

# Double-bracket quantum algorithms for high-fidelity ground state preparation

Matteo Robbiati,<sup>1,2,\*</sup> Edoardo Pedicillo,<sup>2,3,\*</sup> Andrea Pasquale,<sup>2,3,\*</sup> Xiaoyue Li,<sup>4</sup> Andrew Wright,<sup>5</sup> Renato M. S. Farias,<sup>3,6</sup> Khanh Uyen Giang,<sup>4</sup> Jeongrak Son,<sup>4</sup> Johannes Knörzer,<sup>7</sup> Siong Thye Goh,<sup>8</sup> Jun Yong Khoo,<sup>8</sup> Nelly H.Y. Ng,<sup>4</sup> Zoë Holmes,<sup>5</sup> Stefano Carrazza,<sup>1,2,3,9</sup> and Marek Gluza<sup>4,†</sup>

<sup>1</sup>European Organization for Nuclear Research (CERN), Geneva 1211, Switzerland

<sup>2</sup>TIF Lab, Dipartimento di Fisica, Università degli Studi di Milano

<sup>3</sup>Quantum Research Center, Technology Innovation Institute, Abu Dhabi, UAE

<sup>4</sup>School of Physical and Mathematical Sciences, Nanyang Technological University, 21 Nanyang Link, 637371 Singapore, Republic of Singapore

<sup>5</sup>Institute of Physics, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

<sup>6</sup>Instituto de Física, Universidade Federal do Rio de Janeiro, P.O. Box 68528, Rio de Janeiro, Rio de Janeiro 21941-972, Brazil

<sup>7</sup>Institute for Theoretical Studies, ETH Zurich, 8092 Zurich, Switzerland

<sup>8</sup>Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A\*STAR), 1 Fusionopolis Way, #16-16 Connexis, Singapore 138632, Singapore

<sup>9</sup>INFN Sezione di Milano, Milan, Italy

Ground state preparation is a key area where quantum computers are expected to prove advantageous. Double-bracket quantum algorithms (DBQAs) have been recently proposed to diagonalize Hamiltonians and in this work we show how to use them to prepare ground states. We propose to improve an initial state preparation by adding a few steps of DBQAs. The interfaced method systematically achieves a better fidelity while significantly reducing the computational cost of the procedure. For a Heisenberg model, we compile our algorithm using CZ and single-qubit gates into circuits that match capabilities of near-term quantum devices. Moreover, we show that DBQAs can benefit from the experimental availability of increasing circuit depths. Whenever an approximate ground state can be prepared without exhausting the available circuit depth, then DBQAs can be enlisted to algorithmically seek a higher fidelity preparation.

**Introduction.**— Approximating the ground state of a target Hamiltonian is a challenging problem, which has been widely investigated in quantum and classical computing. Quantum phase estimation (QPE) [1–4] is expected to be the go-to method for computing ground state energies in the fault-tolerant era, but the circuit depths it requires are prohibitive on near-term devices. On the other hand, the variational quantum eigensolver (VQE) [5] is a popular approach when tackling the problem in the near term [6–8]. However, as a heuristic optimization algorithm over a highly non-convex landscape, VQE has no convergence guarantees and is known to experience numerous optimization barriers [9–14].

Recently, double-bracket quantum algorithms (DBQAs) have been proposed as a recursive algorithm for obtaining circuits that diagonalize a Hamiltonian [15]. In this manuscript, we investigate the potential of DBQAs for preparing ground states. Unlike quantum phase estimation, DBQAs can be implemented as near-term quantum circuits without the need for auxiliary qubits. Crucially, in contrast to VQE, these algorithms involve classical optimization only for performance improvement and can be analytically guaranteed to converge [16–18].

In this work, as schematically shown in Fig. 1, we propose a two-stage ground state preparation protocol: first, apply an existing state preparation method that uses rel-

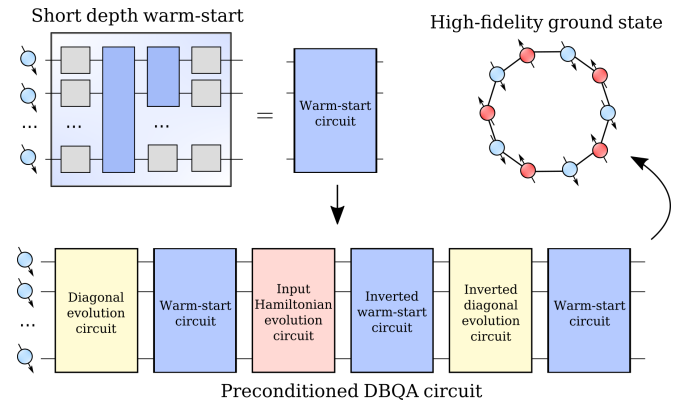


Figure 1. We propose a two-stage ground state preparation protocol: first, apply a relatively short-depth warm-start circuit; second, apply a DBQA circuit to further the ground state preparation fidelity.

atively short-depth circuits; second, apply a DBQA to further improve the state preparation. To be specific, we chose to initialize with VQE to highlight that even a limited approximation (namely one that might potentially be achieved with a short-depth circuit on near-term hardware) can be improved by DBQA. However, we stress that our results are largely agnostic to the details of the method used to find the starting state and there exist many potential candidates, including: Hartree-Fock circuits [19], quantum imaginary-time evolution [20–23] and auxiliary qubits qubitization methods [1–3].

To demonstrate the effectiveness of our method, we

\* MR, EP, and AP contributed equally

† [marekludwik.gluza@ntu.edu.sg](mailto:marekludwik.gluza@ntu.edu.sg)

perform numerical simulations of DBQA for the Heisenberg model on a lattice. We investigate the circuit depths required by DBQA depending on the quality of the initial approximation of the ground state. Considering all the investigated VQE ansätze and target Hamiltonians, we found that a single DBQA step (with circuit depth of 50-100 CZ gates per qubit) can improve the energy estimation by an order of magnitude when initialized with energy closer to the ground state than the first excited state.

**Warm-starting DBQA.**— DBQAs are quantum protocols inspired by flow equations originally used to study quantum many-body systems [24–26]. As defined in Ref. [15], DBQA circuits are obtained recursively from double-bracket iterations (DBIs),

$$\hat{A}_{k+1} = e^{s_k \hat{W}_k} \hat{A}_k e^{-s_k \hat{W}_k} \quad (1)$$

where  $\hat{A}_0$  is the DBI input and the DBI generators are commutators  $\hat{W}_k := [\hat{D}_k, \hat{A}_k]$  of  $\hat{A}_k$  with operators  $\hat{D}_k$  which should be taken diagonal to target diagonalization [15, 16].

To understand the DBI circuit ansatz, which we explain in more detail in App. A, first note that whenever  $\hat{D}_k$  and  $\hat{A}_k$  are Hermitian and  $s_k \in \mathbb{R}$ , then  $\hat{R}_k = e^{-s_k \hat{W}_k}$  are unitary. Second, the DBI initialization  $\hat{A}_0$  can be an input Hamiltonian  $\hat{H}_0$  as explored in Ref. [15]. Third, among others, a natural task for DBIs is diagonalization.

In fact, when all  $\hat{D}_k$  are equal and non-degenerate, and  $s_k$  are small enough, then  $\hat{A}_k$  asymptotically converges (exponentially) to a diagonal fixed-point  $\hat{A}^{(\infty)}$  [16–18, 27] and  $|\psi_\infty\rangle := \lim_{k \rightarrow \infty} \hat{R}_0 \hat{R}_1 \dots \hat{R}_k |0\rangle$  is an exact eigenstate of  $\hat{A}_0$ . In the general case, this could be any eigenstate and the circuit depth required to converge is exponential in  $k$  [15]. However, we will show that even a few steps can be used for substantial gains.

DBIs are parametrized by the step durations  $s_k$  and diagonal operators  $\hat{D}_k$ . These can be varied to maximize the diagonalizing effect of each DBI step as captured by a given diagonalization cost function. For example, in Ref. [15], the DBI parameters are optimized to minimize the magnitude of the off-diagonal terms of  $\hat{A}_{k+1}$  when applying Eq. (1).

In this work, we address the problem of preparing the ground state of an input Hamiltonian  $\hat{H}_0$ . To do this, we consider a warm-start unitary  $\hat{Q}$  by setting

$$\hat{A}_0 = \hat{Q}^\dagger \hat{H}_0 \hat{Q}. \quad (2)$$

This mechanism allows us to bridge the gap between an initial method for approximating the ground state (which generates  $\hat{Q}$ ) and our DBQA. Once this is done, we take as cost function the average energy of the states  $|\psi_k\rangle := \hat{Q} \hat{R}_0 \dots \hat{R}_k |0\rangle$  at each iteration under the input Hamiltonian

$$E^{(k)} := \langle \psi_k | \hat{H}_0 | \psi_k \rangle = \langle 0 | \hat{A}_{k+1} | 0 \rangle. \quad (3)$$

Thus this warm-start mechanism allows us to interface VQE and DBQA by defining a common cost function.

VQE [5] is a variational quantum computing routine that minimizes the same cost function by optimizing a parametric quantum circuit  $\hat{U}_\theta$ , i.e. varies parameters  $\theta$  to find an optimal set of parameters

$$\theta^* = \operatorname{argmin}_\theta \{ \langle 0 | \hat{U}_\theta^\dagger \hat{H}_0 \hat{U}_\theta | 0 \rangle \} \quad (4)$$

that minimize the energy. Because of VQE’s variational character, trainability and expressibility problems appear [7, 9, 10, 13, 14, 28], limiting its effectiveness. However, for DBQAs we only need an approximation of the ground state and VQE might be effective for this despite its limitations. Thus we will use VQE as a warm start for DBQA and set  $\hat{Q} = \hat{U}_{\theta^*}$  in Eq. (2). We will refer to the interfaced approach as VQExDBQA when discussing numerical results.

Ref. [15] showed that DBQAs are effective candidates for eigenstates preparation by virtue of known convergence results [16, 18] but it did not show how to specifically target ground states. As we will see, VQExDBQA consistently decreases the energy of the state. We explain this qualitatively as a combination of two factors: the spectral gap of the Hamiltonian as well as the exponential convergence of DBQA. Given the latter, as DBQA progresses  $\hat{A}_k$  is largely diagonal and consequently the DBI bracket  $[\hat{D}_k, \hat{A}_k]$  is “small”, in the sense that the resulting unitaries  $\hat{R}_k$  cannot in one step change the energy by a lot compared to the spectral gap of the model. Thus, to converge, the state must quickly tend to an eigenstate that is close in energy. When a warm-start initialization, e.g. by VQE, is closer to the ground state than other excited states then  $|\psi_\infty\rangle$  will be the ground state.

