Transformer Quantum State: A Multi-Purpose Model for Quantum Many-Body Problems

Yuan-Hang Zhang[1, *] and Massimiliano Di Ventra[1, †]

[1]*Department of Physics, University of California, San Diego, CA 92093, USA*

Inspired by the advancements in large language models based on transformers, we introduce the transformer quantum state (TQS), a versatile machine learning model for quantum many-body problems. In sharp contrast to Hamiltonian/task specific models, TQS can generate the entire phase diagram, predict field strengths with experimental measurements, and transfer such knowledge to new systems it has never seen before, all within a single model. With specific tasks, fine-tuning the TQS produces accurate results with small computational cost. Versatile by design, TQS can be easily adapted to new tasks, thereby pointing towards a general-purpose model for various challenging quantum problems.

Determining the state of a quantum many-body system is one of the fundamental problems in physics. While the exponential growth of the Hilbert space precludes brute-force calculations, computational methods such as quantum Monte Carlo [1] and tensor network-based methods [2] allow for efficient simulations of certain problems, each with their own strengths and weaknesses.

More recently, the advancements in machine learning techniques and models have influenced the physics community. In fact, the introduction of neural networks (NNs) as variational states for quantum many-body problems has greatly expanded the types and sizes of systems that can be efficiently tackled. For instance, the restricted Boltzmann machine [3, 4] was the first NN model applied to correlated quantum systems [5], followed by models with different architectures such as feed-forward [6, 7], convolutional [8, 9], recurrent [10], and autoregressive [11–13] ones. With the ability to encode area and volume-law entanglement [14], NNs are especially advantageous in dealing with high-dimensional systems. And with proper tricks, they can also greatly ease the fermion sign problem [6].

Yet, despite these successes, the previous approaches are limited to specific tasks. Recently, a new task-agnostic model has been put forward by the machine learning community: the transformer architecture [15]. Since its introduction, this model has dominated the field by achieving state-of-the-art results in almost every natural language processing task [16–19], thus rendering the recurrent neural networks obsolete in merely a few years. Transformers have also been adapted to different tasks such as image recognition [20], audio processing [21] and graph classification [22], all achieving remarkable results.

This feat relies on an impressive aspect of transformer models: their ability to scale to very large sizes. As a notable example, the language model GPT-3 [19], has 175 billion parameters and is trained on 45 TB of text data. A much larger model size implies capability to handle a much wider variety of data, which encourages knowledge transfer between different tasks. For GPT-3, while the training goal is nothing more than predicting the next token in a sequence, it is capable of a wide range of tasks, including writing articles, solving math problems and generating computer codes. More recently, the generalist agent Gato [23] could play Atari games, control robot arms, caption images and chat, all within the same neural sequence model. When facing with a new task, few-shot learning [19] allows a general purpose model to easily adapt with merely a few examples in natural language. And when better performance is desired, fine-tuning on a small dataset produces satisfactory results within a short time [18].

These results give hope that the such an architecture may be of great help in quantum physics as well. However, the application of transformers in this field is still rather limited, with a few results concerning quantum lattice models [13], open systems [24], quantum state tomography [25] and quantum circuit simulation [26], while the task-agnostic property is barely used. Therefore, the full potential of the transformer architecture has yet to be explored.

Contrary to the general-purpose models mentioned above, NN models in physics are usually highly specialized, serving a single purpose such as representing wave functions [5–12], preparing and controlling quantum states [26, 27], recognizing phase transitions [28, 29], realizing quantum state tomography [25, 30, 31], etc. Such tasks share a lot of common knowledge, making it ideal to have a single, unified model that handles them all, with the possibility of discovering new physics at the intersection of different tasks.

As a first step, we may consider using NNs as variational wave functions. Traditionally, each NN can only represent a specific quantum state, and tasks such as generating a phase diagram requires retraining of the same NN from scratch for hundreds of times, even if nearby data points have similar features. A possible improvement on this issue is transfer learning, namely reusing the features learned on one task, applying them to similar tasks to accelerate training. For instance, Ref. [32] demonstrated the effectiveness of this approach on restricted Boltzmann machines.

In this paper, we consider a new perspective: instead of modeling a specific quantum state, we attempt to represent a *family of quantum states* within a single neural network. More precisely, we focus on the joint distribution of the wave function and relevant physical parameters such as interaction strength, external field, and/or system size. For the underlying NN, we choose the transformer architecture for its versatility and strong performance across different tasks.

We call this model a *transformer quantum state* (TQS), and show that it is capable of generating the entire phase diagram

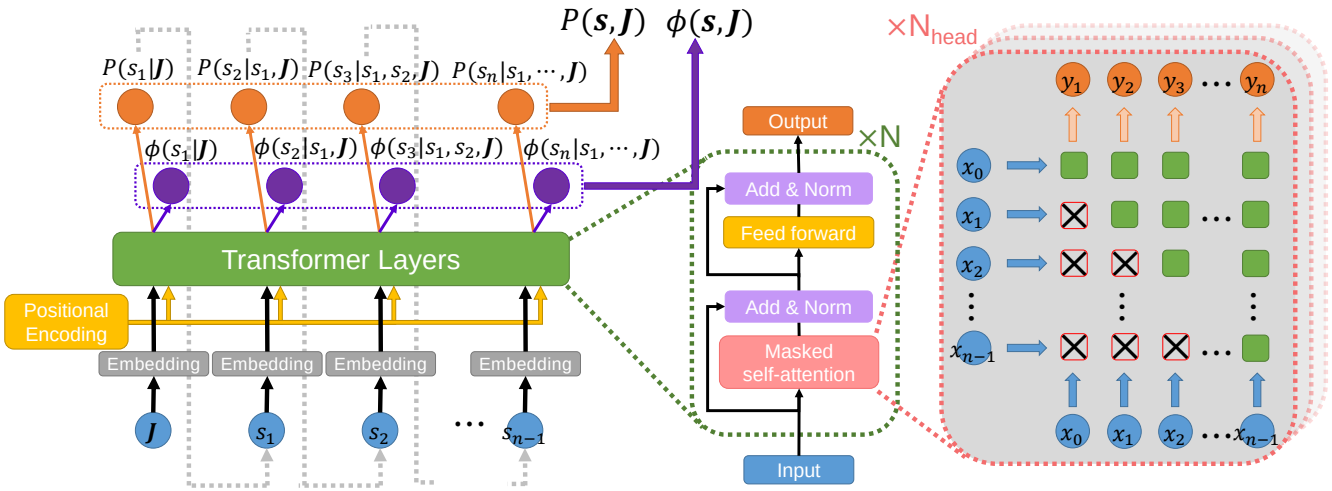

FIG. 1. The structure of a TQS. Left: the overall architecture of our model. We use the standard encoder-only transformer architecture, utilizing an embedding layer to map different inputs into a single unified feature space, and pass them through N identical transformer encoder blocks, followed by two different output heads, parameterizing the amplitude P and phase ϕ, respectively. Middle: The structure of a transformer encoder block. Right: the mask structure in a masked self-attention operator. Squares with a cross represent the masks, blocking the flow of information, so that each site only has access to its predecessors. This ensures that the autoregressive property is satisfied.

