# Faster Methods for Contracting Infinite 2D Tensor Networks

M.T. Fishman,[1] L. Vanderstraeten,[2] V. Zauner-Stauber,[3] J. Haegeman,[2] and F. Verstraete[3,2]

[1] *Institute for Quantum Information and Matter, California Institute of Technology, Pasadena, California 91125, USA*
[2] *Ghent University, Faculty of Physics, Krijgslaan 281, 9000 Gent, Belgium*
[3] *Vienna Center for Quantum Technology, University of Vienna, Boltzmanngasse 5, 1090 Wien, Austria*

We revisit the Corner Transfer Matrix Renormalization Group (CTMRG) method of Nishino and Okunishi for contracting 2-dimensional tensor networks, and demonstrate that its performance can be substantially improved by determining the tensors using an eigenvalue solver as opposed to the power method used in CTMRG. We also generalize the variational uniform Matrix Product State (VUMPS) ansatz[1] for diagonalizing 1D quantum Hamiltonians to the case of 2D transfer matrices, and discuss similarities with the corner methods. These two new algorithms will be crucial in improving the performance of variational Projected Entangled Pair State (PEPS) methods.

arXiv:1711.05881v1 [cond-mat.str-el] 16 Nov 2017

## I. INTRODUCTION

2D tensor networks are ubiquitous in many-body physics[2]. They occur naturally in the context 2D classical many-body systems as representations of partition functions[3] and can represent ground states and real time evolution of 1D quantum systems, e.g. for systems with local interactions in terms of Trotter-Suzuki decompositions[4–7]. Additionally, they occur in the context of Tensor Product State (TPS)[8–10] or Projected Entangled Pair State (PEPS)[11] representations of 2D quantum systems and boundaries of 3D classical systems. Most 2D tensor networks of interest do not allow exact solutions and can only be studied approximately, and a copious array of numerical methods have been developed over many decades for their study[3,6,7,11–21].

These methods for contracting 2D tensor networks fall into two main categories, which we refer to as "coarse graining methods" and "boundary methods". Examples of coarse graining methods are Tensor Renormalization Group (TRG)[15] and extensions such as Higher Order Tensor Renormalization Group (HOTRG)[16], Second Renormalization Group (SRG)[17], and Tensor Network Renormalization (TNR)[18–20]. A common feature of these methods is that the local degrees of freedom are combined and truncated, so the Hilbert space of the network is explicitly changed at each step. For boundary methods, a matrix product state is used as an ansatz for the environment, and this matrix product state has to be determined in a variational way. These methods include the Density Matrix Renormalization Group (DMRG)[12,22,23], Corner Transfer Matrix Renormalization Group (CTMRG)[13], Time Evolving Block Decimation (TEBD)[6,7,14], etc. Boundary methods have certain advantages: they can be optimized iteratively (instead of optimized layer by layer like many coarse graining methods), the form of the environments can make it much easier to calculate arbitrary correlation functions, and they appear to be very well suited for performing PEPS calculations[11,24–26].

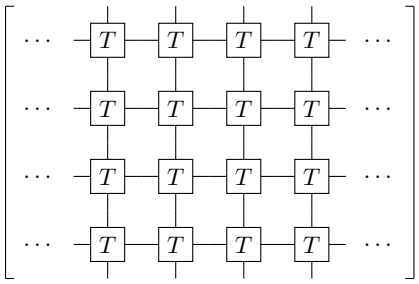
The history of these boundary methods goes back to the Corner Transfer Matrix (CTM) of Baxter[3] and White's DMRG algorithm, which served as the inspiration for Nishino and Okunishi[13] to introduce CTMRG as a powerful numerical tool for contracting 2D classical partition functions. The original PEPS[11] and IPEPS[27] calculations were done by approximating the leading eigenvector of the corresponding transfer matrices using matrix product states. However, CTMRG has lately been the most commonly used contraction method in IPEPS calculations[25,28–35].