**Compiling DBQAs.**— To execute DBQA circuits, an explicit strategy for compiling the DBI unitaries  $\hat{R}_k$  as quantum circuits is necessary. This can be achieved by *group commutator iterations* (GCIs), where we replace  $\hat{R}_k$  by an approximation derived in Ref. [29],

$$\hat{V}_k = e^{ir_k \phi \hat{D}_k} e^{ir_k \hat{A}_k} e^{-ir_k(\phi+1)\hat{D}_k} e^{-ir_k(1-\phi)\hat{A}_k} e^{ir_k \hat{D}_k}, \quad (5)$$

with  $\phi = \frac{1}{2}(\sqrt{5} - 1)$ . We then define the unitary of  $k$  GCI steps as

$$\hat{U}_k = \hat{V}_0 \hat{V}_1 \dots \hat{V}_{k-1}, \quad (6)$$

such that now  $\hat{A}_k = \hat{U}_k^\dagger \hat{A}_0 \hat{U}_k$ . Following Ref. [29, Eq. (8)], for  $r_k = \sqrt{s_k}$  this gives  $\hat{V}_k^\dagger \hat{A}_k \hat{V}_k \approx \hat{R}_k^\dagger \hat{A}_k \hat{R}_k$ , with an error bounded as  $O(s_k^2)$ . See App. B for an overview of other group commutator approximations and scalings [15, 29, 30]. The GCI unitary can be recursively compiled using the identity  $e^{ir_k \hat{A}_k} = \hat{U}_k^\dagger \hat{U}_{\theta^*}^\dagger e^{ir_k \hat{H}_0} \hat{U}_{\theta^*} \hat{U}_k$  and  $\hat{U}_{k+1} = \hat{U}_k \hat{V}_k$  and by calling Hamiltonian simulation [31, 32] for each appearance of an evolution operator in  $\hat{V}_k$ , see App. B for more details. This recursive

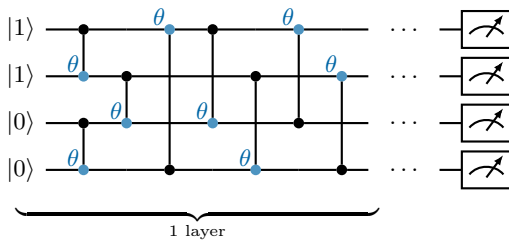


Figure 2. Hamming-weight-preserving architecture for  $L = 4$  qubits and  $S = 2$ . A single circuit layer consists of Reconfigurable Beam Splitter (RBS) gates (see Eq. (7)) connecting nearest- and next-nearest neighbors. We account for periodic boundary conditions by adding RBS gates connecting the last and first qubits. After a chosen number of layers, we add measurements in the computational basis.

unfolding for each step  $k$  leads to a circuit depth that grows exponentially with  $k$  so we use DBQA for up to three steps and after a warm-start.

**Tailoring VQE.**— We will report below quantitative results for preparing the ground state of the XXZ model, which satisfies a total-spin conservation [33]. For an even number of qubits  $L$  — as in the examples examined in this work — and total spin  $S$ , the ground state of the XXZ Hamiltonian is constrained to the half-filling subspace, i.e. a superposition of states associated to total spin  $S = L/2$ . We account for this symmetry by using a *Hamming-weight-preserving* VQE ansatz [13, 14, 34–44]. This reduces the search space to a Hilbert space of size  $\binom{L}{L/2}$ , providing a polynomial compression. Even though this compression is not ideal, i.e. exponential, fulfilling the system symmetries allows for faster training due to circuit updates being constrained to a subspace of interest [41, 45–48]. One way to construct an Hamming-weight preserving ansatz consists of creating a network of Reconfigurable Beam Splitter gates

$$\begin{array}{c} \bullet \\ | \\ \hline \\ \hline \\ \bullet \end{array} \begin{array}{c} \theta \\ | \\ \hline \\ \hline \\ \bullet \end{array} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (7)$$

Such a gate is a common building block for constructing circuits which preserve the Hamming weight of bitstrings that label computational basis states in a given superposition. In Fig. 2 we describe the circuit architecture implemented in our numerical experiments. In App. C we also include results using a hardware-efficient ansatz [49], which explores the entirety of the Hilbert space of size  $2^L$ . The increased expressibility complicates the training procedure, but interfacing with DBQA has again an advantageous effect.

**Numerical results for XXZ.**— We consider the

one-dimensional XXZ Heisenberg model,

$$\hat{H}_{\text{XXZ}} = \sum_{i=1}^L (\hat{X}_i \hat{X}_{i+1} + \hat{Y}_i \hat{Y}_{i+1} + \Delta \hat{Z}_i \hat{Z}_{i+1}), \quad (8)$$

where the subscript  $i$  indicates the Pauli operators are acting on the  $i$ -th qubit, and we use periodic boundary conditions. The XXZ model (8) is well-understood by means of Bethe ansatz [51–53], tensor networks [54] and VQE [55].

After training the VQE circuit until a target epoch, we use it as warm-start for GCI. For each GCI step, we optimize the  $\hat{D}_k$  operators which are parametrized as classical Ising models (see App. D). This allows us to compile  $e^{-it\hat{D}_k}$  with at most two layers of CZ gates and we use second-order Trotter-Suzuki decomposition [31] for a short-depth compilation of  $e^{-it\hat{H}_{\text{XXZ}}}$ , see App. E.

A detailed description of the procedure by which we obtained the results we present in this section can be found in App. D.

To visualize the advantages of VQExDBQA, in Fig. 3 we show the results for  $L = 10$  qubits obtained by executing the procedure once. We find that the DBQA halves the energy residue if applied to the early training epochs and essentially reaches the ground state when executed later in the process. More specifically, if we use three layers of the ansatz from Fig. 2, so a warm-start circuit with depth of 12 CZ gates per qubit, then following up with one GCI step yields a VQExDBQA circuit with depth 75 CZ gates per qubit. We quantify its performance through the energy approximation ratio

$$\Delta E := (\tilde{E}_0 - E_0)/E_0, \quad (9)$$

where  $\tilde{E}_0$  is the energy achieved by VQExDBQA and  $E_0$  is the true ground state energy. Fig. 3 shows an improvement by an order of magnitude from  $\Delta E \approx 1\%$  to  $\Delta E \approx 0.1\%$ . This cost function gain is statistically significant, see Tab. I where we consider 50 executions of VQExDBQA for each VQE circuit configuration. Consistently, DBQA steps improve energy estimation for all depths analyzed. Energy measurements can be translated to fidelity lower bounds [56] as described in App. D. Tab II shows that one step of VQExDBQA allowed us to get  $\mathcal{F} \geq 99.6 \pm 0.3\%$  fidelity.

By increasing the circuit depth of the chosen ansatz of Fig. 2 to 20 CZ gates per qubit, it can be trained to reach  $\Delta E \approx 0.4\%$ . In general, whenever an initialization method is improved to match the previously top performing VQExDBQA circuit, DBQA can be interfaced with that enhanced initialization  $\hat{Q}$ . Indeed, when initialized with the 20 CZ gates per qubit circuit depth with  $\Delta E \approx 0.4\%$ , just one DBQA step again gives an order of magnitude gain  $\Delta E \approx 0.03\%$ . We were not able to reach  $\Delta E \approx 0.03\%$  by training deeper VQE circuits.

The VQExDBQA circuits executed to produce Fig. 3 involve a VQE warm-start with 3 layers of the ansatz from Fig. 2, which translates to circuit depth of 12 CZ

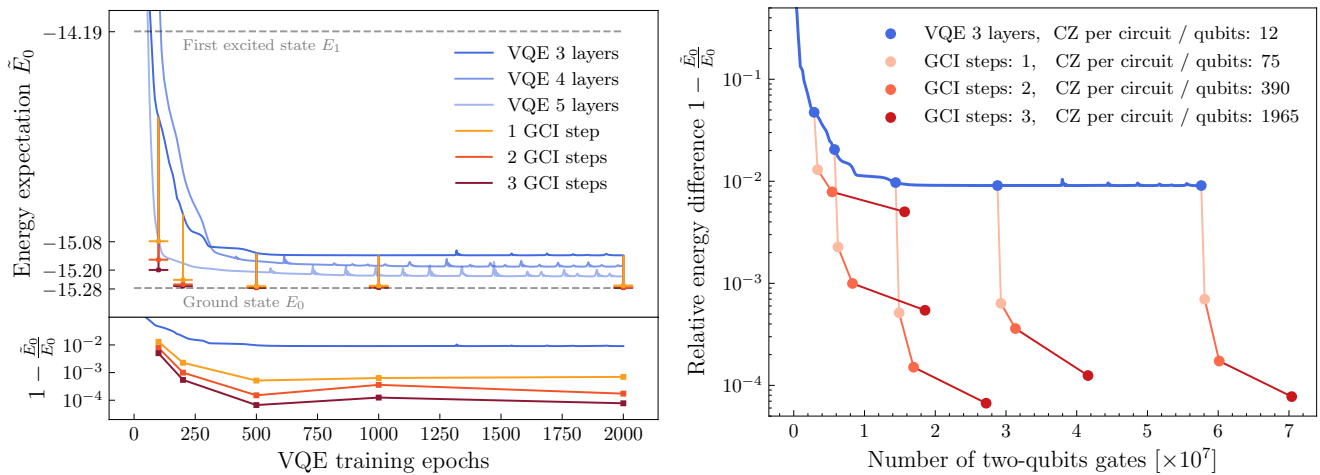


Figure 3. Visualization of the impact of VQExDBQA on cost function for a single VQE random seed, see Tab. I for statistical analysis. (*left*) Training of VQE (blue lines) for 3, 4, and 5 layers of Hamming-weight preserving ansatz (hues of blue) achieves within 500 training epochs ground state energy residue  $\Delta E \approx 1\%$ . For more epochs, the initially rapid decrease in the cost function saturates and shows marginal improvement afterward. We initialize DBQA with VQE for selected epochs  $\in \{100, 200, 500, 1000, 2000\}$  and optimize DBQA parameters with CMA-ES [50]. In the bottom panel we show the relative difference value between the achieved energy  $\tilde{E}_0$  and the true ground state energy  $E_0$ . (*right*) Token cost estimates of VQExDBQA by counting the total number of CZ gates required for the complete protocol: training the VQE until a target epoch and the optimization of DBQA.

Layers	Warm-start		1 GCI step		2 GCI steps		1 DBI step		Long VQE training	
	$1 - \tilde{E}_0/E_0$		$1 - \tilde{E}_0/E_0$		$1 - \tilde{E}_0/E_0$		$1 - \tilde{E}_0/E_0$		$1 - \tilde{E}_0/E_0$	
3	$0.012 \pm 0.004$		$0.0011 \pm 0.0007$		$0.0005 \pm 0.0004$		$0.0009 \pm 0.0006$		$0.010 \pm 0.004$	
4	$0.008 \pm 0.004$		$0.0006 \pm 0.0005$		$0.0002 \pm 0.0002$		$0.0005 \pm 0.0004$		$0.004 \pm 0.002$	
5	$0.005 \pm 0.003$		$0.0003 \pm 0.0002$		$0.0001 \pm 0.0001$		$0.0002 \pm 0.0002$		$0.003 \pm 0.002$	
	Depth	Cumulative cost	Depth	Cumulative cost	Depth	Cumulative cost	-		Depth	Cumulative cost
3	12	$1.44 \times 10^7$	75	$1.49 \times 10^7$	390	$1.69 \times 10^7$	-		12	$5.76 \times 10^7$
4	16	$2.56 \times 10^7$	95	$2.62 \times 10^7$	490	$2.88 \times 10^7$	-		16	$10.24 \times 10^7$
5	20	$4.0 \times 10^7$	115	$4.07 \times 10^7$	590	$4.38 \times 10^7$	-		20	$16.0 \times 10^7$

Table I. (*above*) Energy approximation ratio for the XXZ model of Eq. (8) with  $L = 10$  qubits, and  $\Delta = 0.5$ . The estimates with their uncertainties were calculated using the median and the median absolute deviation of a sample of results obtained by repeating the execution fifty times with different initial conditions. (*below*) Circuit depth expressed as number of CZ gates per qubit, alongside with cumulative number of CZ gates used to reach  $\tilde{E}_0$  (See App. D). Warm-start VQE approximations (500 epochs of training) are presented alongside VQExDBQA results, executed considering both compiled (GCI) and theoretical (DBI) approaches. For DBI, we compute  $\hat{R}_k$  through dense matrix representation so there is no gate count. Longer VQE training (2000 epochs) is reported in the last column of the table.