of a many-body system, predicting field strengths with as few as one experimental measurement, and transferring knowledge to new systems it has never seen before, all within a single model.

Transformer quantum state —Consider the probability distribution $P(\mathbf{s},\mathbf{J}) \equiv P(s_1,\cdots,s_n,J_1,\cdots,J_m)$, where $s_i \in \{0,1,\cdots,d-1\}$ are discrete variables representing the physical degrees of freedom such as spin or occupation number, and J_j correspond to other physical parameters, either continuous or discrete. Such a state space grows exponentially with the number of variables, and a compact representation is desired.

To represent $P(\mathbf{s},\mathbf{J})$, we adopt the transformer architecture, and autoregressively model the entire distribution as a product of conditional distributions,

$$P(\mathbf{s},\mathbf{J}) = P(\mathbf{J})\prod_{i=1}^{n} P(s_i|s_1,\cdots,s_{i-1},\mathbf{J}). \tag{1}$$

The structure of the transformer is shown in Fig. 1, with each output of the neural network representing one of the conditional distributions. Contrary to energy-based models such as restricted Boltzmann machines [5], the autoregressive structure allows for efficient sampling [11]. Since each conditional probability $P(s_i|s_1,\cdots,s_{i-1},\mathbf{J})$ does not depend on any variable s_j with $j>i$, starting from s_1, one can sequentially sample s_i according to the previously sampled configurations, using the i-th conditional distribution only. We assume that $\mathbf{J}$ has a predefined prior distribution $P(\mathbf{J})$, which, in general, can be chosen as a uniform distribution over the range of interest (e.g., to study the transition in the Heisenberg J_1-J_2 model [9, 33], one can fix $J_1=1$ and make J_2 uniform over $[0,1]$).

Our aim is to model quantum states $|\psi(\mathbf{J})\rangle$, which are complex-valued quasiprobability distributions. To this end, we expand them in the computational basis and separate their amplitude A and phase ϕ,

$$\begin{aligned}|\psi(\mathbf{J})\rangle &= \sum_{\mathbf{s}} \psi(\mathbf{s},\mathbf{J})|\mathbf{s}\rangle \\ &= \sum_{\mathbf{s}} A(\mathbf{s},\mathbf{J})\exp(i\phi(\mathbf{s},\mathbf{J}))|\mathbf{s}\rangle.\end{aligned} \tag{2}$$

Since squared amplitude has the probability interpretation, we choose

$$A(\mathbf{s},\mathbf{J}) = \sqrt{P(\mathbf{s},\mathbf{J})}, \tag{3}$$

with $P(\mathbf{s},\mathbf{J})$ specified in Eq. (1). The phase ϕ has no restrictions and can be either positive or negative, and we represent it with a similar autoregressive structure:

$$\phi(\mathbf{s},\mathbf{J}) = \sum_{i} \phi(s_i|s_1,\cdots,s_{i-1},\mathbf{J}). \tag{4}$$

Ground state of a family of Hamiltonians —The first task we consider is finding the ground state $|\psi\rangle$ of many-body Hamiltonians. Per the standard procedure, this can be done by minimizing the variational energy estimation, $\langle\psi|\hat{H}|\psi\rangle$, over the target Hamiltonian $\hat{H}$. A minor complication is that, instead of a single Hamiltonian $\hat{H}$, we have now a *family* of Hamiltonians $\{\hat{H}(\mathbf{J})\}$. In the supplemental material (SM), we show that, the family of ground states $|\psi(\mathbf{J})\rangle$ corresponds to the ground state $|\Psi\rangle$ of the super-Hamiltonian $\hat{\mathcal{H}} = \bigoplus_{\mathbf{J}} \frac{\hat{H}(\mathbf{J})}{|E_g(\mathbf{J})|}$ in the extended Hilbert space, and we can optimize the TQS by minimizing $\langle\Psi|\hat{\mathcal{H}}|\Psi\rangle$, which follows the standard procedure.

Once we have the family of ground states $\psi(\mathbf{s}, \mathbf{J})$, an immediate application is to estimate the physical parameters $\mathbf{J}$ using samples $\mathbf{s}$ from the wave function. This follows trivially from the conditional probability:

$$P(\mathbf{J}|\mathbf{s}) = \frac{P(\mathbf{s}, \mathbf{J})}{P(\mathbf{s})}. \tag{5}$$

In practice, given a set of measurements $\{\mathbf{s}_k\}$, $\mathbf{J}$ can be predicted using standard maximum likelihood estimation [34], by maximizing the log-likelihood functional,

$$\mathcal{L}(\mathbf{J}) = \sum_k \log P(\mathbf{s}_k|\mathbf{J}). \tag{6}$$

In this way, we can efficiently determine physical properties of a quantum system with few measurements.

This task is somewhat similar to shadow tomography [35, 36], in the sense that we are predicting properties of a quantum system with a few measurements, but with more restrictions and with certain prior knowledge required. On the other hand, Ref. [28] considered another similar task of recognizing phases from measurements using machine learning, which is formulated as a classification task. In comparison, our task falls in the middle of the two mentioned above, and to the best of our knowledge, it has never been proposed. Under this setting, the TQS can handle this task extremely efficiently. In fact, with the prior knowledge that a quantum state $|\psi\rangle$ comes from a family of states $|\psi(\mathbf{J})\rangle$, we can efficiently determine the physical parameters $\mathbf{J}$, with as few as one measurement only.