Here we present two new approaches that improve upon the speed of CTMRG for contracting 2D tensor networks in the thermodynamic limit. First, we present the use of the recently introduced variational uniform Matrix Product State (VUMPS)[1] algorithm for contracting 2D tensor networks. We also present a new corner method analogous to CTMRG which better exploits translational invariance. We present benchmark results for VUMPS and our new corner method, showing remarkable speedups over CTMRG, particularly for systems near criticality. Our benchmarks include a variety of both 2D statistical mechanics models and 2D quantum systems represented as PEPS.

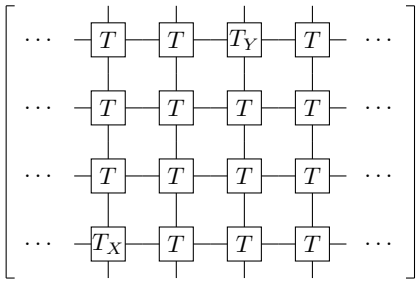
## II. PROBLEM STATEMENT

We are interested in the approximate numerical contraction of infinite 2D tensor networks. For simplicity, throughout the paper we will focus on tensor networks on an infinite square lattice with a single site unit cell. We are agnostic about where the tensor network comes from: it could be a 2D classical partition function, the norm of a PEPS, etc.

For concreteness, we are interested in evaluating:

$$\Omega^{MN} = Tr$$  (1)

where we work directly in the thermodynamic limit, i.e. the number of lattice sites in the $x, y$ direction, $M, N$, approaches $\infty$. In (1), $Tr[...]$ denotes two traces, one over the open horizontal indices and another over the open vertical indices. $\Omega$ is related to the free energy per site if the network is the partition function of a 2D statistical mechanics model, or the overlap per site if the network is a double layer PEPS. We are also interested in calculating observables such as expectation values of local operators or correlation functions. In terms of the tensor network, these are represented as impurity sites, such as:

$$\langle XY \rangle = Tr$$  $$/ \ \Omega^{MN}$$ (2)

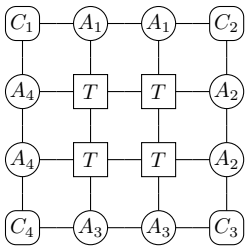
We want a contraction method that makes it easy to calculate arbitrary correlation functions, since they show up in e.g. calculating structure factors or summing Hamiltonian terms in IPEPS ground state optimizations[24–26]. For this reason we focus on MPS boundary methods, which make it much easier to calculate arbitrary correlation functions. It is much more challenging in methods like TRG/TNR where all of the tensors at each layer must properly be kept track of, and calculating arbitrary correlation functions on the lattice could get very complicated.

## III. ALGORITHM OVERVIEW

In this section, we review CTMRG, explain how VUMPS can be applied to contracting 2D tensor networks, and explain our new fixed point corner method.

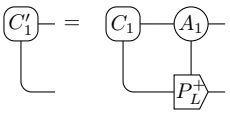
### A. Corner Transfer Matrix Renormalization Group Review

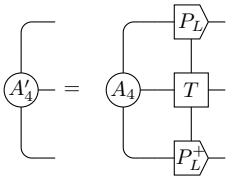
The general ansatz used for the environment in the corner transfer matrix renormalization group (CTMRG) algorithm is as follows:

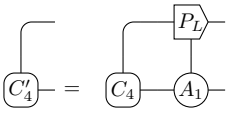
 (3)

The corner tensors $\{C_i\}$ in (3) are known as the "corner transfer matrices"[3,13], representing approximations of the infinite corners of the tensor network. The boundary MPS tensors $\{A_i\}$ in (3) represent approximations of the half infinite rows or columns of the tensor network. The CTMRG algorithm is thought of in terms of "growing" the lattice and then absorbing the introduced lattice sites into the current approximation for the environment.