Layers	Warm-start	1 GCI step	2 GCI steps	3 GCI steps
3	$0.83 \pm 0.06$	$0.95 \pm 0.01$	$0.993 \pm 0.006$	$0.997 \pm 0.003$
4	$0.89 \pm 0.05$	$0.992 \pm 0.007$	$0.997 \pm 0.003$	$0.998 \pm 0.001$
5	$0.93 \pm 0.04$	$0.996 \pm 0.003$	$0.998 \pm 0.002$	$0.9992 \pm 0.0008$

Table II. Fidelity lower bounds [56] (see App. D) extending results presented in Tab. I.

gates per qubit and then one GCI step yields depth 75 while two steps have depth 390. The right plot in Fig. 3

shows the training cost of the algorithm quantified in terms of cumulative number of CZ gates during optimization which captures the runtime of the classical optimization. More details about the cumulative cost metric can be found in App. D. We find that the cost of training is dominated by VQE when running few DBQA steps. For tasks requiring high ground-state preparation fidelity, VQE and DBQA should be used in sequence as individually each would necessitate unnecessarily large token expenditures. This cost advantage is confirmed in Table I, where we report the gate count in the case where VQExDBQA has been compiled into circuits via the GCI

formalism (see appendix E).

Alongside with compiled GCI circuits, in the next-to-last column of Tab. I, we show VQE×DBQA results using DBI unitaries which perform similarly to GCI, highlighting the usefulness of Eq. (5).

Further investigations are presented in appendix F and appendix C, where we consider next-nearest neighbors interactions in the target Hamiltonian and a VQE architecture which does not respect XXZ symmetries so requires longer training but yields better warm-start energy in shorter depth. Even in these cases, we have verified that VQE×DBQA remains advantageous.

**Conclusions.**— Current quantum hardware allows to execute dozens, if not hundreds of CZ gates per qubit. However, this capability cannot be used to prepare high-fidelity ground state approximations due to a lack of compiling methods (VQEs encounter training limitations for large architectures; QPE algorithms need fault-tolerance). We suggest that the recently proposed DBQA approach could potentially be used to compile circuits to take advantage of the capacity of existing quantum hardware.

The fact that the combination of VQE and DBQA can achieve lower energies with shorter training times than using VQE alone could be advantageous in the near future when remote access to powerful quantum hardware will be readily available, however at a token cost for each circuit execution. Our implementation [57] within Qibo [58] provides a compilation of circuits that can be used with any QASM compatible API and so quantum computations that demonstrate experimental advances become readily available.

We note that in this work we have not explicitly taken into account the effect of shot noise or hardware noise in our numerics. Once the effects of noise are included we expect the performance of VQE to substantially degrade due to the barren plateau phenomenon. In contrast, the optimization procedures for DBI are optional in the sense that even without learning the optimal step durations  $s_k$ , certain DBIs can still be guaranteed to converge, though likely with less efficiency. Thus, while noise would be detrimental to both VQE and DBI, we expect DBI to be substantially more robust to its effects. We leave a full exploration of these effects to future work.

Nonetheless, our high-fidelity ground state preparation circuits relied on the initialization being sufficiently good, which will become increasingly difficult for larger system sizes. However, Figure 3 suggests that DBQA re-

mains advantageous for “undertrained” initializations by markedly lowering the energy even if executed for just one or two steps. If more improvement is needed then DBQA is algorithmic so at the expenditure of circuit depth it can allow to lower the energy further. Moreover, DBQA is agnostic to the initialization method and can be combined with a range of different state preparation strategies, see App. G for overview.

We expect DBQAs to remain useful in the fault-tolerant era as an initialization for QPE. Quantum dynamic programming [59] allows to reduce the circuit depth of DBQAs and can be sequenced after VQE and DBQA and before QPE. Thus, our work demonstrates that, with respect to circuit depth, DBQAs could plausibly be used to “interpolate” between the near-term and the fault-tolerant eras.

Finally, it would also be interesting to investigate more generally how DBQAs might impact the optimization of variational algorithms. This could involve adopting an approach opposite to the one developed in this work, using DBQA as a warm-start for variational routines. Such a study would fit into a research context rich in optimization techniques to interface with: Riemannian gradient flows [60], natural gradients [61], quantum imaginary time evolution [22] among the others, with the intention of considering all these tools in order to leverage their mutual benefits.

All results presented in this work are reproducible using the open-source code at Ref. [57].

**Acknowledgments.**— We thank K. Bharti, G. Crognalenti, A. Ghanesh, K. Mitarai, A. Sopena, and K. Yamamoto for useful discussions. MR is supported by the CERN Doctoral Program through the CERN Quantum Technology Initiative. JS and NN are supported by the start-up grant for Nanyang Assistant Professorship of Nanyang Technological University, Singapore. XL, KUG, and MG are supported by the Presidential Postdoctoral Fellowship of the Nanyang Technological University, Singapore. JK gratefully acknowledges support from Dr. Max Rössler, the Walter Haefner Foundation, and the ETH Zürich Foundation. ZH acknowledges support from the Sandoz Family Foundation-Monique de Meuron program for Academic Promotion. This work was further supported by the National Research Foundation Singapore, under its Quantum Engineering Programme 2.0 (National Quantum Computing Hub, NRF2021-QEP2-02-P01). STG and JYK acknowledge funding support from A\*STAR C230917003.

---

[1] Y. Dong, L. Lin, and Y. Tong, Ground-state preparation and energy estimation on early fault-tolerant quantum computers via quantum eigenvalue transformation of unitary matrices, *PRX Quantum* **3**, 10.1103/prxquantum.3.040305 (2022).

[2] L. Lin and Y. Tong, Heisenberg-limited ground-state energy estimation for early fault-tolerant quantum computers, *PRX Quantum* **3**, 010318 (2022).

[3] Y. Ge, J. Tura, and J. I. Cirac, Faster ground state preparation and high-precision ground energy estimation

- with fewer qubits, *Journal of Mathematical Physics* **60**, 022202 (2019).
- [4] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, An adaptive variational algorithm for exact molecular simulations on a quantum computer, *Nature Communications* **10**, 1 (2019).
- [5] A. Peruzzo *et al.*, A variational eigenvalue solver on a photonic quantum processor, *Nature Communications* **5**, 10.1038/ncomms5213 (2014).
- [6] K. Bharti *et al.*, Noisy intermediate-scale quantum algorithms, *Rev. Mod. Phys.* **94**, 015004 (2022).
- [7] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, *et al.*, Variational quantum algorithms, *Nature Reviews Physics* **3**, 625 (2021).
- [8] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, *et al.*, The variational quantum eigensolver: a review of methods and best practices, *Physics Reports* **986**, 1 (2022).
- [9] E. R. Anschuetz and B. T. Kiani, Beyond barren plateaus: Quantum variational algorithms are swamped with traps, *Nature Communications* **13**, 7760 (2022).
- [10] J. R. McClean *et al.*, Barren plateaus in quantum neural network training landscapes, *Nature Communications* **9**, 10.1038/s41467-018-07090-4 (2018).
- [11] L. Bittel and M. Kliesch, Training variational quantum algorithms is NP-Hard, *Phys. Rev. Lett.* **127**, 120502 (2021).
- [12] D. Stilck Franca and R. Garcia-Patron, Limitations of optimization algorithms on noisy quantum devices, *Nature Physics* **17**, 1221 (2021).
- [13] M. Larocca *et al.*, A review of barren plateaus in variational quantum computing (2024), [arXiv:2405.00781 \[quant-ph\]](https://arxiv.org/abs/2405.00781).
- [14] M. Cerezo, M. Larocca, D. García-Martín, N. L. Diaz, P. Braccia, E. Fontana, M. S. Rudolph, P. Bermejo, A. Ijaz, S. Thanasilp, *et al.*, Does provable absence of barren plateaus imply classical simulability? or, why we need to rethink variational quantum computing, [arXiv preprint arXiv:2312.09121](https://arxiv.org/abs/2312.09121) (2023).
- [15] M. Gluza, Double-bracket quantum algorithms for diagonalization, *Quantum* **8**, 1316 (2024).
- [16] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems* (Springer London, 1994).
- [17] J. B. Moore, R. E. Mahony, and U. Helmke, Numerical gradient algorithms for eigenvalue and singular value calculations, *SIAM J. Matrix Anal. Appl.* **15**, 881 (1994).
- [18] S. T. Smith, *Geometric optimization methods for adaptive filtering* (Harvard University, 1993).
- [19] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, S. Boixo, M. Broughton, B. B. Buckley, D. A. Buell, *et al.*, Hartree-fock on a superconducting qubit quantum computer, *Science* **369**, 1084 (2020).
- [20] M. Motta *et al.*, Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution, *Nature Physics* **16**, 205 (2020).
- [21] H. Nishi, T. Kosugi, and Y.-i. Matsushita, Implementation of quantum imaginary-time evolution method on NISQ devices by introducing nonlocal approximation, *npj Quantum Information* **7**, 85 (2021).
- [22] S. McArdle *et al.*, Variational ansatz-based quantum simulation of imaginary time evolution, *npj Quantum Information* **5**, 75 (2019).
- [23] N. Gomes *et al.*, Adaptive variational quantum imaginary time evolution approach for ground state preparation, *Advanced Quantum Technologies* **4**, 2100114 (2021).
- [24] F. Wegner, Flow-equations for Hamiltonians, *Annalen der Physik* **506**, 77 (1994).
- [25] S. D. Glazek and K. G. Wilson, Renormalization of Hamiltonians, *Phys. Rev. D* **48**, 5863 (1993).
- [26] F. Wegner, Flow equations and normal ordering: a survey, *Journal of Physics A: Mathematical and General* **39**, 8221 (2006).
- [27] D. Pekker, B. K. Clark, V. Oganesyan, and G. Refael, Fixed points of Wegner-Wilson flows and many-body localization, *Phys. Rev. Lett.* **119**, 075701 (2017).
- [28] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, Connecting ansatz expressibility to gradient magnitudes and barren plateaus, *PRX Quantum* **3**, 10.1103/prxquantum.3.010313 (2022).
- [29] Y.-A. Chen *et al.*, Efficient product formulas for commutators and applications to quantum simulation, *Phys. Rev. Res.* **4**, 013191 (2022).
- [30] C. M. Dawson and M. A. Nielsen, The Solovay-Kitaev algorithm, *Quantum Information & Computation* **6**, 81 (2006).
- [31] A. M. Childs and Y. Su, Nearly optimal lattice simulation by product formulas, *Phys. Rev. Lett.* **123**, 050503 (2019).
- [32] E. Campbell, Random compiler for fast hamiltonian simulation, *Phys. Rev. Lett.* **123**, 070503 (2019).
- [33] P. Nataf and F. Mila, Exact diagonalization of Heisenberg  $SU(n)$  models, *Phys. Rev. Lett.* **113**, 127204 (2014).
- [34] M. Raissi, P. Perdikaris, and G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics* **378**, 686 (2019).
- [35] G.-L. R. Anselmetti *et al.*, Local, expressive, quantum-number-preserving VQE ansätze for fermionic systems, *New Journal of Physics* **23**, 113010 (2021).
- [36] J. M. Arrazola *et al.*, Universal quantum circuits for quantum chemistry, *Quantum* **6**, 742 (2022).
- [37] S. Johri *et al.*, Nearest centroid classification on a trapped ion quantum computer, *npj Quantum Information* **7**, 122 (2021).
- [38] I. Kerenidis and A. Prakash, Quantum machine learning with subspace states (2022), [arXiv:2202.00054 \[quant-ph\]](https://arxiv.org/abs/2202.00054).
- [39] N. Jain, J. Landman, N. Mathur, and I. Kerenidis, Quantum Fourier networks for solving parametric PDEs, *Quantum Science and Technology* **9**, 035026 (2024).
- [40] M. Ragone *et al.*, Representation theory for geometric quantum machine learning (2023), [arXiv:2210.07980 \[quant-ph\]](https://arxiv.org/abs/2210.07980).
- [41] L. Monbroussou, J. Landman, A. B. Grilo, R. Kukla, and E. Kashefi, Trainability and expressivity of hamming-weight preserving quantum circuits for machine learning, [arXiv preprint arXiv:2309.15547](https://arxiv.org/abs/2309.15547) (2023).
- [42] C. Tüysüz *et al.*, Symmetry breaking in geometric quantum machine learning in the presence of noise (2024), [arXiv:2401.10293 \[quant-ph\]](https://arxiv.org/abs/2401.10293).
- [43] D. Raveh and R. I. Nepomechie, Estimating Bethe roots with VQE (2024), [arXiv:2404.18244 \[quant-ph\]](https://arxiv.org/abs/2404.18244).
- [44] R. M. S. Farias *et al.*, Quantum encoder for fixed Hamming-weight subspaces (2024), [arXiv:2405.20408 \[quant-ph\]](https://arxiv.org/abs/2405.20408).
- [45] E. A. Cherrat *et al.*, Quantum vision transformers, *Quan-*