Furthermore, we show that the TQS can transfer knowledge to new systems it has never seen before. This follows the pre-training plus fine-tuning methodology commonly adopted in natural language models [16, 17]. In the zero-shot setting [19], after training on the family of Hamiltonians $\hat{H}(\mathbf{J})$, TQS can correctly generate the ground state of new Hamiltonians $\hat{H}(\mathbf{J}^*)$ with $\mathbf{J}^* \notin \{\mathbf{J}\}$, albeit with slightly larger error. When higher accuracy is desired, one can fine-tune the TQS on the specific Hamiltonian $\hat{H}(\mathbf{J}^*)$, to obtain accurate results within a much shorter time comparing to learning from scratch.

Results —As a prototypical test bed, we first examine the 1D transverse field Ising (TFI) model, whose Hamiltonian is

$$\hat{H} = -J \sum_{i=1}^{n-1} \sigma_i^z \sigma_{i+1}^z - h \sum_{i=1}^{n} \sigma_i^x, \tag{7}$$

where J is the coupling constant and σ^z and σ^x are Pauli matrices. In the SM, we also provide numerical results on the 1D XYZ model.

To begin with, we pre-train the TQS on the family of TFI Hamiltonians $\hat{H}(n, h)$, specified in Eq. (7). We fix $J = 1$, and assume a uniform distribution of the transverse field $h \in [0.5, 1.5]$. The system size n can take any even integer value with equal probability in the range of $[10, 40]$. We explicitly enforced parity and spin flip symmetry on the TQS, with details elaborated in the SM.

After pre-training for 10^5 iterations, we plot the ground state energy and magnetization for $n = 40$, $h \in [0, 2]$ in Fig. 2. Since we explicitly symmetrized the TQS with the $|0\rangle \leftrightarrow |1\rangle$ spin flip symmetry, we always have $\langle\sigma^z\rangle = 0$, so $\langle|\langle\sigma^z\rangle|\rangle$ (inner expectation taken over each sample, and outer expectation taken over the entire batch) is plotted instead. Note that while the TQS is only trained in the range of $h \in [0.5, 1.5]$, it can correctly infer the properties of the ground state when $h \in [0, 0.5)$ and $h \in (1.5, 2]$, without any additional inputs except the value of h.

At this point, we further fine-tune the TQS on specific points $\hat{H}(n^*, h^*)$ for an additional 2×10^3 iterations, and the results are also shown in Fig. 2. Outside of the pre-trained region, the accuracy improved dramatically up to a few orders of magnitudes. Within the pre-trained region, there is also a small improvement in accuracy, but not as much since the pre-trained model already works well.

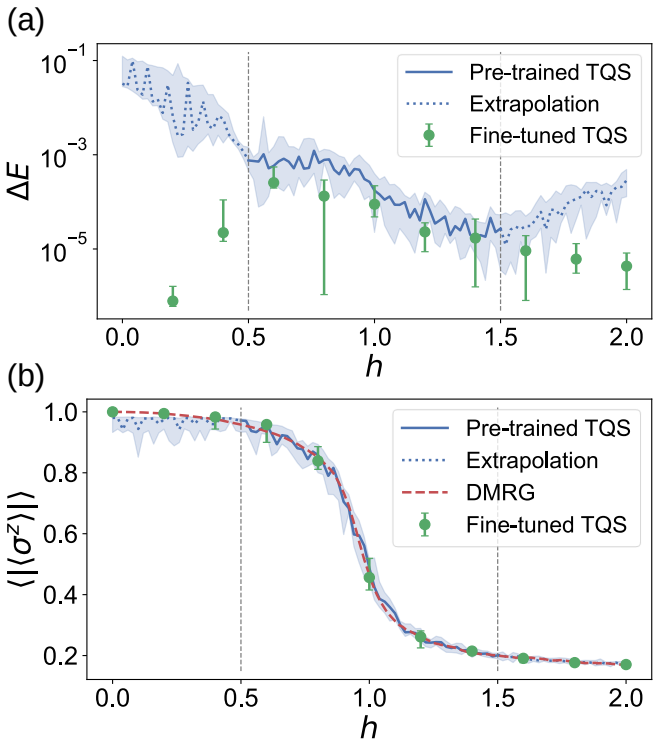

FIG. 2. Results on the ground state of the TFI Hamiltonian, Eq. (7), with $n = 40$. Lines and data points are medians of 10 estimations, while shaded regions and error bars enclose 10^{th} to 90^{th} percentile. (a) The relative error of the ground state energy, $\Delta E = |(E - E_{\text{ground}})/E_{\text{ground}}|$. E_{ground} is estimated with DMRG, which is accurate up to 10^{-10}. (b) Absolute value of the magnetization along the z direction, $\langle|\langle\sigma^z\rangle|\rangle$. The inner expectation is taken over each sample, and the outer expectation is taken over the entire batch of samples. We can clearly observe the transition near $h = 1$.

As a further test, we fix $h = 1$, and compute the ground state energy of systems with different sizes $n \in [10, 80]$. The

result is plotted in Fig. 3. Again, even if the pre-trained model has never seen any system with more than 40 spins, it can correctly predict the ground state energy of much larger systems, and their energy estimations can be greatly improved by fine-tuning for an additional 2×10^3 iterations.

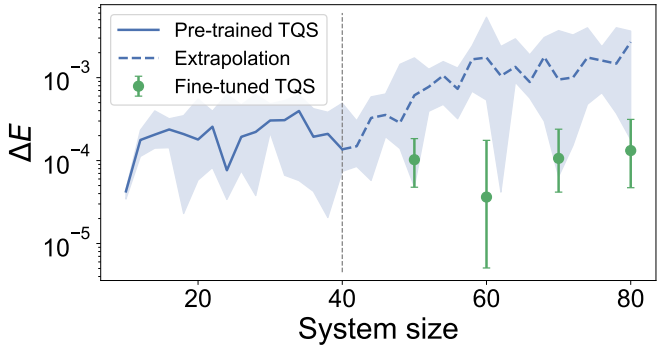

FIG. 3. Relative error of the ground state energy of the TFI Hamiltonian, Eq. (7), with $h = 1$. Lines and data points are medians of 10 estimations, while shaded regions and error bars enclose 10^{th} to 90^{th} percentile. The pre-trained TQS can correctly infer the ground state of much larger systems than what it is trained on, without any additional input except the system size n. With fine-tuning, we can boost the accuracy by another order of magnitude.