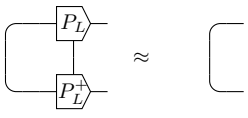
If the process is repeated indefinitely, the environment tensors would grow exponentially in size, so some sort of truncation scheme is required, which is referred to as renormalization. This renormalization of the enlarged environment is performed by introducing projectors into the network. There are multiple methods available for how to grow the lattice, as well as how to choose the projectors. One popular growth method relevant to our discussion is the single site approach, where iteratively the lattice is grown out one site in a specified direction, and the directions are cycled through until convergence. In this scheme, the so-called "left-move" is shown below:

 (4)

 (5)

 (6)

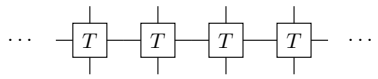
where $P_L$ is a projector and $P_L^+$ is it's (pseudo) inverse, satisfying:

 (7)

The diagrammatic notation we use for the projector is suggestive, identifying it as an MPS tensor. This notation will prove useful later on when we present our new algorithms. Projectors ($P_L^{(+)}$ for the left move above) are obtained from the current guess for the environment, and used to renormalize the lattice and obtain new environment tensors. Many methods for obtaining these projectors have been proposed over the years[13,28,29,31–33,36,37]. Symmetries in the lattice can greatly simplify obtaining the projectors, i.e. for a network that is Hermitian about all reflections, a simple Hermitian eigendecomposition can be used as originally proposed by Nishino and Okunishi[13,32,38]. However, when the network does not have symmetries to exploit, the methods for obtaining the proper projectors are much less straightforward.

### B. VUMPS for 2D tensor networks

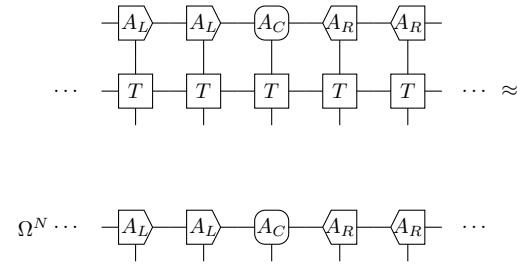
Here we present the VUMPS algorithm[1] in the context of contracting 2D tensor networks. We are interested in fixed points of the so-called row-to-row transfer matrix:




(8)

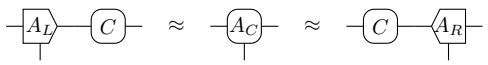
If this MPO is a Hermitian operator from the top legs to the bottom legs, it corresponds to traditional IDMRG (for example, it could be a step of imaginary time evolution of a Hamiltonian, in which case the fixed point is the ground state).

We wish to find the maximal (top) eigenvector approximated by an MPS:



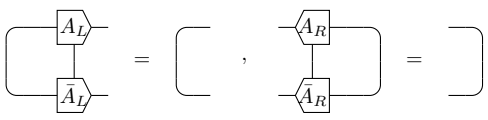

(9)

where $\Omega$ is the overlap per site of the network (for a partition function related to the free energy per site). We use the (mixed) canonical gauge:




(10)

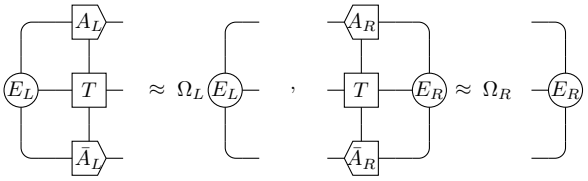
Note that these generally will not all simultaneously be satisfied exactly during the optimization, but should be satisfied to very high accuracy at the fixed point. $A_L$ and $A_R$ are isometric tensors satisfying:




(11)

at all times. The VUMPS algorithm proceeds by repeating the following steps until convergence:

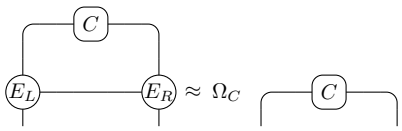
1. Solve for environments:



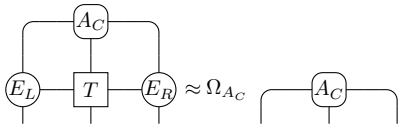

(12)

   where $\Omega_L \approx \Omega_R$ up to errors in (10).

2. Solve for zero-site and single-site tensors:




(13)




(14)

   where $\Omega_{A_C}/\Omega_C \approx \Omega_{L/R}$ near or at the fixed point.

3. Solve for new $A_L$ and $A_R$ from $A_C$ and $C$ using techniques described in the original proposal[1], and repeat 1-3 until convergence.