- tum **8**, 1265 (2024).
- [46] G. Crognaletti, G. D. Bartolomeo, M. Vischi, and L. L. Viteritti, [Equivariant variational quantum eigensolver to detect phase transitions through energy level crossings](#) (2024), [arXiv:2403.07100 \[quant-ph\]](#).
- [47] O. Kiss, M. Grossi, P. Lougovski, F. Sanchez, S. Vallecorsa, and T. Papenbrock, Quantum computing of the  ${}^6\text{Li}$  nucleus via ordered unitary coupled clusters, *Phys. Rev. C* **106**, 034325 (2022).
- [48] M. Grossi, O. Kiss, F. De Luca, C. Zollo, I. Gremese, and A. Mandarino, Finite-size criticality in fully connected spin models on superconducting quantum hardware, *Phys. Rev. E* **107**, 024113 (2023).
- [49] A. Kandala *et al.*, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, *Nature* **549**, 242–246 (2017).
- [50] N. Hansen, [The CMA evolution strategy: A tutorial](#) (2023), [arXiv:1604.00772 \[cs.LG\]](#).
- [51] F. Levkovich-Maslyuk, The Bethe ansatz, *Journal of Physics A: Mathematical and Theoretical* **49**, 323004 (2016).
- [52] A. Sopena, M. H. Gordon, D. García-Martín, G. Sierra, and E. López, Algebraic Bethe circuits, *Quantum* **6**, 796 (2022).
- [53] R. Ruiz *et al.*, The Bethe ansatz as a quantum circuit, *Quantum* **8**, 1356 (2024).
- [54] V. Murg, V. E. Korepin, and F. Verstraete, Algebraic Bethe ansatz and tensor networks, *Physical Review B* **86**, 10.1103/physrevb.86.045125 (2012).
- [55] R. I. Nepomechie, [Bethe ansatz on a quantum computer?](#) (2021), [arXiv:2010.01609 \[quant-ph\]](#).
- [56] M. Cramer, M. B. Plenio, S. T. Flammia, R. Somma, D. Gross, S. D. Bartlett, O. Landon-Cardinal, D. Poulin, and Y.-K. Liu, Efficient quantum state tomography, *Nature communications* **1**, 149 (2010).
- [57] Boost VQEs with DBI, [qiboteam/boostvqe](#) (2024).
- [58] S. Efthymiou *et al.*, Qibo: a framework for quantum simulation with hardware acceleration, *Quantum Science and Technology* **7**, 015018 (2021).
- [59] J. Son, M. Gluza, R. Takagi, and N. H. Y. Ng, [Quantum dynamic programming](#) (2024), [arXiv:2403.09187 \[quant-ph\]](#).
- [60] R. Wiersema and N. Killoran, Optimizing quantum circuits with Riemannian gradient flow, *Phys. Rev. A* **107**, 062421 (2023).
- [61] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, Quantum natural gradient, *Quantum* **4**, 269 (2020).
- [62] S. Kehrein, *The flow equation approach to many-particle systems*, Vol. 217 (2006).
- [63] R. Brockett, Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems, *Linear Algebra and its Applications* **146**, 79 (1991).
- [64] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2010).
- [65] S. Carrazza, S. Efthymiou, M. Lazzarin, and A. Pasquale, An open-source modular framework for quantum computing, *Journal of Physics: Conference Series* **2438**, 012148 (2023).
- [66] S. Efthymiou, M. Lazzarin, A. Pasquale, and S. Carrazza, Quantum simulation with just-in-time compilation, *Quantum* **6**, 814 (2022).
- [67] S. Efthymiou *et al.*, Qibolab: an open-source hybrid quantum operating system, *Quantum* **8**, 1247 (2024).
- [68] A. Pasquale, A. Papaluca, R. M. S. Farias, M. Robbiati, E. Pedicillo, and S. Carrazza, [Beyond full statevector simulation with qibo](#) (2024), [arXiv:2408.00384 \[quant-ph\]](#).
- [69] E. Pedicillo, A. Candido, S. Efthymiou, H. Sargsyan, Y. P. Tan, J. Cereijo, J. Y. Khoo, A. Pasquale, M. Robbiati, and S. Carrazza, [An open-source framework for quantum hardware control](#) (2024), [arXiv:2407.21737 \[quant-ph\]](#).
- [70] A. Pasquale, S. Efthymiou, S. Ramos-Calderer, J. Wilkens, I. Roth, and S. Carrazza, [Towards an open-source framework to perform quantum calibration and characterization](#) (2024), [arXiv:2303.10397 \[quant-ph\]](#).
- [71] A. Pérez-Salinas, J. Cruz-Martinez, A. A. Alhajri, and S. Carrazza, Determining the proton content with a quantum computer, *Physical Review D* **103**, 10.1103/physrevd.103.034027 (2021).
- [72] C. Bravo-Prieto *et al.*, Style-based quantum generative adversarial networks for Monte Carlo events, *Quantum* **6**, 777 (2022).
- [73] M. Robbiati, J. M. Cruz-Martinez, and S. Carrazza, [Determining probability density functions with adiabatic quantum computing](#) (2023), [arXiv:2303.11346 \[quant-ph\]](#).
- [74] J. M. Cruz-Martinez, M. Robbiati, and S. Carrazza, Multi-variable integration with a variational quantum circuit, *Quantum Science and Technology* **9**, 035053 (2024).
- [75] S. Bordoni, D. Stanev, T. Santantonio, and S. Gigagu, Long-lived particles anomaly detection with parametrized quantum circuits, *Particles* **6**, 297–311 (2023).
- [76] K. A. Woźniak *et al.*, [Quantum anomaly detection in the latent space of proton collision events at the LHC](#) (2023), [arXiv:2301.10780 \[quant-ph\]](#).
- [77] J. J. M. de Lejarza, L. Cieri, M. Grossi, S. Vallecorsa, and G. Rodrigo, [Loop Feynman integration on a quantum computer](#) (2024), [arXiv:2401.03023 \[hep-ph\]](#).
- [78] M. Robbiati, S. Efthymiou, A. Pasquale, and S. Carrazza, [A quantum analytical Adam descent through parameter shift rule using Qibo](#) (2022), [arXiv:2210.10787 \[quant-ph\]](#).
- [79] M. Robbiati, A. Sopena, A. Papaluca, and S. Carrazza, [Real-time error mitigation for variational optimization on quantum hardware](#) (2023), [arXiv:2311.05680 \[quant-ph\]](#).
- [80] A. D’Elia *et al.*, Characterization of a transmon qubit in a 3D cavity for quantum machine learning and photon counting, *Applied Sciences* **14**, 10.3390/app14041478 (2024).
- [81] F. Chollet *et al.*, Keras, <https://keras.io> (2015).
- [82] M. Abadi *et al.*, [TensorFlow: Large-scale machine learning on heterogeneous systems](#) (2015), software available from tensorflow.org.
- [83] P. Virtanen *et al.*, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, *Nature Methods* **17**, 261 (2020).
- [84] D. P. Kingma and J. Ba, [Adam: A method for stochastic optimization](#) (2017), [arXiv:1412.6980 \[cs.LG\]](#).
- [85] M. Schuld *et al.*, Evaluating analytic gradients on quantum hardware, *Physical Review A* **99**, 10.1103/physreva.99.032331 (2019).
- [86] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, *Physical Review A* **98**, 10.1103/physreva.98.032309 (2018).

- [87] F. Vatan and C. Williams, Optimal quantum circuits for general two-qubit gates, *Phys. Rev. A* **69**, 032315 (2004).
- [88] V. V. Shende, S. S. Bullock, and I. L. Markov, Synthesis of quantum logic circuits, in *Proceedings of the 2005 Asia and South Pacific Design Automation Conference* (2005) pp. 272–275.
- [89] J. V. José, 40 years of Berezinskii-Kosterlitz-Thouless theory (2013).
- [90] S. R. White and I. Affleck, Dimerization and incommensurate spiral spin correlations in the zigzag spin chain: Analogies to the Kondo lattice, *Phys. Rev. B* **54**, 9862 (1996).
- [91] H. Westerheim, J. Chen, Z. Holmes, I. Luo, T. Nura, D. Patel, S. Rethinasamy, K. Wang, and M. M. Wilde, Dual-vqe: A quantum algorithm to lower bound the ground-state energy, arXiv preprint arXiv:2312.03083 <https://doi.org/10.48550/arXiv.2312.03083> (2023).
- [92] C. Kokail, C. Maier, R. van Bijnen, T. Brydges, M. K. Joshi, P. Jurcevic, C. A. Muschik, P. Silvi, R. Blatt, C. F. Roos, *et al.*, Self-verifying variational quantum simulation of lattice models, *Nature* **569**, 355 (2019).
- [93] K. Yamamoto, S. Duffield, Y. Kikuchi, and D. Muñoz Ramo, Demonstrating Bayesian quantum phase estimation with quantum error detection, *Phys. Rev. Res.* **6**, 013221 (2024).
- [94] K. Yamamoto, Private communication (2024).

## Appendix A: Summary of double-bracket quantum algorithms

We first list qualitative aspects together with main references and then will discuss these step by step with explicit formulas. The approach we take stems from *i*) double-bracket flows which are non-linear differential equations in the space of matrices [16, 62]. However, on a quantum computer implementing such equations exactly seems unfeasible as opposed to discretizations [15] so we turn to *ii*) iterations rather than continuous flows, see also [17, 18]. Next, the form of the iterations turns out to be *iii*) recursive and so the recursive step depends on the iterated matrix and only implicitly on the input matrix. This property allows us to “dress” the input Hamiltonian by the initialization unitary without changing the formulation. To implement quantum recursions on a quantum computer it appears that either the *iv*) depth or *v*) width of the circuit grows exponentially. For this reason we *vi*) extend the durations of steps in the iterated discretization of double-bracket flow such that each recursion step gives as much gain of the cost function as possible. Additionally, we depart from using strictly the operators that would arise in continuous flows and *vii*) allow for variational optimization, again to maximize the *viii*) cost function gain in each step.

For completeness, we give *i*) an example of a double-bracket flow, namely the Brockett flow [16, 63]: Given an input matrix  $A_0$ , we consider  $A_\ell$  to be the solution for any  $\ell \in \mathbb{R}$  of

$$\partial_\ell A_\ell = [[N, A_\ell], A_\ell]. \quad (\text{A1})$$

If the matrix  $N$  is diagonal and has a non-degenerate spectrum then  $A_\infty$  will be diagonal and the eigenvalues will be sorted as in  $N$  [16, 63] for almost all initial matrices  $A_0$ . The Euler scheme discretization of such differential equations with  $\Delta\ell \in \mathbb{R}$

$$A^{(k+1)} = A^{(k)} + \Delta\ell[[N, A^{(k)}], A^{(k)}] \quad (\text{A2})$$

is the linear expansion of the unitary recursions defined next.