Next, with the learned distribution $P(\mathbf{s}, n, h)$, we want to predict the transverse field h using experimentally available measurements. To this end, we simulate the experiment by computing the ground state of the TFI model using density matrix renormalization group (DMRG) [2, 37], and generate a synthetic dataset with projective measurements in the computational basis.

We fix $n = 40$, and predict h by maximizing Eq. (6), the log-likelihood functional, with varying number of measurements. The results are shown in Fig. 4. Surprisingly, with as few as one measurement, TQS gives reasonable estimations of h. Increasing the number of measurements improves the quality of prediction, and an empirical power law scaling of the prediction error versus the number of measurements is observed.

Discussion —In summary, our results demonstrate how the TQS learns various ground state properties of a physical system, and appropriately uses the acquired knowledge to solve new problems.

TQS marks the first step towards a general purpose model for quantum physics. Although we only explored here the ground states of many-body Hamiltonians, it is possible to encode many additional operations and information into the TQS, such as unitary transformations, time evolution, positive operator-valued measurements, etc. Thanks to the flexibility of neural sequence models and the transformer architecture, all the additional information can be formulated as new tokens to be passed into the embedding layer, thus maintaining the model structure simple and unified.

Moreover, our formulation of TQS is compatible with state-

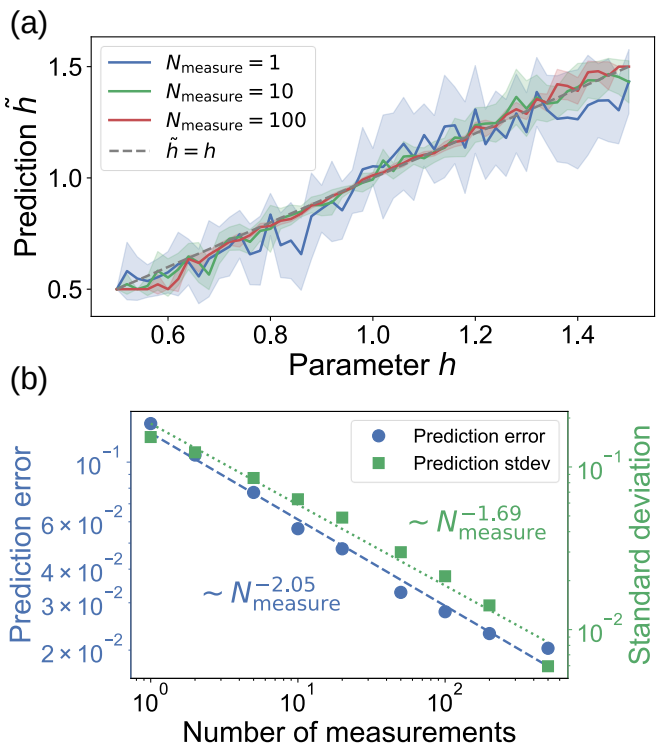

FIG. 4. (a) The predicted field strength $\tilde{h}$ vs. the actual field strength h, with varying number of measurements. Solid lines are mean values of 10 predictions, and shaded regions enclose one standard deviation. The dashed line represents the expected result, $\tilde{h} = h$. (b) Scaling of the prediction error, $|\tilde{h} - h|$, and standard deviation, $\sigma_{\tilde{h}}$, vs. the number of measurements. Each data point is computed with 10 predictions. We observe an empirical power law scaling, with $|\tilde{h} - h| \sim N_{\text{measure}}^{-2.05}$ and $\sigma_{\tilde{h}} \sim N_{\text{measure}}^{-1.69}$.

of-the-art language models [17–19], and it is possible to fine-tune a physics model on top of a language model such as GPT-3 [19]. This would make it possible for models to "comprehend" physical tasks using natural language, enabling prompt-based problem solving [19], and avoiding the inconsistencies in hand-designed features and formats for various different tasks, thus leading towards a unified general purpose model.

Limited by available computational resources, we were unable to train larger models for a wider range of tasks. But we believe that, with the advancements in the development of new computing paradigms such as MemComputing [38], such models can be pushed even further. This would help researchers understand various challenging quantum phenomena, and assist them in the design and characterization of near-term quantum devices.

Data availability —The code for all simulations performed in this paper, the weights of a pre-trained TQS on the Ising model, and a synthetic dataset generated using DMRG are available at https://github.com/yuanhangzhang98/transformer_quantum_state.

Acknowledgements—We acknowledge financial support from the Department of Energy under Grant No. DE-

SC0020892.

