2$  Heisenberg antiferromagnet, *Phys. Rev. Lett.* **121**, 107202 (2018).

[6] J. Yang, A. W. Sandvik, and L. Wang, Quantum criticality and spin liquid phase in the Shastry-Sutherland model, *Phys. Rev. B* **105**, L060409 (2022).

[7] S. R. White, Density-matrix algorithms for quantum renormalization groups, *Phys. Rev. B* **48**, 10345 (1993).

[8] S. Östlund and S. Rommer, Thermodynamic limit of density matrix renormalization, *Phys. Rev. Lett.* **75**, 3537 (1995).

[9] J. Haegeman, B. Pirvu, D. J. Weir, J. I. Cirac, T. J. Osborne, H. Verschelde, and F. Verstraete, Variational matrix product ansatz for dispersion relations, *Phys. Rev. B* **85**, 100408(R) (2012).

[10] M. C. Bañuls, K. Cichy, J. I. Cirac, and K. Jansen, The mass spectrum of the Schwinger model with matrix product states, *J. High Energy Phys.* **2013**, 1 (2013).

[11] W. Hu and G. K.-L. Chan, Excited-state geometry optimization with the density matrix renormalization group, as applied to polyenes, *J. Chem. Theory Comput.* **11**, 3000 (2015).

[12] V. Khemani, F. Pollmann, and S. L. Sondhi, Obtaining highly excited eigenstates of many-body localized Hamiltonians by the density matrix renormalization group approach, *Phys. Rev. Lett.* **116**, 247204 (2016).

[13] B. Roberts, T. Vidick, and O. I. Motrunich, Implementation of rigorous renormalization group method for ground space and low-energy states of local Hamiltonians, *Phys. Rev. B* **96**, 214203 (2017).

[14] M. Van Damme, R. Vanhove, J. Haegeman, F. Verstraete, and L. Vanderstraeten, Efficient matrix product state methods for extracting spectral information on rings and cylinders, *Phys. Rev. B* **104**, 115142 (2021).

[15] Q. Li, Y. Gao, Y.-Y. He, Y. Qi, B.-B. Chen, and W. Li, Tangent space approach for thermal tensor network simulations of the 2D Hubbard model, *Phys. Rev. Lett.* **130**, 226502 (2023).

[16] X. Li, Z. Zhou, G. Xu, R. Chi, Y. Guo, T. Liu, H. Liao, and T. Xiang, Accurate determination of low-energy eigenspectra with multi-target matrix product states, *ArXiv:2305.15868*.

[17] M. Kolodrubetz, D. Sels, P. Mehta, and A. Polkovnikov, Geometry and non-adiabatic response in quantum and classical systems, *Phys. Rep.* **697**, 1 (2017).

[18] M. V. Berry, Transitionless quantum driving, *J. Phys. A: Math. Theor.* **42**, 365303 (2009).

[19] M. Demirplak and S. A. Rice, Adiabatic population transfer with control fields, *J. Phys. Chem. A* **107**, 9937 (2003).