Let  $\hat{H}_0$  be a Hamiltonian and assume that we are given a sequence of hermitian diagonal operators  $\hat{D}_0, \hat{D}_1, \dots$ . We define the *ii*) double-bracket iteration (DBI) as the *iii*) recursion starting with  $k = 0$  and

$$\hat{H}_{k+1} = e^{s_k \hat{W}_k} \hat{H}_k e^{-s_k \hat{W}_k}, \quad (\text{A3})$$

where  $\hat{W}_k$  is a commutator bracket

$$\hat{W}_k = \hat{D}_k \hat{H}_k - \hat{H}_k \hat{D}_k = [\hat{D}_k, \hat{H}_k]. \quad (\text{A4})$$

Next, we single out the *double-bracket unitary*

$$\hat{U}_{k+1} = \hat{U}_k e^{-s_k \hat{W}_k} \quad (\text{A5})$$

which appears in the DBI recursion equation. As we will discuss in the next section, we can approximate  $\hat{U}_k$  using group commutator formulas which reduce the quantum computation to sequences of Hamiltonian simulations. However, as we will see there via explicit gate counting, the *iv*) depth of the circuit grows exponentially. Alternatively, we can use the quantum dynamic programming [59] approach for executing the recursion, which can be carried out using only polynomial depth, albeit *v*) with an exponential circuit width as a trade-off.

We will say that

$$\hat{H}_k(s) = e^{s \hat{W}_k} \hat{H}_k(0) e^{-s \hat{W}_k} \quad (\text{A6})$$



is a *double-bracket rotation* (DBR) since it satisfies a Heisenberg equation involving two, not one, brackets

$$\partial_s \hat{H}_k(s) = [[\hat{D}_k, \hat{H}_k(0)], \hat{H}_k(s)] . \quad (\text{A7})$$

This allows us to define a greedy optimization scheme of a cost function  $f : \mathbb{C}^{2^L \times 2^L} \rightarrow \mathbb{R}_{\geq}$  by considering  $vi)$  a global minimum of how much a DBR can reduce the cost function

$$s_k = \operatorname{argmin}_{s \in \mathbb{R}} f(\hat{H}_k(s)) . \quad (\text{A8})$$

Given this optimizer conditioned implicitly on the  $\hat{D}_k$  operator we can  $vii)$  optimize these operators e.g. by repeatedly taking a starting guess, updating it and finding the minimum of  $f$ . We then set  $\hat{H}_{k+1} = \hat{H}_k(s_k)$ .

Finally, let us discuss  $viii)$  explicit forms of cost functions. For the DBR ansatz we consider the magnitude of the off-diagonal terms of  $\hat{H}_k$ . More specifically, we define  $\sigma(\hat{H}_k)$  to be the restriction of  $\hat{H}_k$  to its off-diagonal, i.e.  $\sigma(\hat{H}_k)$  and  $\hat{H}_k$  have the same off-diagonal matrix elements but the diagonal matrix elements of  $\sigma(\hat{H}_k)$  are all zero. While any matrix norm applied to  $\sigma(\hat{H}_k)$  would be a measure of the magnitude of the off-diagonal terms of  $\hat{H}_k$ , we consider the Hilbert-Schmidt norm which arises from the Hilbert-Schmidt scalar product [64]

$$\langle \hat{A}, \hat{B} \rangle_{\text{HS}} = \operatorname{tr} [\hat{A}^\dagger \hat{B}] \quad (\text{A9})$$

via  $\|\hat{A}\|_{\text{HS}}^2 = \operatorname{tr} [\hat{A}^\dagger \hat{A}]$ . For this choice we can cast the Taylor expansion into the form

$$\|\sigma(\hat{H}_k)\|_{\text{HS}}^2 - \|\sigma(\hat{H}_{k-1})\|_{\text{HS}}^2 = -2s_{k-1} \langle \hat{W}_{k-1}, [\hat{H}_{k-1}, \sigma(\hat{H}_{k-1})] \rangle_{\text{HS}} + O(s_{k-1}^2) . \quad (\text{A10})$$

This formula means that as long as we set the sign of  $\hat{D}_k$  correctly then we will get a reduction of the Hilbert-Schmidt norm of the off-diagonal restriction. When iterated we will converge towards a fixed point which will be diagonal. This diagonalization is achieved by the unitaries  $\hat{U}_k$  so when applying to computation basis states we will get an approximation to an eigenstate.

In this work, we show that DBIs allow to reduce the expected value of energy which is defined by

$$E_k = \langle \psi_k | \hat{H}_0 | \psi_k \rangle = \langle \mathbf{0} | \hat{H}_k | \mathbf{0} \rangle \quad (\text{A11})$$

where

$$|\psi_k\rangle = \hat{U}_k |\mathbf{0}\rangle . \quad (\text{A12})$$

Finding the global minimum of the energy cost function  $f_E(\psi) = \min \langle \psi | \hat{H}_0 | \psi \rangle$  results in the preparation of an eigenstate, albeit a particular one. If one wishes to prepare eigenstates then the energy fluctuation

$$\Xi(\psi) = \sqrt{\langle \psi | \hat{H}_0^2 | \psi \rangle - \langle \psi | \hat{H}_0 | \psi \rangle^2} , \quad (\text{A13})$$

is an experimentally measurable cost function. We then can define the energy fluctuation during a DBI

$$\Xi_k = \sqrt{\langle \psi_k | \hat{H}_0^2 | \psi_k \rangle - \langle \psi_k | \hat{H}_0 | \psi_k \rangle^2} = \sqrt{\langle \mathbf{0} | \hat{H}_k^2 | \mathbf{0} \rangle - \langle \mathbf{0} | \hat{H}_k | \mathbf{0} \rangle^2} . \quad (\text{A14})$$

Finally, we argue why the energy of the individual state obtained from the VQE

$$|\psi(\boldsymbol{\theta})\rangle = \hat{U}_{\boldsymbol{\theta}} |\mathbf{0}\rangle \quad (\text{A15})$$

is lowered by DBI. In principle, the DBI is designed to lower the magnitude of the off-diagonal elements in the Hamiltonian. However, by going from the Heisenberg picture in Eq. (A3) to the Schrödinger picture in Eq. (A12) we see that  $|\psi_k\rangle$  becomes gradually closer to an eigenstate thanks to the lowering of  $\sigma(\hat{H}_k)$ . While  $\sigma(\hat{H}_k)$  is lowered other eigenstate signifiers will be enhanced by proxy. For example, because the Hamiltonian  $\hat{H}_k$  becomes increasingly more diagonal, there are always fewer off-diagonal matrix elements and  $\Xi_k$  will be directly lowered in the process. For the energy, we additionally take into account that the VQE prepares a low-energy state. In the case that such a state has already an energy expectation value lower than the spectral gap, for  $\sigma(\hat{H}_k)$  to continue lowering the energy must continue to approach the ground state value - otherwise the iterated DBI state will continue having overlaps with higher energy eigenstates.

## Appendix B: Details on group commutator iterations

We will consider the notation

$$\hat{A}_{k+1} = \hat{V}_k^\dagger \hat{A}_k \hat{V}_k \quad (\text{B1})$$

for group commutations where the recursion step unitary  $\hat{V}_k$  is obtained by various approximations to the double-bracket rotation unitary

$$\hat{U}_k = e^{-s_k [\hat{D}_k, \hat{A}_k]} \quad (\text{B2})$$

The group commutator unitary for hermitian input generators  $\hat{A}, \hat{B}$  is given by

$$\hat{V}^{(\text{GC})}(\hat{A}, \hat{B}) \equiv e^{i\hat{A}} e^{i\hat{B}} e^{-i\hat{A}} e^{-i\hat{B}} . \quad (\text{B3})$$

For an iteration, we will set

$$\hat{V}_k = \hat{V}^{(\text{GC})}(\sqrt{s_k} \hat{A}_k, -\sqrt{s_k} \hat{D}_k) = e^{-s_k [\hat{D}_k, \hat{A}_k]} + O(s_k^{3/2}) . \quad (\text{B4})$$

In general  $\hat{V}^{(\text{GC})}(\hat{A}, \hat{B}) \approx e^{-[A, B]}$ . We consider this ordering because when rotating  $\hat{A}_k$  in Eq. (B1) we can equivalently use the reduced group commutator formula

$$\hat{V}^{(\text{RGC})}(\hat{A}, \hat{B}) = e^{i\hat{B}} e^{-i\hat{A}} e^{-i\hat{B}} . \quad (\text{B5})$$

Setting

$$\hat{V}_k = \hat{V}^{(\text{RGC})}(\sqrt{s_k} \hat{A}_k, -\sqrt{s_k} \hat{D}_k) \quad (\text{B6})$$

we do not obtain an approximation of the double-bracket rotation unitary  $\hat{U}_k$  but when rotating  $\hat{A}_k$  we will obtain the same  $\hat{A}_{k+1}$  because  $e^{-i\hat{A}_k} \hat{A}_k e^{i\hat{A}_k} = \hat{A}_k$ .

Finally, following Ref. [29] we consider the generalization of the group commutator to give a better approximation. For this, notice that the group commutator approximates the evolution generated by a commutator of two generators by a product of their individual evolutions. The following formula is given in Eq. (8)

$$\hat{V}^{(\text{HOPF})}(\hat{A}, \hat{B}) = e^{i\phi\hat{A}} e^{i\phi\hat{B}} e^{-i\hat{A}} e^{-i(\phi+1)\hat{B}} e^{i(1-\phi)\hat{A}} e^{i\hat{B}} \quad (\text{B7})$$

with  $\phi = \frac{1}{2}(\sqrt{5} - 1)$ . Again, this is design to approximate a double-bracket rotation  $\hat{V}^{(\text{HOPF})}(\hat{A}, \hat{B}) \approx e^{-[A, B]}$ . To use it for a group commutator iteration we set

$$\hat{V}_k = \hat{V}^{(\text{HOPF})}(\sqrt{s_k} \hat{A}_k, -\sqrt{s_k} \hat{D}_k) = e^{-s_k [\hat{D}_k, \hat{A}_k]} + O(s_k^2) . \quad (\text{B8})$$

This gives a higher-order approximation by canceling the first and third-order terms, thus retaining the leading second-order term needed for the approximation of the double-bracket rotation unitary. We will refer to it as the  $3^{\text{rd}}$  order group commutator to stress the cancellation of the third order expansion term (coincidentally, the formula involves 3 evolutions for each of the generators involved, while the regular group commutator involves 2).

Finally, in the numerics, we use again canceling and consider

$$\hat{V}^{(\text{RHOPF})}(\hat{A}, \hat{B}) = e^{i\phi\hat{B}} e^{-i\hat{A}} e^{-i(\phi+1)\hat{B}} e^{i(1-\phi)\hat{A}} e^{i\hat{B}} \quad (\text{B9})$$

as above with  $\phi = \frac{1}{2}(\sqrt{5} - 1)$ . We then set

$$\hat{V}_k = \hat{V}^{(\text{RHOPF})}(\sqrt{s_k} \hat{A}_k, -\sqrt{s_k} \hat{D}_k) = e^{-s_k [\hat{D}_k, \hat{A}_k]} + O(s_k^2) . \quad (\text{B10})$$

and benefit from 1 less query to Hamiltonian simulation.

### 1. Explicit unfolding for GCI with RGC

As above, we set

$$\hat{A}_0 = \hat{U}_{\theta^*}^\dagger \hat{H}_0 \hat{U}_{\theta^*} \quad (\text{B11})$$

so that the energy expectation value agrees with the loss function of the VQE

$$\langle \mathbf{0} | \hat{A}_0 | \mathbf{0} \rangle = \langle \psi_0(\boldsymbol{\theta}) | \hat{H}_0 | \psi_0(\boldsymbol{\theta}) \rangle . \quad (\text{B12})$$

In this section, we will make it explicit how a GCI for  $\hat{A}_k$  results in an explicit unitary that involves only the VQE preconditioning unitary  $\hat{U}_{\theta^*}$ , the input Hamiltonian evolutions  $e^{-it\hat{H}_0}$  and the diagonal evolutions  $e^{-itD_k}$ .