* yuz092@ucsd.edu
† diventra@physics.ucsd.edu

[1] W. Foulkes, L. Mitas, R. Needs, and G. Rajagopal, Reviews of Modern Physics **73**, 33 (2001).
[2] U. Schollwöck, Annals of physics **326**, 96 (2011).
[3] J. J. Hopfield, Proceedings of the national academy of sciences **79**, 2554 (1982).
[4] G. E. Hinton, Neural computation **14**, 1771 (2002).
[5] G. Carleo and M. Troyer, Science **355**, 602 (2017).
[6] Z. Cai and J. Liu, Physical Review B **97**, 035116 (2018).
[7] K. Choo, G. Carleo, N. Regnault, and T. Neupert, Physical review letters **121**, 167204 (2018).
[8] X. Liang, W.-Y. Liu, P.-Z. Lin, G.-C. Guo, Y.-S. Zhang, and L. He, Physical Review B **98**, 104426 (2018).
[9] K. Choo, T. Neupert, and G. Carleo, Physical Review B **100**, 125124 (2019).
[10] M. Hibat-Allah, M. Ganahl, L. E. Hayward, R. G. Melko, and J. Carrasquilla, Physical Review Research **2**, 023358 (2020).
[11] O. Sharir, Y. Levine, N. Wies, G. Carleo, and A. Shashua, Physical review letters **124**, 020503 (2020).
[12] T. D. Barrett, A. Malyshev, and A. Lvovsky, Nature Machine Intelligence **4**, 351 (2022).
[13] D. Luo, Z. Chen, K. Hu, Z. Zhao, V. M. Hur, and B. K. Clark, arXiv preprint arXiv:2101.07243 (2021).
[14] D.-L. Deng, X. Li, and S. D. Sarma, Physical Review X **7**, 021021 (2017).
[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Advances in neural information processing systems **30** (2017).
[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, arXiv preprint arXiv:1810.04805 (2018).
[17] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, OpenAI blog (2018).
[18] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, OpenAI blog **1**, 9 (2019).
[19] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, Advances in neural information processing systems **33**, 1877 (2020).
[20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, arXiv preprint arXiv:2010.11929 (2020).
[21] L. Dong, S. Xu, and B. Xu, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2018) pp. 5884–5888.
[22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, in *International Conference on Learning Representations* (2018).
[23] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, *et al.*, arXiv preprint arXiv:2205.06175 (2022).
[24] D. Luo, Z. Chen, J. Carrasquilla, and B. K. Clark, Physical review letters **128**, 090501 (2022).
[25] P. Cha, P. Ginsparg, F. Wu, J. Carrasquilla, P. L. McMahon, and E.-A. Kim, Machine Learning: Science and Technology **3**, 01LT01 (2021).
[26] J. Carrasquilla, D. Luo, F. Pérez, A. Milsted, B. K. Clark, M. Volkovs, and L. Aolita, Physical Review A **104**, 032610 (2021).
[27] Y.-H. Zhang, P.-L. Zheng, Y. Zhang, and D.-L. Deng, Physical Review Letters **125**, 170501 (2020).
[28] J. Carrasquilla and R. G. Melko, Nature Physics **13**, 431 (2017).
[29] E. P. Van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Nature Physics **13**, 435 (2017).
[30] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Nature Physics **14**, 447 (2018).
[31] Y.-H. Zhang and M. Di Ventra, arXiv preprint arXiv:2112.14866 (2021).
[32] R. Zen, L. My, R. Tan, F. Hébert, M. Gattobigio, C. Miniatura, D. Poletti, and S. Bressan, Physical Review E **101**, 053301 (2020).
[33] H. Schulz, T. Ziman, and D. Poilblanc, Journal de Physique I **6**, 675 (1996).
[34] I. J. Myung, Journal of mathematical Psychology **47**, 90 (2003).
[35] S. Aaronson, SIAM Journal on Computing **49**, STOC18 (2019).
[36] H.-Y. Huang, R. Kueng, and J. Preskill, Nature Physics **16**, 1050 (2020).
[37] J. Hauschild and F. Pollmann, SciPost Phys. Lect. Notes , 5 (2018), code available from `https://github.com/tenpy/tenpy`, arXiv:1805.00055.
[38] M. Di Ventra, *MemComputing: Fundamentals and Applications* (Oxford University Press, 2022).
[39] U. Schollwöck, J. Richter, D. J. Farnell, and R. F. Bishop, *Quantum magnetism*, Vol. 645 (Springer, 2008).
[40] A. Zheng and A. Casari, *Feature engineering for machine learning: principles and techniques for data scientists* (" O'Reilly Media, Inc.", 2018).
[41] Z. Wang and J.-C. Liu, Translating math formula images to latex sequences using deep neural networks with sequence-level training (2019), arXiv:1908.11415 [cs.LG].
[42] V. Nair and G. E. Hinton, in *Icml* (2010).
[43] J. S. Bridle, in *Neurocomputing* (Springer, 1990) pp. 227–236.
[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, Advances in neural information processing systems **32** (2019).
[45] D. P. Kingma and J. Ba, arXiv preprint arXiv:1412.6980 (2014).
[46] J. A. Nelder and R. Mead, The computer journal **7**, 308 (1965).
[47] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, Nature Methods **17**, 261 (2020).
[48] J. Carrasquilla, G. Torlai, R. G. Melko, and L. Aolita, Nature Machine Intelligence **1**, 155 (2019).

Supplemental Material: Transformer Quantum State: A Multi-Purpose Model for Quantum Many-Body Problems

In this supplemental material, we provide all the technical details necessary for reproducing the results in our paper, and show some additional numerical results on the Heisenberg XYZ model [39].

Transformer implementation details

As illustrated in Fig. 1, we adopt the standard encoder-only transformer structure [15]. The discrete spin variables s_i are first one-hot encoded [40], and the parameters J_j are represented with a scaled one-hot vector. To input interaction strengths and external fields, the scale is the value of the interaction itself. To input the system size n, we choose the scale to be $\ln n$, and append another parity dimension to the input vector, indicating whether n is even or odd.

Since the input does not entirely consist of one-hot vectors, the embedding layer performs a linear transformation, mapping the input vectors into a d_e dimensional embedding space.

We use a mixed-style positional encoding. The spin variables s_i have a well-defined position, and we use the D-dimensional sinusoidal positional encoding [15, 41] on them, where D is the spatial dimension of the physical system. This ensures that the neural network can correctly extrapolate to larger system sizes it has never seen before. On the other hand, the parameters J_j do not have a position, and we use a learnable positional encoding [20] instead.

After embedding and positional encoding, we pass the embedded inputs through N identical transformer encoder layers, with structures defined in [15]. The feed-forward sublayer consists of two linear layers, with the hidden dimension in the middle also being d_e. For all models we trained, we use multi-head self-attention [15] with 8 heads, and ReLU activation [42] is used throughout the neural network.

After N transformer encoder layers, we use two output heads to model the amplitude and phase of the target wave function. The amplitude head is a linear layer followed by a softmax activation [43], and the phase head is a linear layer followed by a softsign activation, which is defined in [10] and computes the function $(-\infty < x < +\infty)$

$$\mathrm{softsign}(x) = \frac{x}{1+|x|}. \tag{S1}$$

We scale the softsign output by π, to output a phase in the range of $(-\pi, \pi)$.

All TQS structures mentioned in the paper have $N = 8$ transformer encoder layers with embedding size $d_e = 32$, and the number of parameters is about 7.7×10^4. The implementation of TQS is carried out using the PyTorch library [44].

Variational optimization of the ground state energy

TQS is trained by minimizing the ground state energies of a family of Hamiltonians, $\{\hat{H}(\mathbf{J})\}$.

For a single Hamiltonian $\hat{H}$, the energy derivative reads [5]:

$$\frac{\partial E}{\partial \theta_k} = 2\mathrm{Re}\left(\left\langle E_{\mathrm{loc}}(\mathbf{s}) \frac{\partial \log \psi(\mathbf{s})^*}{\partial \theta_k} \right\rangle_{P(\mathbf{s})}\right) \tag{S2}$$

where $\langle\cdot\rangle_{P(\mathbf{s})}$ denotes expectation over the distribution $P(\mathbf{s})$, and

$$E_{\mathrm{loc}}(\mathbf{s}) = \sum_{\mathbf{s}'} \hat{H}(\mathbf{s},\mathbf{s}') \frac{\psi(\mathbf{s}')}{\psi(\mathbf{s})} \tag{S3}$$

is the local energy estimator.