- [20] A. del Campo, M. M. Rams, and W. H. Zurek, Assisted finite-rate adiabatic passage across a quantum critical point: Exact solution for the quantum Ising model, *Phys. Rev. Lett.* **109**, 115703 (2012).
- [21] H. Saberi, T. Opatrny, K. Mølmer, and A. del Campo, Adiabatic tracking of quantum many-body dynamics, *Phys. Rev. A* **90**, 060301(R) (2014).
- [22] D. Sels and A. Polkovnikov, Minimizing irreversible losses in quantum systems by local counterdiabatic driving, *Proc. Natl. Acad. Sci.* **114**, E3909 (2017).
- [23] P. W. Claeys, M. Pandey, D. Sels, and A. Polkovnikov, Floquet-engineering counterdiabatic protocols in quantum many-body systems, *Phys. Rev. Lett.* **123**, 090602 (2019).
- [24] I. Čepaitė, A. Polkovnikov, A. J. Daley, and C. W. Duncan, Counterdiabatic optimized local driving, *PRX Quantum* **4**, 010312 (2023).
- [25] N. N. Hegade, K. Paul, Y. Ding, M. Sanz, F. Albarrán-Arriagada, E. Solano, and X. Chen, Shortcuts to adiabaticity in digitized adiabatic quantum computing, *Phys. Rev. Appl.* **15**, 024038 (2021).
- [26] P. Chandarana, N. N. Hegade, K. Paul, F. Albarrán-Arriagada, E. Solano, A. del Campo, and X. Chen, Digitized-counterdiabatic quantum approximate optimization algorithm, *Phys. Rev. Res.* **4**, 013141 (2022).
- [27] P. Chandarana, N. N. Hegade, I. Montalban, E. Solano, and X. Chen, Digitized counterdiabatic quantum algorithm for protein folding, *Phys. Rev. Appl.* **20**, 014024 (2023).
- [28] M. Pandey, P. W. Claeys, D. K. Campbell, A. Polkovnikov, and D. Sels, Adiabatic eigenstate deformations as a sensitive probe for quantum chaos, *Phys. Rev. X* **10**, 041017 (2020).
- [29] M. Schwarz, K. Temme, and F. Verstraete, Preparing projected entangled pair states on a quantum computer, *Phys. Rev. Lett.* **108**, 110502 (2012).
- [30] M. Schwarz, K. Temme, F. Verstraete, D. Perez-Garcia, and T. S. Cubitt, Preparing topological projected entangled pair states on a quantum computer, *Phys. Rev. A* **88**, 032321 (2013).
- [31] Y. Ge, A. Molnár, and J. I. Cirac, Rapid adiabatic preparation of injective projected entangled pair states and Gibbs states, *Phys. Rev. Lett.* **116**, 080503 (2016).
- [32] E. Cruz, F. Baccari, J. Tura, N. Schuch, and J. I. Cirac, Preparation and verification of tensor network states, *Phys. Rev. Res.* **4**, 023161 (2022).
- [33] Z.-Y. Wei, D. Malz, and J. I. Cirac, Efficient adiabatic preparation of tensor network states, *Phys. Rev. Res.* **5**, L022037 (2023).
- [34] M. Ljubotina, B. Roos, D. A. Abanin, and M. Serbyn, Optimal steering of matrix product states and quantum many-body scars, *PRX Quantum* **3**, 030343 (2022).
- [35] S. Sugiura, P. W. Claeys, A. Dymarsky, and A. Polkovnikov, Adiabatic landscape and optimal paths in ergodic systems, *Phys. Rev. Res.* **3**, 013102 (2021).
- [36] B. Pirvu, V. Murg, J. I. Cirac, and F. Verstraete, Matrix product operator representations, *New J. Phys.* **12**, 025012 (2010).
- [37] G. K.-L. Chan, A. Keselman, N. Nakatani, Z. Li, and S. R. White, Matrix product operators, matrix product states, and ab initio density matrix renormalization group algorithms, *J. Chem. Phys.* **145**, 014102 (2016).