In the first step we will have for  $\hat{A} = r_0 \hat{H}_0$  and  $\hat{B} = -r_0 \hat{D}_0$  with  $r_0 = \sqrt{s_0}$

$$\hat{V}_1 = e^{-ir_0 \hat{D}_0} \hat{U}_{\theta^*}^\dagger e^{-ir_0 \hat{H}_0} \hat{U}_{\theta^*} e^{ir_0 \hat{D}_0} . \quad (\text{B13})$$

With this we find

$$\hat{A}_1 = \hat{V}_1^\dagger \hat{U}_{\theta^*}^\dagger \hat{H}_0 \hat{U}_{\theta^*} \hat{V}_1 . \quad (\text{B14})$$

This means that the state will be

$$\langle \mathbf{0} | \hat{A}_1 | \mathbf{0} \rangle = \langle \psi_1(\boldsymbol{\theta}) | \hat{H}_0 | \psi_1(\boldsymbol{\theta}) \rangle \quad (\text{B15})$$

where

$$|\psi_1(\boldsymbol{\theta})\rangle = \hat{U}_{\theta^*} \hat{V}_1 | \mathbf{0} \rangle . \quad (\text{B16})$$

Here we  $\hat{V}_1$  acts on the computational basis state, not the VQE state. This suggests that the DBQA step has the role of entangling the computational basis state subtly and preparing it for the action of the VQE unitary. However, let us notice that

$$e^{is_0 \hat{D}_0} | \mathbf{0} \rangle = | \mathbf{0} \rangle \quad (\text{B17})$$

whenever  $\hat{D}_0$  is traceless and otherwise we acquire an immaterial global phase. Thus more explicitly we have

$$|\psi_1(\boldsymbol{\theta})\rangle = \hat{U}_{\theta^*} e^{-is_0 \hat{D}_0} \hat{U}_{\theta^*}^\dagger e^{-is_0 \hat{H}_0} \hat{U}_{\theta^*} | \mathbf{0} \rangle \quad (\text{B18})$$

which means that the Hamiltonian evolution in the first step acts indeed on the low-lying state  $|\psi(\boldsymbol{\theta})\rangle$ .

To do one more GCI step with  $r_1 = \sqrt{s_1}$ , we set

$$\hat{V}_2 = e^{-ir_1 \hat{D}_1} \hat{V}_1^\dagger e^{-ir_1 \hat{A}_0} \hat{V}_1 e^{ir_1 \hat{D}_1} \quad (\text{B19})$$

$$= e^{-i(r_0 \hat{D}_0 + r_1 \hat{D}_1)} \hat{U}_{\theta^*}^\dagger e^{ir_0 \hat{H}_0} \hat{U}_{\theta^*} e^{ir_0 \hat{D}_0} \hat{U}_{\theta^*}^\dagger e^{-ir_1 \hat{H}_0} \hat{U}_{\theta^*} e^{-ir_0 \hat{D}_0} \hat{U}_{\theta^*}^\dagger e^{-ir_0 \hat{H}_0} \hat{U}_{\theta^*} e^{i(r_0 \hat{D}_0 + r_1 \hat{D}_1)} . \quad (\text{B20})$$

Here, it is key that there are in total 3 queries to the evolution governed by the input Hamiltonian. Using  $v_2$  we will have

$$|\psi_2(\boldsymbol{\theta})\rangle = \hat{U}_{\theta^*} \hat{V}_1 \hat{V}_2 | \mathbf{0} \rangle \quad (\text{B21})$$

and again by disregarding the global phase the implementation of the last diagonal circuit  $e^{ir_0 \hat{D}_0 + ir_1 \hat{D}_1}$  can be omitted in  $\hat{V}_2$  (but not in  $\hat{V}_1$ ). Thus, in total 2 steps of GCI involve 4 queries to  $e^{-it\hat{H}_0}$ .

We will say that Eq. (B20) is the unfolded form of the second GCI step. We repeat this unfolding of the recursion for further steps until all unitaries appearing in the sequence are generated either by  $\hat{D}_k$  or  $\hat{H}_0$ .

In the numerical results, we do not use the first but rather the third-order higher-order product formula approximating the double-bracket rotation. The unfolding proceeds in analogy just involving more queries to the input Hamiltonian simulation.

### Appendix C: VQE using Hardware-Efficient ansatz

In this section, we show how DBQAxVQE performs with a hardware-efficient ansatz targeting the XXZ Hamiltonian. This ansatz is in principle more expressive of the Hamming-weight preserving ansatz of Fig. 2, in the sense it could cover the whole Hilbert space if the chosen circuit is expressive enough. From one side, this means the optimal solution to our problem lies in the search space of our algorithm but, on the other hand, the training is more difficult because no prior knowledge is exploited as it is when choosing the Hamming-weight preserving ansatz. Practically, we are also exploring regions of the Hilbert space that don't contain the target solution. Comparing the gates composing our chosen HEA circuit with the one introduced in Fig. 2, we note the former presents more parameters but a lower number of CZ. The choice of the ansatz must take into account the hardware availability and calibration accuracy.

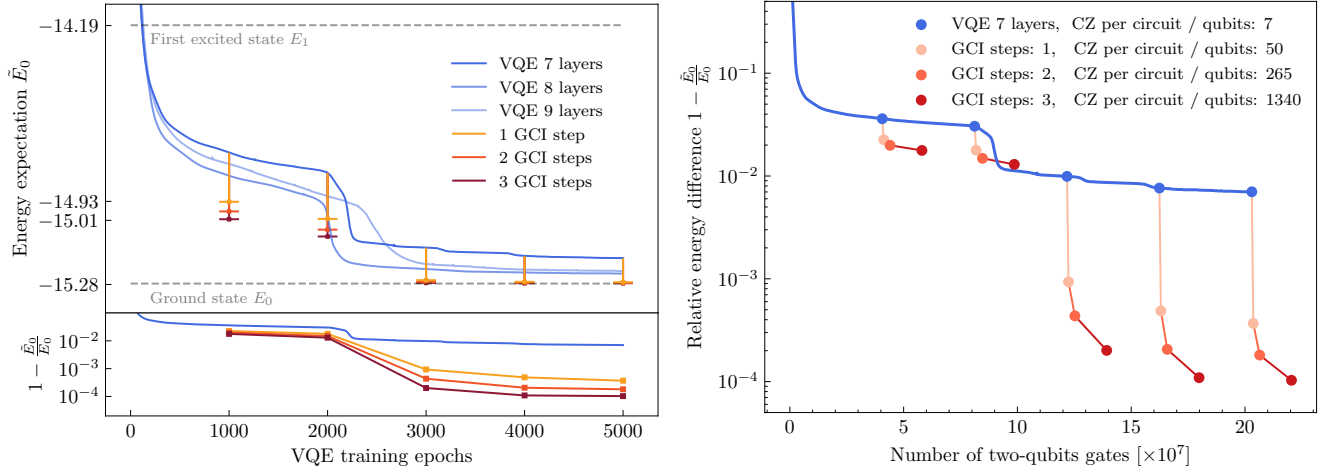


Figure 4. One example of VQExDBQA for XXZ using a hardware efficient ansatz and obtained fixing the simulation random seed; the image is intended to provide qualitative information about the impact of VQExDBQA. A more robust study of the performance is presented in Tab. I. (left) Training of VQE (blue lines) for 7, 8, and 9 layers (hues of blue) achieve within 500 training epochs ground state energy residue of about 1%. We initialize DBQA with VQE for selected epochs  $\in [1000, 2000, 3000, 4000, 5000]$  where we apply a DBQA optimized in its parameters with CMA-ES [50]. (right) Token cost estimates of VQExDBQA by counting the total number of two-qubit gates required to execute the complete protocol: training the VQE until a target epoch and then optimizing and applying the DBQA.

Layers	Warm-start		1 GCI step		2 GCI steps		Long VQE training	
	$1 - \tilde{E}_0/E_0$		$1 - \tilde{E}_0/E_0$		$1 - \tilde{E}_0/E_0$		$1 - \tilde{E}_0/E_0$	
7	$0.023 \pm 0.002$		$0.013 \pm 0.0007$		$0.011 \pm 0.0003$		$0.006 \pm 0.002$	
8	$0.005 \pm 0.003$		$0.0003 \pm 0.0002$		$0.00013 \pm 0.00007$		$0.004 \pm 0.002$	
9	$0.005 \pm 0.003$		$0.0003 \pm 0.0002$		$0.00018 \pm 0.00006$		$0.0030 \pm 0.0003$	
	Depth	Cumulative cost	Depth	Cumulative cost	Depth	Cumulative cost	Depth	Cumulative cost
7	7	$12.18 \times 10^7$	50	$12.25 \times 10^7$	265	$12.52 \times 10^7$	7	$20.03 \times 10^7$
8	8	$15.84 \times 10^7$	55	$15.91 \times 10^7$	290	$16.21 \times 10^7$	8	$26.4 \times 10^7$
9	9	$19.98 \times 10^7$	60	$20.06 \times 10^7$	315	$20.39 \times 10^7$	9	$33.3 \times 10^7$

Table III. (above) Energy approximation ratio for the XXZ model of Eq. (8) with  $L = 10$  qubits, and  $\Delta = 0.5$ . The estimates with their uncertainties were calculated using the median and the median absolute deviation of a sample of results obtained by repeating the execution fifty times with different initial conditions. (below) Circuit depth expressed as number of CZ gates per qubit, alongside with cumulative number of CZ gates used to reach  $\tilde{E}_0$  (See App. D). Warm-start VQE approximations (3000 epochs of training) are presented alongside VQExDBQA results, executed considering compiled GCI circuits. Longer VQE training (5000 epochs) is reported in the last column of the table.

We proceed with the same strategy presented in the main text, but considering  $N = 5$  trainings per configuration. In Fig. 4 we show an example of VQExDBQA execution with fixed random seed. After an initially rapid cost function decrease, the VQE training saturates to essentially marginal improvements regardless the number of layers

considered. On the other hand, we find that the DBQA halves the energy residue if applied to very early training epochs (1000, 2000) and essentially reaches the ground state when executed later in the process.

The lower left plot shows that DBQA initialized in the first training plateau is in the basin of attraction for the ground-state fixed point where convergence is exponentially fast in the number of steps.

The plot on the right shows the cumulative number of CZ gates required to execute the VQExDBQA protocol together with the circuit depth, expressed as number of CZ gates per qubit. We find that the training cost is dominated by VQE if considering a couple of DBQA steps. On the other hand, the DBQA training optimizes the energy more with fewer queries. For tasks requiring high ground-state preparation fidelity, both methods should be used in sequence as individually each would necessitate unnecessarily large token expenditures. A more detailed analysis of the performances is presented in Tab. III and Tab. IV, where the approximation accuracy is respectively quantified in terms of relative differences of Eq. (D1) and fidelity lower bound of Eq. (D2).

Layers	Warm-start	1 GCI step	2 GCI steps	Long VQE training
7	$0.67 \pm 0.03$	$0.82 \pm 0.05$	$0.85 \pm 0.05$	$0.92 \pm 0.03$
8	$0.93 \pm 0.04$	$0.995 \pm 0.002$	$0.998 \pm 0.001$	$0.94 \pm 0.03$
9	$0.93 \pm 0.02$	$0.996 \pm 0.001$	$0.998 \pm 0.001$	$0.96 \pm 0.004$

Table IV. Fidelity lower bound  $\mathcal{F}$  (see Eq. (D2)) for the XXZ model of Eq. (8) with  $L = 10$  qubits, and  $\Delta = 0.5$  extending results of Tab. III.

## Appendix D: Details about numerical simulations

In this section, we discuss the choices we made in carrying out the simulations that led to the results shown in this manuscript. After detailing the numerical simulations, we describe the computational cost of the VQE and VQExDBQA approaches, which are reported in the tables of this work.

All the simulations have been performed using Qibo [58, 65–70], an open-source quantum computing framework widely used to run quantum machine learning algorithms both in simulation [71–77] and on quantum hardware [78–80]. The optimizations have been performed through the Qibo interface, which integrates robust Python based frameworks such as keras [81], tensorflow [82], scipy [83] and pycma [50].