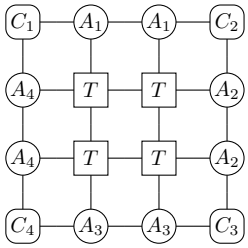
For an MPO tensor $T$ Hermitian about the horizontal, this scheme is just like the original VUMPS proposal[1]. For 2D statistical mechanics and PEPS, it is in fact simpler than the original proposal, because we generally do not have to be concerned about summing Hamiltonian terms or a Jordan block structure of the MPO's of the 2D tensor network, and methods like Arnoldi can be employed to find the fixed points.

For a non-Hermitian network, to get the environment for calculating local observables one must additionally solve for the bottom fixed point (and in order to calculate arbitrary correlation functions, the left and right fixed point MPS's as well). For a network that isn't "very asymmetric", the top fixed point can be used as a good starting point for the bottom fixed point MPS.

### C. New fixed point corner method

Here we present a new corner method, which we refer to as the fixed point corner method (FPCM), which is similar to the CTMRG algorithm but solves for environment tensors in terms of fixed points. We use the same ansatz as that used for CTMRG (as presented in Sec.

III A):




(15)

Using this ansatz for the environment, the FPCM proceeds by repeating the following steps until convergence:

1. "Bi-orthogonalize" the top and bottom uMPS's comprised of MPS tensors $A_1$ and $A_3$. Using $A_{1/3}$, we find $P^{(+)}_{L/R}$ along with a new set of $\{C_i\}$ satisfying:

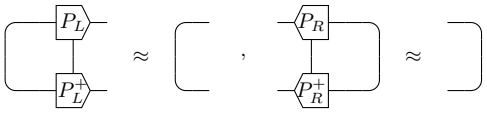

(16)


(17)


(18)

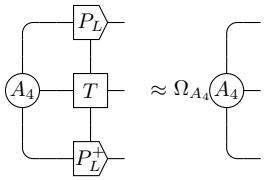

(19)

where $P^{(+)}_{L/R}$ satisfy

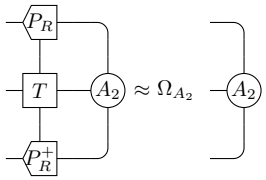



(20)

2. Obtain the left and right environments using the gauged MPS tensors $P^{(+)}_{L/R}$ found in step 1. This is done by numerically solving the following fixed point equations (in practice using an iterative method like Arnoldi):




(21)




(22)

3. Bi-orthogonalize the left and right uMPS's comprised of MPS tensors $A_2$ and $A_4$. This step is simply step 1 with the lattice rotated by $\pi/2$, so $A_{4/2}$ found in step 2 are used to find the gauged MPS tensors $P^{(+)}_{U/D}$ and a new set of $\{C_i\}$.

4. Obtain the top and bottom environment MPS tensors $A_1$ and $A_3$. This step is simply a rotated version of step 2 with the lattice rotated by $\pi/2$.

The bi-orthogonalization in steps 1 and 3 are key to the algorithm. The method we use is described in detail in Appendix A 2. The gauged MPS tensors play an analogous role to the projectors in CTMRG, and in fact can be interpreted as translational invariant versions of the CTMRG projectors.

The algorithm also looks very similar to the VUMPS algorithm, especially in the case when the network is Hermitian about a certain direction (horizontal or vertical). However, like in CTMRG, the corner matrices are used explicitly, not the center matrix of VUMPS/DMRG, and the corners can be seen roughly as "square roots" of the center matrix[2].

The leading cost of this algorithm, the calculation of the new boundaries, is $O(\chi^3 d^2)$ where $\chi$ is the bond dimension of the boundary and $d$ is the bond dimension of the network (assuming the fixed point is calculated in a sparse way with an iterative method such as Arnoldi and for simplicity assuming a large $\chi$ limit). This is the same leading cost as single site VUMPS or IDMRG. The cost of CTMRG, following the most standard schemes, is generally a full eigendecomposition, singular value decomposition, or QR decomposition of some part of the grown boundary. Since the boundary is grown from a bond dimension $\chi$ to a bond dimension $\chi d$, these decompositions lead to a scaling of the algorithm of $O(\chi^3 d^3)$, so asymptotically both (single site) VUMPS and our new corner method scale better than traditional CTMRG in the network bond dimension (exactly the limit needed for accurate IPEPS calculations).