- [38] C. Hubig, I. P. McCulloch, and U. Schollwöck, Generic construction of efficient matrix product operators, *Phys. Rev. B* **95**, 035129 (2017).
- [39] E. Jeckelmann, Dynamical density-matrix renormalization-group method, *Phys. Rev. B* **66**, 045114 (2002).
- [40] S. V. Dolgov and D. V. Savostyanov, Alternating minimal energy methods for linear systems in higher dimensions, *SIAM J. Sci. Comput.* **36**, A2248 (2014).
- [41] J. Haegeman, J. I. Cirac, T. J. Osborne, I. Pižorn, H. Verschelde, and F. Verstraete, Time-dependent variational principle for quantum lattices, *Phys. Rev. Lett.* **107**, 070601 (2011).
- [42] J. Haegeman, C. Lubich, I. Oseledets, B. Vandereycken, and F. Verstraete, Unifying time evolution and optimization with matrix product states, *Phys. Rev. B* **94**, 165116 (2016).
- [43] G. Vidal, Efficient simulation of one-dimensional quantum many-body systems, *Phys. Rev. Lett.* **93**, 040502 (2004).
- [44] S. R. White and A. E. Feiguin, Real-time evolution using the density matrix renormalization group, *Phys. Rev. Lett.* **93**, 076401 (2004).
- [45] C. V. Kraus, N. Schuch, F. Verstraete, and J. I. Cirac, Fermionic projected entangled pair states, *Phys. Rev. A* **81**, 052338 (2010).
- [46] G. Evenbly and G. Vidal, Entanglement renormalization in noninteracting fermionic systems, *Phys. Rev. B* **81**, 235102 (2010).
- [47] M. T. Fishman and S. R. White, Compression of correlation matrices and an efficient method for forming matrix product states of fermionic Gaussian states, *Phys. Rev. B* **92**, 075132 (2015).
- [48] N. Schuch and B. Bauer, Matrix product state algorithms for Gaussian fermionic states, *Phys. Rev. B* **100**, 245121 (2019).
- [49] Y.-H. Wu, L. Wang, and H.-H. Tu, Tensor network representations of parton wave functions, *Phys. Rev. Lett.* **124**, 246401 (2020).
- [50] G. Petrica, B.-X. Zheng, G. K.-L. Chan, and B. K. Clark, Finite and infinite matrix product states for Gutzwiller projected mean-field wave functions, *Phys. Rev. B* **103**, 125161 (2021).
- [51] L. Campos Venuti and P. Zanardi, Quantum critical scaling of the geometric tensors, *Phys. Rev. Lett.* **99**, 095701 (2007).
- [52] M. Kolodrubetz, V. Gritsev, and A. Polkovnikov, Classifying and measuring geometry of a quantum ground state manifold, *Phys. Rev. B* **88**, 064304 (2013).
- [53] P. Zanardi and N. Paunković, Ground state overlap and quantum phase transitions, *Phys. Rev. E* **74**, 031123 (2006).
- [54] P. Sierant, A. Maksymov, M. Kuś, and J. Zakrzewski, Fidelity susceptibility in Gaussian random ensembles, *Phys. Rev. E* **99**, 050102(R) (2019).
- [55] B. Damski and M. M. Rams, Exact results for fidelity susceptibility of the quantum Ising model: the interplay between parity, system size, and magnetic field, *J. Phys. A: Math. Theor.* **47**, 025303 (2013).
- [56] B. Damski, The quantum Ising model: Finite sums and hyperbolic functions, *Nat. Publ. Group UK London* **5**, 15779 (2015).