### 1. VQE training and computational cost

**Training description** — We train the VQE using a gradient-based optimization approach. We use the Adam optimizer [84], which has been proven to be one of the most effective optimizers when training big machine-learning models. We set the learning rate to 0.05 after performing a hyper-optimization on a grid of values between 0.1 and 0.001. We keep the default values of the remaining hyper-parameters according to the Keras’ implementation.

The explored VQE architectures are the Hamming weight preserving ansatz introduced in Fig. 2 and a Hardware efficient ansatz composed of RY, RZ and CZ gates. The target Hamiltonians have been chosen as cases of the general Heisenberg Hamiltonian: a first target involves nearest neighbors interactions and a penalty  $\Delta = 0.5$  to the  $\hat{Z}$ s, while in a second moment we consider a more complex case lighting up the next-nearest neighbors interactions. We refer to these two cases as XXZ and  $J_1$ - $J_2$  respectively.

Once the VQE ansatz and the target Hamiltonian are specified, we explore its expressibility by varying the number of layers in the VQE ansatz. We trained both ansätze with layers ranging from 3 to 9 and we repeated each training with five different initial configurations. Namely, we randomly sampled the initial values of the angles parametrizing the variational model from the uniform distribution  $\mathcal{U}(-\pi, \pi)$ .

After selecting the circuit sizes for which the ground state approximations were most accurate, we conducted a series of more exhaustive simulations to quantify the training error. In particular, we handle the XXZ target training Hamming-weight preserving ansätze composed of 3, 4, and 5 layers for 2000 epochs, and hardware-efficient ansätze composed of 7, 8, and 9 layers for 5000 epochs. We instead tackle the  $J_1$ - $J_2$  target training Hamming-weight preserving ansätze composed of 3, 4, 5, and 6 layers for 2000 epochs.

For each fixed target, ansatz, and number of layers we repeat the training fifty times, with a different set of initial parameters sampled from the uniform distribution  $\mathcal{U}(-\pi, \pi)$ . Each of these training instances corresponds to a ground state energy approximation  $\tilde{E}_0$ , whose quality is quantified by computing the relative difference with the target (known

or numerically computed) ground state energy  $E_0$

$$\text{RD} = 1 - \frac{\tilde{E}_0}{E_0}. \quad (\text{D1})$$

Alternatively, we can use the ground state energy approximations  $\tilde{E}_0$  to compute the following fidelity lower bound [56]

$$\mathcal{F} = 1 - \frac{\tilde{E}_0 - E_0}{E_1 - E_0}, \quad (\text{D2})$$

where  $E_0$  and  $E_1$  are the true ground state and first excited state respectively.

Once all the fifty RD or  $\mathcal{F}$  values are collected, a final estimate is computed through the median value of the list,  $\text{median}(\mathbf{x})$ , where we call  $\mathbf{x}$  the list of RD/ $\mathcal{F}$  values for simplicity. The estimation of the uncertainty is instead computed using the median absolute deviation:

$$\text{MAD} = 1.4826 \cdot \text{median}(|x_i - \text{median}(\mathbf{x})|). \quad (\text{D3})$$

We choose the median and the median absolute deviation as estimators of the variable and its uncertainty to be more robust to outliers, which could be expected when applying the VQExDBQA algorithm. Since we make use of a global cost function in this work, it can happen that, if the initial approximation provided by the VQE is not close enough to the ground state, the final effect of the DBQA doesn't correspond to a further reduction of the energy. Moreover, since the optimization cost of the DBQA is particularly intense in the case of the non-compiled DBI, in some rare cases it can happen the optimizers are not able to find an optimal configuration of the parameters when the optimization process is limited in time.

**Computational cost** — To evaluate the computational cost of the VQE training we need to take into account the total number of two-qubit gates composing the circuit architecture and the number of times an expectation value has to be computed to evaluate predictions and gradients during the training. When using a gradient-based approach on a quantum device, the gradients must be calculated using parameter shift rules [85, 86]. Since we parametrize the quantum circuit through rotational gates, it is well known the partial derivative of our cost function w.r.t. to a circuit's parameter  $\theta$  can be calculated using two expectation values. Considering the choice of Adam optimization, the entire gradient of the cost function has to be computed at each optimization iteration. Finally, the total amount of two-qubit gates (we take CZ as reference) can be computed as:

$$N_{\text{CZ}}^{\text{VQE}}(e) = k \cdot p \cdot e \cdot n_{\text{CZ}}^{\text{VQE}}, \quad (\text{D4})$$

where we indicate with  $e$  the number of Adam iterations,  $p$  the number of parameters of the circuit, and  $n_{\text{CZ}}^{\text{VQE}}$  the number of CZ gates that compose the circuit according to the chosen ansatz. The constant  $k$  refers to the number of expectation values required to execute the parameter shift rule. In the case of the hardware-efficient ansatz  $k = 2$  and in the case of the Hamming weight preserving ansatz  $k = 4$  since each RBS gate is decomposed into two rotations depending on the same angle  $\theta$  [45]. According to the same decomposition, the number of CZ gates composing the Hamming-weight preserving ansatz circuit can be calculated as twice the number of RBS gates.

## 2. VQExDBQA procedure and computational cost

**VQExDBQA procedure** — To fully exploit the algorithm, it is necessary to prepare an approximation of the ground state. In this work, we make use of a VQE, but any ground-state preparation algorithm can be used. This first approximation is then used to precondition the target Hamiltonian, which is then rotated according to Eq. (2). The following key step consists in compiling the DBI circuit and this can be done following an equivalent procedure to the one presented in Sec. E. In the following discussion of the computational cost of the algorithm, we will take into account the same VQE trainings exposed in the previous section, which are used as preconditioning ground state approximations in the VQExDBQA process. We evaluate Eq. (D1) after applying the VQExDBQA to a given ground state approximation provided by VQE stopped at target epoch  $e$ . Following the same procedure of Sec. D 1 we collect all the RD values and provide the final estimations as the median and median absolute deviation of the obtained results.

**Computational cost** — Both the initial state approximation and the GCI compilation into a circuit present a computational cost and have to be cumulatively taken into account to evaluate the whole computational expense of the algorithm. The total number of two qubits gates required by the VQExDBQA process involves then a first

contribution due to the VQE cost described in Eq. (D4). We have to consider then the cost of executing the new circuit, in which the VQE unitary is repeated exponentially as DBI iterations increase, and there is an additional cost due to the Hamiltonian decomposition as explained in Sec. E. Denoting  $n_{\text{CZ}}^{\text{DBQA}}$  the number of CZ gates composing the circuit already involving both the GCI compilation and the recursive VQE call, we finally must consider the cost of optimizing the GCI's parameters. In fact, the GCI circuit can be parametrized both in the step duration  $s$  and in the diagonal operators  $\hat{D}_k$ . In this work, we parametrize  $\hat{D}_k$  as a classical Ising model with nearest-neighbor (NN) interactions:

$$\hat{D}_k(B^{(k)}, J^{(k)}) = \sum_{i=0}^N (\alpha_i^{(k)} \hat{Z}_i + \beta_i^{(k)} \hat{Z}_{i+1} \hat{Z}_i), \quad (\text{D5})$$

where  $N$  is the number of sites in the chain, the parametrization is implemented through the coefficients  $\{\alpha_i\}_{i=0}^N$  and  $\{\beta_i\}_{i=0}^N$  and the superscript  $k$  highlights the optimization procedure is repeated for each DBQA step. We use Scipy's Powell and CMA-ES optimizers to find an optimal DBI configuration and this optimization has to be considered as additional cost in the overall count. To evaluate this final contribution to the total number of two-qubit gates we multiply  $n_{\text{CZ}}^{\text{DBQA}}$  with the total number of cost function evaluations  $n_{\text{fval}}$  executed by the optimizers. The total number of two-qubit gates required to finalize the VQExDBQA ground state approximation can be finally obtained as

$$N_{\text{CZ}}^{\text{DBQA}} = N_{\text{CZ}}^{\text{VQE}}(e) + n_{\text{fval}} \cdot n_{\text{CZ}}^{\text{DBQA}}. \quad (\text{D6})$$

### Appendix E: Compiling of XXZ evolution

We next provide a discussion on how to explicitly express the diagonalization DBQA as above into an explicit circuit. For the XXZ model we write the Hamiltonian as

$$\hat{H}_0 = \sum_{a=1}^L \hat{H}^{(a)} \quad (\text{E1})$$

where each summand addresses only two qubits

$$\hat{H}^{(a)} = \hat{X}_a \hat{X}_{a+1} + \hat{Y}_a \hat{Y}_{a+1} + \hat{Z}_a \hat{Z}_{a+1} \quad (\text{E2})$$

for  $a < L$  and

$$\hat{H}^{(L)} = \hat{X}_L \hat{X}_1 + \hat{Y}_L \hat{Y}_1 + \hat{Z}_L \hat{Z}_1 \quad (\text{E3})$$

because of periodic boundary conditions. We next use  $M$  steps of the linear Trotter-Suzuki decomposition

$$e^{-it\hat{H}_0} = \left( \prod_{a=1}^L e^{-i\frac{t}{M}\hat{H}^{(a)}} \right)^M + O(M^{-1}) \quad (\text{E4})$$

which means that, if we decompose  $e^{-it/M\hat{H}^{(a)}}$  into CNOT and single qubit rotations, we get a circuit approximation with accuracy  $O(t^2/M)$ . This is done by noticing that  $e^{-it/M\hat{H}^{(a)}}$  is a unitary acting on two qubits, i.e.  $a$  and  $a+1$  and any two-qubit unitary can be decomposed into a circuit made of single qubit rotations and 3 CNOT gates [87].

We next discuss generalizations. The Hamiltonian in Eq. (E1) can be arbitrary and only the last step has to be modified: Then the terms  $\hat{H}^{(a)}$  can be acting on more than 2 qubits and the evolutions they generate  $e^{-it/M\hat{H}^{(a)}}$  can be compiled into CNOTs and single qubit rotations using the quantum Shannon decomposition [88] which implements a unitary on  $K$  qubits using  $O(4^K)$  CNOTs which is, in general, an optimal scaling.

Additionally, we can use a higher-order Trotter-Suzuki decomposition. We assume again that we have a 2 qubit Hamiltonian on a line of an even number of qubits  $L$  so that  $\hat{H}_0^{(o)} = \sum_{a=1}^{L/2} \hat{H}^{(2a-1)}$  and  $\hat{H}_0^{(e)}$  contains only commuting terms. Then we write

$$e^{-it\hat{H}_0} = \left( e^{-it/(2M)\hat{H}_0^{(o)}} e^{-it/M\hat{H}_0^{(e)}} e^{-it/(2M)\hat{H}_0^{(o)}} \right)^M + O(M^{-2}) \quad (\text{E5})$$

which gives an improved error scaling. From commutativity, we have the exact equality

$$e^{-it/(2M)\hat{H}_0^{(\circ)}} = \prod_{a=1}^{L/2} e^{-it/M\hat{H}^{(2a+1)}} \quad (\text{E6})$$

Each of the factors above can again be decomposed into CNOTs and single qubit rotations by standard methods [87, 88].

### 1. Other 2-local models

The addition of the magnetic field Hamiltonian can be compiled in the above formalism by setting

$$\hat{H}^{(a)} = \hat{X}_a\hat{X}_{a+1} + \hat{Y}_a\hat{Y}_{a+1} + \hat{Z}_a\hat{Z}_{a+1} + B_aZ_a \quad (\text{E7})$$

for  $a < L$  and for periodic boundary conditions we set

$$\hat{H}^{(L)} = \hat{X}_L\hat{X}_1 + \hat{Y}_L\hat{Y}_1 + \hat{Z}_L\hat{Z}_1 + B_L\hat{Z}_L \quad (\text{E8})$$

or for open boundary conditions  $\hat{H}^{(L)} = 0$  so that

$$\hat{H}_0 = \hat{H}_{\text{XXZ}} + \hat{H}(B) = \sum_{a=1}^L \hat{H}^{(a)}. \quad (\text{E9})$$

For the transverse-longitudinal field Ising model we set

$$\hat{H}^{(a)} = \hat{X}_a\hat{X}_{a+1} + B_a\hat{Z}_a + C_a\hat{X}_a \quad (\text{E10})$$

and boundary terms similar to those above. This would give as above a compiling solution with 3 CNOT gates.