Even so, each step of traditional CTMRG can be much faster than the new schemes presented, because of the fixed points we must calculate. However, we will see in the next section that exploiting fixed points leads to a large speedup in total convergence time, because substantially fewer steps are needed for convergence. The speedup is particularly pronounced for networks with small gaps. One way to understand this is that the original CTMRG can be viewed as a power method, where

only a single row or column of tensors is absorbed into the environment at a time, and the projectors are only determined in a local way. The new schemes properly exploit the translational invariance of the system, and iterative methods like Arnoldi are known to be much faster than power methods for finding eigenvectors of matrices with small gaps (and the gaps of the transfer matrices are expected to be related to the gap of the system[39]). In addition, the projectors that are used for renormalization in the new corner method are obtained from the current guess for the entire (translationally invariant) boundary, not just a set of local tensors.

## IV. RESULTS

Here we present benchmark results for the methods described in the previous section: CTMRG, VUMPS, and the new fixed point corner method (FPCM). We benchmark the 2D classical ferromagnetic Ising model in Section IV A, the 2D classical XY model in Section IV B, the 2D quantum spin-1/2 Heisenberg model in Section IV C, and the chiral Resonating Valence Bond (RVB) PEPS in Section IV D.

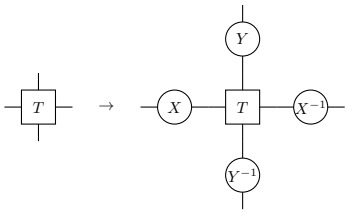
For all of the examples shown, the networks are on the square lattice and have a single site unit cell, and all tensors used are dense. Each calculation is performed with a single BLAS thread. For a fair comparison between different methods, the starting boundary states are chosen to be small (usually with bond dimension 2), the methods are run until convergence with the small bond dimension, and then the bond dimension is increased to the final one (CTMRG is used to grow the bond dimension for the FPCM, and the bond dimension growth scheme introduced in[1] is used for VUMPS). Calculations were performed using the Extreme Science and Engineering Discovery Environment (XSEDE)[40].

### A. 2D Classical Ising Model

In Figures 1 and 2 we present benchmark results for the isotropic 2D ferromagnetic classical Ising model. The MPO comprising the partition function for this model has a link bond dimension of $d = 2$, and the tensor can be taken to be real and symmetric about all rotations and reflections. The environment tensors we use for all methods are restricted to being real valued. For CTMRG and the FPCM, in the ansatz in (3) we impose $A_1 = A_2 = A_3 = A_4 = A$ and $C_1 = C_2 = C_3 = C_4 = C$, and additionally impose $A^s = (A^s)^T$ and $C = C^T$. For VUMPS, in (9) we impose $A^s_R = (A^s_L)^T$ and $A^s_C = (A^s_C)^T$. Additionally, when we calculate observables we set the bottom fixed point uMPS equal to the top fixed point uMPS obtained. For CTMRG, we find the projector to renormalize the boundary using a symmetric eigendecomposition, which is fast and numerically very stable.

From Figures 1 and 2, we see that as we approach the critical point of the 2D classical Ising model, the performance improvement of VUMPS and our new fixed point corner method over CTMRG increases. This can be understood by the fact that the boundary tensors for the new methods are obtained from solving fixed point equations (in practice with Arnoldi and Lanczos methods), which are known to be faster than power methods for finding extremal eigenvectors of matrices with small gaps. This indicates that these new methods are better suited for studying systems close to or at criticality, e.g. in combination with the theory of 'finite entanglement scaling'[41–44].