- [57] F. H. L. Essler and M. Fagotti, Quench dynamics and relaxation in isolated integrable quantum spin chains, *J. Stat. Mech.: Theory Exp.* **2016**, 064002 (2016).
- [58] G. B. Mbeng, A. Russomanno, and G. E. Santoro, The quantum Ising chain for beginners, *ArXiv:2009.09208*.
- [59] E. Dagotto, Correlated electrons in high-temperature superconductors, *Rev. Mod. Phys.* **66**, 763 (1994).
- [60] M. Maple, High-temperature superconductivity, *J. Magn. Magn. Mater.* **177–181**, 18 (1998).
- [61] J. Orenstein and A. Millis, Advances in the physics of high-temperature superconductivity, *Science* **288**, 468 (2000).
- [62] S. L. Sondhi, S. M. Girvin, J. P. Carini, and D. Shahar, Continuous quantum phase transitions, *Rev. Mod. Phys.* **69**, 315 (1997).
- [63] V. Piazza, V. Pellegrini, F. Beltram, W. Wegscheider, T. Jungwirth, and A. H. MacDonald, First-order phase transitions in a quantum Hall ferromagnet, *Nature* **402**, 638 (1999).
- [64] D. Bitko, T. F. Rosenbaum, and G. Aeppli, Quantum critical behavior for a model magnet, *Phys. Rev. Lett.* **77**, 940 (1996).
- [65] H. Kim, T. N. Ikeda, and D. A. Huse, Testing whether all eigenstates obey the eigenstate thermalization hypothesis, *Phys. Rev. E* **90**, 052105 (2014).
- [66] E. M. Stoudenmire, L. O. Wagner, S. R. White, and K. Burke, One-dimensional continuum electronic structure with the density-matrix renormalization group and its implications for density-functional theory, *Phys. Rev. Lett.* **109**, 056402 (2012).
- [67] M. Dolfi, B. Bauer, M. Troyer, and Z. Ristivojevic, Multigrid algorithms for tensor network states, *Phys. Rev. Lett.* **109**, 020604 (2012).
- [68] E. M. Stoudenmire and S. R. White, Sliced basis density matrix renormalization group for electronic structure, *Phys. Rev. Lett.* **119**, 046401 (2017).
- [69] R. Haghshenas, Z.-H. Cui, and G. K.-L. Chan, Numerical continuum tensor networks in two dimensions, *Phys. Rev. Res.* **3**, 023057 (2021).
- [70] S. Dutta, A. Buyskikh, A. J. Daley, and E. J. Mueller, Density matrix renormalization group for continuous quantum systems, *Phys. Rev. Lett.* **128**, 230401 (2022).
- [71] S.-J. Ran, Encoding of matrix product states into quantum circuits of one- and two-qubit gates, *Phys. Rev. A* **101**, 032310 (2020).
- [72] L. Slattery and B. K. Clark, Quantum circuits for two-dimensional isometric tensor networks, *ArXiv:2108.02792*.
- [73] I. Kassal, S. P. Jordan, P. J. Love, M. Mohseni, and A. Aspuru-Guzik, Polynomial-time quantum algorithm for the simulation of chemical dynamics, *Proc. Natl. Acad. Sci.* **105**, 18681 (2008).
- [74] B. P. Lanyon, J. D. Whitfield, G. G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, and M. Barbieri, *et al.*, Towards quantum chemistry on a quantum computer, *Nat. Chem.* **2**, 106 (2010).
- [75] A. Hamed Moosavian and S. Jordan, Faster quantum algorithm to simulate fermionic quantum field theory, *Phys. Rev. A* **98**, 012332 (2018).
- [76] J. Preskill, Simulating quantum field theory with a quantum computer, *ArXiv:1811.10085*.
- [77] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire, Towards quantum machine learning with tensor networks, *Quantum Sci. Technol.* **4**, 024001 (2019).
- [78] J.-G. Liu, Y.-H. Zhang, Y. Wan, and L. Wang, Variational quantum eigensolver with fewer qubits, *Phys. Rev. Res.* **1**, 023025 (2019).
- [79] D. A. Fedorov, B. Peng, N. Govind, and Y. Alexeev, VQE method: A short survey and recent developments, *Mater. Theory* **6**, 1 (2022).
- [80] J. Dborin, F. Barratt, V. Wimalaweera, L. Wright, and A. G. Green, Matrix product state pre-training for quantum machine learning, *Quantum Sci. Technol.* **7**, 035014 (2022).
- [81] Z. Xu, Y. Fan, H. Shang, and C. Guo, Differentiable matrix product states for simulating variational quantum computational chemistry, *ArXiv:2211.07983*.
- [82] X. Hou, Q. Li, M.-H. Yung, X. Xu, Z. Wang, C. Guo, and X. Wang, A sequentially generated variational quantum circuit with polynomial complexity, *ArXiv:2305.12856*.
- [83] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, *ArXiv:1411.4028*.
- [84] L. K. Grover, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96 (Association for Computing Machinery, New York, NY, USA, 1996), p. 212.
- [85] E. M. Stoudenmire and X. Waintal, Grover's algorithm offers no quantum advantage, *ArXiv:2303.11317*.
- [86] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution, *ArXiv:quant-ph/0001106*.
- [87] H. Kim and A. Polkovnikov, Integrability is attractive, *ArXiv:2308.09745*.
- [88] M. F. Herbst, B. Stamm, S. Wessel, and M. Rizzi, Surrogate models for quantum spin systems based on reduced-order modeling, *Phys. Rev. E* **105**, 045303 (2022).
- [89] H. Yarloo, H.-C. Zhang, and A. E. B. Nielsen, Adiabatic time evolution of highly excited states, *ArXiv:2306.13967*.
- [90] ITensorTDVP.jl, <https://github.com/ITensor/ITensorTDVP.jl> (2023).
- [91] J. Bezanson, S. Karpinski, V. B. Shah, and A. Edelman, Julia: A fast dynamic language for technical computing, *ArXiv:1209.5145*.
- [92] M. Fishman, S. R. White, and E. M. Stoudenmire, Codebase release 0.3 for ITensor, *SciPost Phys. Codebases* **4**, 4-r0.3 (2022).
- [93] KrylovKit.jl, <https://github.com/Jutho/KrylovKit.jl> (2022).
- [94] Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* **7**, 856 (1986).
- [95] ITensorAGP.jl, <https://github.com/ITensor/ITensorAGP.jl> (2023).
- [96] GraphTikZ.jl, <https://github.com/mtfishman/GraphTikZ.jl> (2023).
- [97] C. M. Keever and M. Lubasch, Towards adiabatic quantum computing using compressed quantum circuits, *PRX Quantum* **5**, 020362 (2024).