Since the autoregressive wave function is explicitly normalized, it is shown in [10] that the variance of the gradient can be reduced by subtracting a baseline energy,

$$\frac{\partial E}{\partial \theta_k} = 2\mathrm{Re}\left(\left\langle \left(E_{\mathrm{loc}}(\mathbf{s}) - \langle E_{\mathrm{loc}}(\mathbf{s}')\rangle_{P(\mathbf{s}')}\right) \frac{\partial \log \psi(\mathbf{s})^*}{\partial \theta_k} \right\rangle_{P(\mathbf{s})}\right) \tag{S4}$$

without introducing bias. This follows from

$$\begin{aligned} &\mathrm{Re}\left\langle \langle E_{\mathrm{loc}}(\mathbf{s}')\rangle_{P(\mathbf{s}')} \frac{\partial \log \psi(\mathbf{s})^*}{\partial \theta_k} \right\rangle_{P(\mathbf{s})} \\ =&\langle E_{\mathrm{loc}}(\mathbf{s}')\rangle_{P(\mathbf{s}')} \sum_{\mathbf{s}} P(\mathbf{s}) \frac{1}{2} \frac{1}{P(\mathbf{s})} \frac{\partial P(\mathbf{s})}{\partial \theta_k} \\ =&\frac{E}{2} \frac{\partial}{\partial \theta_k} \sum_{\mathbf{s}} P(\mathbf{s}) = \frac{E}{2} \frac{\partial}{\partial \theta_k} 1 = 0. \end{aligned} \tag{S5}$$

In our problem setting, we have a family of Hamiltonians $\hat{H}(\mathbf{J})$ parameterized by $\mathbf{J}$, with ground state energies $E_g(\mathbf{J})$. Without loss of generality, we suppose all $E_g < 0$; otherwise we can simply shift the energy levels by adding a constant. Then, we define the super-Hamiltonian,

$$\hat{\mathcal{H}} = \bigoplus_{\mathbf{J}} \frac{\hat{H}(\mathbf{J})}{|E_g(\mathbf{J})|}, \tag{S6}$$

to be the direct sum of all (possibly infinite) Hamiltonians $H(\mathbf{J})$, weighted by their ground state energies, $\frac{1}{|E_g(\mathbf{J})|}$. Note that $\hat{\mathcal{H}}$ is block diagonal, with no interaction across different $\mathbf{J}$s. One can easily show that, the ground state of $\hat{\mathcal{H}}$ is the direct sum of all ground states, $|\Psi\rangle = \bigoplus_{\mathbf{J}} |\psi(\mathbf{J})\rangle$,

$$\begin{aligned} \hat{\mathcal{H}}|\Psi\rangle =& \left(\bigoplus_{\mathbf{J}} \frac{\hat{H}(\mathbf{J})}{|E_g(\mathbf{J})|}\right)\left(\bigoplus_{\mathbf{J}} |\psi(\mathbf{J})\rangle\right) \\ =& \bigoplus_{\mathbf{J}} \frac{\hat{H}(\mathbf{J})|\psi(\mathbf{J})\rangle}{|E_g(\mathbf{J})|} \\ =& -\bigoplus_{\mathbf{J}} |\psi(\mathbf{J})\rangle = -|\Psi\rangle \end{aligned} \tag{S7}$$

with eigenvalue -1. Therefore, we can follow the standard procedure and minimize

$$\langle\Psi|\hat{\mathcal{H}}|\Psi\rangle = \sum_{\mathbf{J}} \frac{\langle\psi(\mathbf{J})|\hat{H}(\mathbf{J})|\psi(\mathbf{J})\rangle}{|E_g(\mathbf{J})|}. \tag{S8}$$

We don't have access to the exact ground state energies $E_g(\mathbf{J})$, so we instead approximate them with variationally approximated ground state energies, $\tilde{E}_g(\mathbf{J}) = \langle E_{\mathrm{loc}}(\mathbf{s},\mathbf{J})\rangle$, which become increasingly more accurate as optimization goes on.

In practice, at each optimization iteration, we sample a random $\mathbf{J}$ according to $P(\mathbf{J})$, and compute the energy derivative Eq. (S4), scaled by $\frac{1}{|\tilde{E}_g(\mathbf{J})|}$. We set an upper limit of 5 to the scaling factor, to avoid divergences when $\tilde{E}_g(\mathbf{J}) \to 0$ during optimization.

The entire training procedure is carried out using the Adam optimizer [45], with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. We varied the learning rate during training according to the formula,

$$\mathrm{lr}(i_{\mathrm{step}}) = 5d_e^{-0.5}\min(i_{\mathrm{step}}^{-0.75}, i_{\mathrm{step}} i_{\mathrm{warmup}}^{-1.75}) \tag{S9}$$

where d_e is the embedding size of the model, i_{step} is the current number of training steps, and i_{warmup} is the number of warm up steps. We used $i_{\mathrm{warmup}} = 4000$. This corresponds to linearly increasing the learning rate during the first 4000 iterations, and polynomially decreasing it during the rest of the training. This learning rate schedule is inspired from [15]. During fine-tuning, we use a different learning rate schedule:

$$\mathrm{lr}(i_{\mathrm{step}}) = 5d_e^{-0.5}(i_{\mathrm{step}} + 10^5)^{-0.75}. \tag{S10}$$

Sampling algorithm

The autoregressive structure of TQS already makes sampling efficient, and the efficiency is further improved by adopting the sampling algorithm in [12], which only samples unique configuration strings.

During sampling, we first fix a large batch size, N_{batch}, and autoregressively sample the spins to form partial strings, $\mathbf{s}^k = s_1 s_2 \cdots s_i$, with associated number of occurrences, n_k. At the $(i+1)$-th sampling step, s_{i+1} is sampled from the conditional distribution $P(s_{i+1}|s_1,\cdots,s_i,\mathbf{J})$, resulting in n_{k0} occurrences of $s_{i+1} = 0$ and n_{k1} occurrences of $s_{i+1} = 1$, with $n_{k0} + n_{k1} = n_k$. After this step, we obtain two unique partial strings, $\mathbf{s}^{k0} = s_1 s_2 \cdots s_i 0$ and $\mathbf{s}^{k1} = s_1 s_2 \cdots s_i 1$, with occurrences n_{k0} and n_{k1}, respectively. This procedure starts from an empty set and is repeated until the number of unique strings reaches a maximum, N_{unique}, after which no new partial string branches are generated, and the remaining spins are sampled in the regular way.