### 2. Special purpose compiling for the transverse-field Ising model

We next consider the special transverse-field Ising model

$$\hat{H}^{(a)} = \hat{X}_a\hat{X}_{a+1} + B_a\hat{Z}_a \quad (\text{E11})$$

obtained from the above by setting  $C_a = 0$ . We use that  $\hat{X}_a\hat{X}_b = \text{CNOT}(a,b)\hat{X}_a\text{CNOT}(a,b)$  and  $\hat{Z}_a = \text{CNOT}(a,b)\hat{Z}_a\text{CNOT}(a,b)$  to write

$$\text{CNOT}(a,b)\hat{H}^{(a)}\text{CNOT}(a,b) = \hat{X}_a + B_a\hat{Z}_a \quad (\text{E12})$$

which means that we can get for any  $t$

$$e^{-it\hat{H}^{(a)}} = \text{CNOT}(a,b)e^{-it(\hat{X}_a+B_a\hat{Z}_a)}\text{CNOT}(a,b). \quad (\text{E13})$$

In other words, the transverse-field Ising model evolutions can be compiled using 2, not 3, CNOT gates per interaction term.

### 3. Compiling for the classical Ising model

In the numerical calculations, we use parametrizations for the diagonal evolutions which have a low quantum compiling. More specifically we define the classical Ising model

$$\hat{H}(B, J) = \sum_{a=1}^L (B_aZ_a + J_{a,a+1}\hat{Z}_a\hat{Z}_{a+1}) \quad (\text{E14})$$

where as above  $Z_{L+1} = Z_1$ . For  $\hat{H}(B, J)$  we can compile the diagonal evolution using 2 CNOT gates which is more efficient than using the general method of compiling 2-qubit unitaries from Ref. [87]. We use that  $Z_aZ_b = \text{CNOT}(a,b)Z_b\text{CNOT}(a,b)$  and by unitarity of the CNOT gate

$$e^{-itZ_aZ_b} = \text{CNOT}(a,b)e^{-itZ_b}\text{CNOT}(a,b). \quad (\text{E15})$$

The model is commuting so  $e^{-it\hat{H}(B,J=0)}$  consists of independent single qubit rotations.



## Appendix F: VQExDBQA results considering the $J_1$ - $J_2$ model

In the main text, we presented results for the SU(2) symmetric XXZ model where we set  $\Delta = 0.5$  as penalty to the  $\hat{Z}$  interactions. Here we extend the analysis to a more general model:

$$\hat{H}_{J_1-J_2} = J_1 \hat{H}_{\text{XXZ}} + J_2 \sum_{i=1}^L (\hat{X}_i \hat{X}_{i+2} + \hat{Y}_i \hat{Y}_{i+2} + \hat{Z}_i \hat{Z}_{i+2}), \quad (\text{F1})$$

where we light up next-nearest neighbors interactions. In particular, we consider  $J_1 = 1$  and  $J_2 = 0.2$ , which corresponds to a regime such that  $J_2/J_1 = 0.2$ , and has been chosen knowing the target system presents a Berezinskii–Kosterlitz–Thouless transition at  $J_2/J_1 = 0.24116$  [89, 90]. Also in this case we use the Hamming-Weight preserving ansatz presented in Fig. 3 because the same symmetries are respected. We follow the same procedure as the one presented in the main text for XXZ, but in this case, we execute the DBQA in single commutator mode only. We postpone to future works the compilation of the model into a quantum circuit likewise we did in App. E. The obtained results are presented in Tab. V.

Layers	Warm-start	1 DBI step	2 DBI steps	3 DBI steps	Long VQE training
3	0.033 ± 0.005	0.020 ± 0.005	0.016 ± 0.005	0.014 ± 0.004	0.026 ± 0.006
4	0.017 ± 0.007	0.006 ± 0.005	0.003 ± 0.003	0.002 ± 0.002	0.010 ± 0.004
5	0.011 ± 0.007	0.002 ± 0.002	0.0008 ± 0.0008	0.0004 ± 0.0004	0.005 ± 0.003
6	0.009 ± 0.006	0.002 ± 0.001	0.0007 ± 0.0007	0.0004 ± 0.0004	0.005 ± 0.003

Table V. Relative difference between approximated energy  $\tilde{E}_0$  and the target ground state value  $E_0$  for the  $J_1$ - $J_2$  model of Eq. (F1) with  $L = 10$  qubits,  $J_1 = 1$  and  $J_2 = 0.2$  together with the cumulative number of CZ gates,  $N_{\text{CZ}}$ , used to reach  $\tilde{E}_0$ . The estimates with their uncertainties were calculated using the median and the median absolute deviation of a sample of results obtained by repeating the execution fifty times with different initial conditions. Warm-start VQE approximations (500 epochs of training) are presented alongside VQExDBQA results, executed considering DBI unitaries. Longer VQE trainings (2000 epochs) are reported in the last column of the table.

## Appendix G: Details on relation to other methods

There exist many other VQE ideas and we refer to reviews for specific discussion of their detailed performance [6–8]. Overall, it is clear that purely variational methods can be limited by training obstructions (e.g. swamps of local minima or barren plateaus). So, the accuracy in energy estimation is expected to saturate before reaching the global minimum. In all variational cases, one can interface the unitary  $\hat{U}_\theta$  with DBQA via the rotation of the input generator (B11) as done in the main text.

We note that VQExDBQA is conceptually distinct from Ref. [60] which uses DBI methods — framed there in the language of Riemannian flows — to compute the gradients of generic fixed parameterized quantum circuits. That is, Ref. [60] is a quantum algorithmic implementation of Riemannian gradient descent steps. In contrast, we are using DBI methods as a new ansatz design for parameterized quantum circuits.

Gradient-based optimization is often used for training VQE and it is natural to consider these methods for training DBQA as well. An example of such 1-dimensional optimization is to use a greedy strategy to select DBI step durations  $s_k$  as cost function minimizers for the respective DBR. This strategy can be more general, e.g. gradient descent in the space of magnetic fields  $B_i$  or Ising couplings  $J_{i,j}$  to find the best  $\hat{D}_k$  operators in each step starting from an initial guess. We have also tried non-gradient-based optimizers, see App. D.

An interesting avenue of research is to replace VQE unitaries  $\hat{U}_\theta$  by Hartree-Fock warm-starts, see Ref. [19] for a discussion of quantum compiling using Givens rotations and experimental data. In this case, again, the question would be if Hartree-Fock interfaced with DBQA could achieve better ground state approximation. Further generalization to other ansätze in quantum chemistry can be interfaced with DBQA as long as explicit compiling is available.

Apart from direct minimization of energy, there exist methods that aim to implement certain transformations and these transformations have energy-lowering properties. Quantum imaginary-time evolution (QITE) is an iteration where in each step one seeks an approximation to the imaginary-time update  $|\psi_{k+1}\rangle = e^{-\tau \hat{H}_0} |\psi_k\rangle / \|e^{-\tau \hat{H}_0} |\psi_k\rangle\|$  through a unitary  $\hat{U}_k$  such that  $\hat{U}_k |\psi_k\rangle \approx |\psi_{k+1}\rangle$ . Here, the optimal parametrization of  $\hat{U}_k$  is not known so variational methods have been used [20, 22, 23]. QITE circumvents trainability issues by trying to implement steps of an iteration

that is bound to decrease the energy but it is not clear how to find the correct unitary for longer durations where the cost function resolution is limiting. Instead, DBQA is coherent in that it can be applied directly and only relies on measurements to optimize its performance but does not rely on classical supervision for its parametrization. For this reason, it is likely the QITExDBQA interfacing should be preferred over the DBQAxQITE. However, if it is possible to easily parameterize QITE at late stages of energy optimization, then QITExDBQAxQITE could be useful too.

So far we discussed methods that have been suggested without the use of auxiliary qubits. If one allows for that then methods based on quantum phase estimation (QPE) become key. Here again interfacing can be possible. Most likely DBQAxQPE should be preferred over QPExDBQA because QPE can be expected to be more costly than DBQA so it is natural to chain methods to increase resource cost, e.g. VQExDBQAxQPE. QPE methods can be systematically formulated in the formalism of qubitization. Note, that QITE can also be implemented in this way. In all cases, the upper bound on the ground state energy found via VQExDBQA (or DBQAxQPE) could also be complemented with a lower bound of the ground state energy via Dual-VQE [91].

Finally, let us comment broadly on DBQA concerning the so-called *quantum approximate optimization algorithm* (QAOA) [6, 92]. Let us begin with Eq. (A10) and notice the resemblance to a 1-layer QAOA, in that in  $\hat{V}_1$  we have a layer of single-qubit gates (specifically, diagonal  $Z$ -rotations) followed by the evolution under the input Hamiltonian and then again single-qubit rotations. For the first step, this is similar to QAOA which interlaces single-qubit rotation layers with evolutions under the input Hamiltonian. The biggest difference is that in DBQA the two single-qubit layers are related in that they are inverses of each other. Thus the guidance of the DBI equation through the reduced group commutator approximation suggests a more constrained QAOA ansatz. This constraint facilitates the monotonicity relation (A10) which allows us to analytically understand why diagonalization ensues. Similar relations for unconstrained QAOA are not known to us but Ref. [15] conjectured that this might hint at understanding the functioning of optimized QAOAs. Aside from these coincidental similarities, the methods are different in several key aspects. In Ref. [92] the QAOA ansatz involving evolutions with the native Hamiltonian for trapped ions was used to prepare an approximation to the ground state of a different model. In contrast, DBQA is not restricted to target only the ground state but can be applied to target any other eigenstate of the given model. The distinction from QAOA becomes more evident when applying DBQA beyond the first step. For example in the second step, (B20) another difference appears in that  $\hat{V}_2$  involves not only forward evolutions under the input Hamiltonian but also respective backward evolutions. Indeed, while it may be challenging to implement DBQA beyond the first step on analog quantum simulators, it is this nested forward and backward evolution structure that ensures the convergence of DBQA. We are not aware of convergence guarantees of the DBQA type for QAOA.

### 1. Overview of currently available circuit depth

Following the overview above, most methods for preparing ground states on existing quantum hardware suffer one of two severe problems. Variational methods are limited by the resolution of the cost function which is at the root of their operation. This limits the circuit depth that is meaningful in practice to about a dozen entangling layers. QPE-based methods are the opposite in that to become meaningful they need a large circuit depth.

As a specific example, we consider Ref. [93] which compared the performance of QPE for physical and logical qubits. The algorithmic performance in Ref. [93] was obtained by using  $N \approx 920$  CZ gates for  $L = 8$  qubits and the 2-qubit gate fidelity was  $p_e \approx 2 \times 10^{-3}$ .

We can use these experimental results to get ballpark figures for currently available depths. Upcoming devices are set to reach  $p'_e \approx 5 \times 10^{-4}$  [94] so assuming each CZ gate fails independently of the others we can estimate the number  $N'$  of gates that can be used meaningfully by solving heuristically

$$(1 - p_e)^N = (1 - p'_e)^{N'} \quad (\text{G1})$$

This equation gives 16% success probability which in the three-nines fidelity regime  $p'_e$  would appear for  $N' \approx 3690$  CZ gates. From these estimates, we see that there exists quantum hardware which operates meaningfully for circuit depths of about 100 CZ gates per qubit and this is set to even larger values in the very close future. This means that 1 and 2 GCI steps are already feasible for implementation on existing noisy intermediate-scale quantum devices.