The complexity of this sampling algorithm is approximately proportional to N_{unique} and does not depend on N_{batch}. Therefore, we can choose extremely large batch sizes to greatly improve on the accuracy of estimated expectation values, with negligible increase in computation time. For all the experiments mentioned in this paper, we choose $N_{\mathrm{batch}} = 10^6$, $N_{\mathrm{unique}} = 10^2$ during training, and $N_{\mathrm{unique}} = 10^3$ during evaluations.

Implementing symmetries

The transformer architecture itself does not observe any symmetry, but most Hamiltonians do. To impose symmetries without spoiling the autoregressive structure, we follow the approaches in previous works [7, 10, 11] and explicitly symmetrize the wave function in a similar way.

Suppose $\hat{\mathcal{T}}$ is a discrete symmetry of $\hat{H}$, with $\hat{\mathcal{T}}^m = \mathbb{1}$ ($m \in \mathbb{N}$). By definition, we have $[\hat{H}, \hat{\mathcal{T}}] = 0$, and one can simultaneously diagonalize both operators within the same eigenbasis. Under this basis, the ground state $|\psi\rangle$ is also an eigenstate of $\hat{\mathcal{T}}$,

$$\hat{\mathcal{T}}|\psi\rangle = \omega_{\hat{\mathcal{T}}}|\psi\rangle \tag{S11}$$

where $\omega_{\hat{\mathcal{T}}} = e^{2\pi i k/m}$, $k \in \mathbb{N}$. Expanding Eq. (S11) in the computational basis, we get

$$\psi(\hat{\mathcal{T}}^{-1}\mathbf{s}) = \omega_{\hat{\mathcal{T}}}\psi(\mathbf{s}). \tag{S12}$$

In terms of amplitude and phase, Eq. (S12) becomes

$$\begin{aligned} A(\hat{\mathcal{T}}\mathbf{s}) &= A(\mathbf{s}), \\ \phi(\hat{\mathcal{T}}\mathbf{s}) &= \phi(\mathbf{s}) - \frac{2\pi k}{m}. \end{aligned} \tag{S13}$$

The output wave function from TQS clearly does not satisfy Eq. (S13). To explicitly enforce the symmetry $\hat{\mathcal{T}}$, we define

$$\begin{aligned} \tilde{P}(\mathbf{s}) &= \frac{1}{m}\sum_{n=0}^{m-1} P(\hat{\mathcal{T}}^n\mathbf{s}) \\ \tilde{\phi}(\mathbf{s}_0) &= \mathrm{Arg}\left(\sum_{n=0}^{m-1}\psi(\hat{\mathcal{T}}^n\mathbf{s}_0)\right) \\ \tilde{\phi}(\hat{\mathcal{T}}^n\mathbf{s}_0) &= \tilde{\phi}(\mathbf{s}_0) - \frac{2\pi k n}{m} \end{aligned} \tag{S14}$$

where $\psi(\mathbf{s}) = \sqrt{P(\mathbf{s})}e^{i\phi(\mathbf{s})}$, P, ϕ are outputs from the TQS, and $\tilde{P}$, $\tilde{\phi}$ are symmetrized probability and phase, respectively. $\mathbf{s}_0$ is an arbitrary initial configuration in each symmetry sector, predefined so that the phases within the symmetry sector can be assigned consistently. We choose $\mathbf{s}_0$ to be the configuration with the smallest decimal value, converted from its binary bitstring, within each symmetry sector.

Sampling from the symmetrized wave function has almost no additional computational cost. We follow the same procedure detailed in the previous section, and apply a random symmetry operation $\hat{\mathcal{T}}^n$ to the sampled configuration $\mathbf{s}$ in the end [10, 11]. However, to compute the exact value of $\psi(\mathbf{s})$, one needs to evaluate all configurations within the symmetry

sector and explicitly calculate Eq. (S14), which is m times more expensive.

Another symmetry worth mentioning is the $U(1)$ symmetry of the Heisenberg model, which leads to zero magnetization. This symmetry is particularly easy to implement, and we follow the same method developed in [10], by setting the probability of a partial string to 0 whenever the number of up spins or down spins exceeds half of the system size.

Note that, while the Hamiltonian $\hat{H}$ may satisfy several symmetries $\hat{\mathcal{T}}_1, \hat{\mathcal{T}}_2, \cdots$, it is possible that $[\hat{\mathcal{T}}_1, \hat{\mathcal{T}}_2] \neq 0$, making it impossible to diagonalize all symmetries at the same time. However, this does not pose a problem for us. Although $\hat{\mathcal{T}}_1$ and $\hat{\mathcal{T}}_2$ do not commute in general, they do commute in certain symmetry sectors (for example, $\omega_{\hat{\mathcal{T}}_1} = \omega_{\hat{\mathcal{T}}_2} = 1$). To implement symmetries, we need to know $\omega_{\hat{\mathcal{T}}}$ as a prior knowledge, and this information then helps us determine all compatible symmetries.

As another remark, in our implementation of TQS, we only enforced the symmetries $\hat{\mathcal{T}}$ with $\omega_{\hat{\mathcal{T}}} = 1$. We noticed that, any symmetry with $\omega_{\hat{\mathcal{T}}} \neq 1$ would impose a non-trivial phase structure to the wave function, which is somewhat arbitrary and could significantly slow down the training.

Predicting parameters

With the learned distribution $P(\mathbf{s}, \mathbf{J})$, we can predict the parameters $\mathbf{J}$ from a batch of measurements $\{\mathbf{s}_i\}$. As illustrated in the main text, this is achieved through maximizing the log-likelihood functional Eq. (6).

We carried out the maximization using the Nelder-Mead method [46], a heuristic searching algorithm based on a moving simplex, implemented in the SciPy library [47], with a tolerance of 10^{-9}.

An alternative method to predict the parameters would be supervised fine-tuning, which adds a parameter prediction head as an additional output of TQS. This would have the advantage of reducing the computational cost to one forward pass, at the expense of fine-tuning cost. We leave this as a future work.

Note that, we didn't use any phase information during the prediction. To make use of the phase structure, one needs to perform measurements in different bases, and compute a generalized likelihood function that takes all bases into account. For this, we refer the readers to [30]. Alternatively, it is possible to use informationally-complete positive operator-valued measurements (IC-POVM) to encode the complete information of a quantum state, which is developed in [48]. We can adapt the TQS structure to be compatible with IC-POVM, which we also leave as a future work.

DMRG calculations

For the 1D transverse field Ising model in the main text and the 1D XYZ model in the SM, we use density matrix renormalization group (DMRG) as a benchmark to evaluate the performance of our algorithm. DMRG can be extremely accurate for 1D systems, yet performs rather poorly in 2 or more dimensions [2].

We used the TeNPy library [37] to perform DMRG calculations. For all DMRG results mentioned in the paper, we use a maximum bond dimension of 100, and terminate when the energy tolerance 10^{-10} is achieved.

Additional numerical results

In this section, we further benchmark the performance of TQS with additional numerical experiments. We focus on the 1D Heisenberg XYZ model in a longitudinal field [39], whose Hamiltonian is given by

$$\hat{H} = J \sum_{i=1}^{n-1} \Big[(1+\gamma)\sigma_i^x \sigma_{i+1}^x + (1-\gamma)\sigma_i^y \sigma_{i+1}^y + \Delta \sigma_i^z \sigma_{i+1}^z \Big] + h \sum_{i=1}^{n} \sigma_i^z \tag{S15}$$

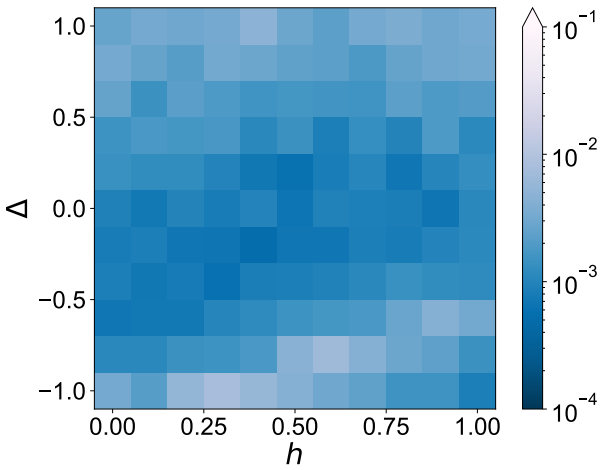

FIG. S1. The relative error of the ground state energy, $|(E - E_{\text{ground}})/E_{\text{ground}}|$, plotted against the external field h and longitudinal interaction strength Δ, with $n = 40$. In this figure, h and Δ are in the pre-training range.

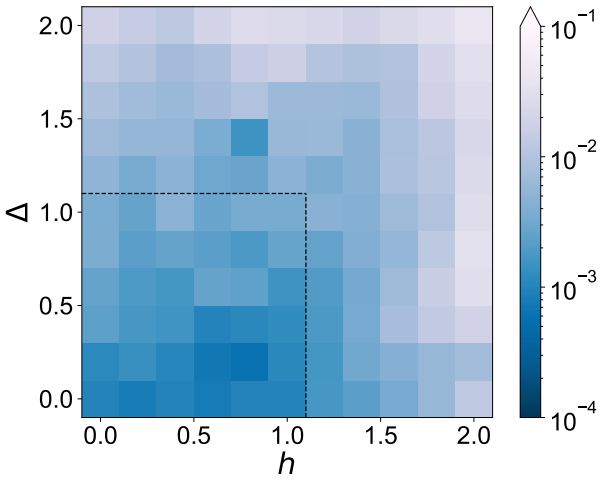

FIG. S2. The relative error of the ground state energy, $|(E - E_{\text{ground}})/E_{\text{ground}}|$, in an extended parameter range. The bottom left corner is part of the pre-training range, separated with black dashed lines for visual clarity.

We fix $J = 1, \gamma = 0.2$, and consider the parameter range $h \in [0, 1]$, $\Delta \in [-1, 1]$. The system size n can take any even integer value with equal probability in the range of $[10, 40]$. The TQS has the same structure as the one described in the main text, with 8 layers and embedding size 32. We trained the TQS for 10^5 iterations without implementing any symmetry, and the relative errors of the ground state energy, $|(E - E_{\text{ground}})/E_{\text{ground}}|$, for system size $n = 40$, are plotted in Figs. S1, S2.

Fig. S1 shows the results in the pre-trained range, $h \in [0, 1], \Delta \in [-1, 1]$, and the accuracy is on the order of 10^{-3}. In Fig. S2, we extended the parameter range to $h \in [0, 2], \Delta \in [0, 2]$, with pre-trained and extended parameter ranges separated by black dashed lines. TQS can still reasonably infer the ground state properties outside of the pre-trained range, but its accuracy gradually decreases as we move further away.

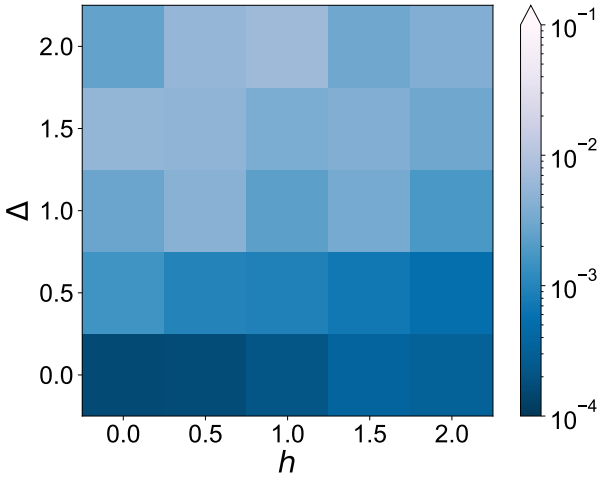

FIG. S3. The relative error of the ground state energy, $|(E - E_{\text{ground}})/E_{\text{ground}}|$, after fine-tuning the TQS on specific parameter points for 2×10^3 iterations.

To improve on the energy estimations, we fine-tune the pre-trained TQS at selected parameters for 2×10^3 iterations. The results are plotted in Fig. S3. With fine-tuning, the ground state energy accuracy improved by another order of magnitude, allowing us to more accurately estimate the ground state properties on a much wider parameter range, with minimal computational cost.