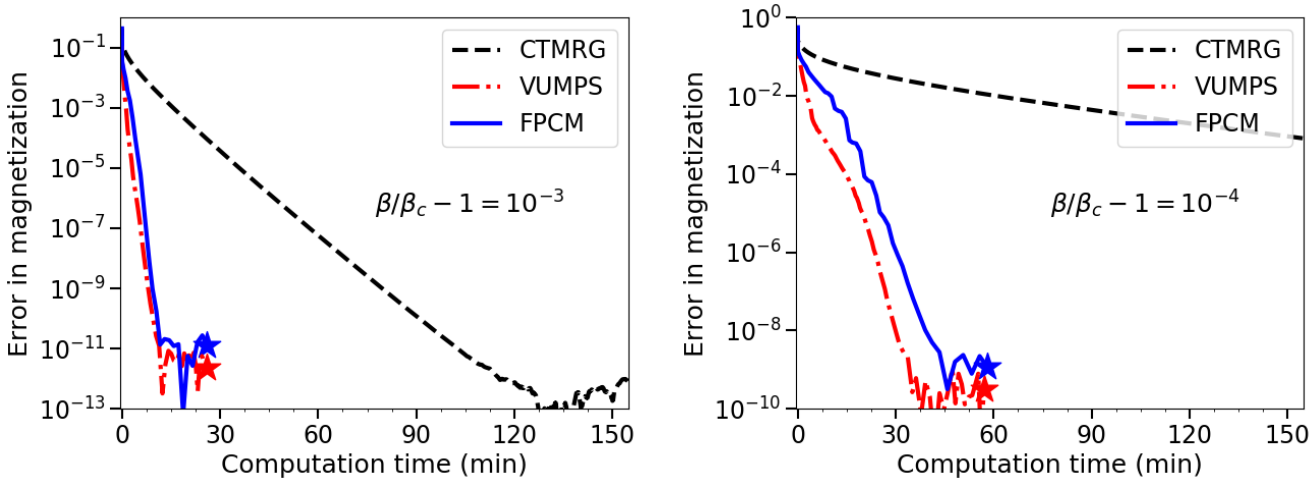
In Figure 3 we present results for the 2D ferromagnetic classical Ising model with a gauge transformation introduced on the links, as follows:




(23)

These gauge transformations, for random nonunitary matrices $X$ and $Y$, artificially break the rotation and reflection symmetries of the Ising model partition function. Gauge transformations like these can be introduced during an IPEPS optimization if explicit symmetries are not enforced, even if the state being targeted is expected to be rotationally symmetric. The environments we use for all methods are allowed to be complex. We see that the FPCM is both faster and allows for higher accuracy than the CTMRG approach.

In the CTMRG approach we compare to Ref. 33, where a pseudo-inverse of singular values is used as part of the process to obtain the projector which renormalizes the boundary. Because of small singular values, in practice this limits the accuracy of the method, in this example to around $10^{-7}$ (the smallest pseudo-inverse cutoff we tested for which the method would converge). We observed that this limit in accuracy could even occur if this nonsymmetric CTMRG algorithm was applied to a fully symmetric network ($X = Y = I$), because the method does not reduce to what is expected for a symmetric network (where the projectors are expected to reduce to isometries). Our method, as seen is Appendix A 1, does reduce correctly to the symmetric version when the network is symmetric, which explains why it might allow for higher accuracy (though in this example we had trouble obtaining accuracies below $10^{-8}$ due to unknown numerical instabilities). We should also note that to get consistent convergence we found it was important in the fully nonsymmetric FPCM to solve the bi-orthogonalization and fixed points to high accuracy early on in the calculation, which was not necessary for symmetric versions of the algorithm.




(a) (b)

Figure 1. Plots of the error in the magnetization for the isotropic 2D classical Ising model at two temperatures near criticality, where (b) is closer to criticality than (a). The network has a bond dimension of $d = 2$, and a boundary MPS bond dimension of $\chi = 600$ is used. A fully symmetric CTM ansatz is used for CTMRG and the FPCM, and full symmetry is exploited in VUMPS. The speedup of VUMPS and the corner method over CTMRG increases as one gets closer to criticality. Stars indicate the environment tensors have reached a fixed point, and data points beyond those points are numerical fluctuations and were not shown in order to simplify the plot.

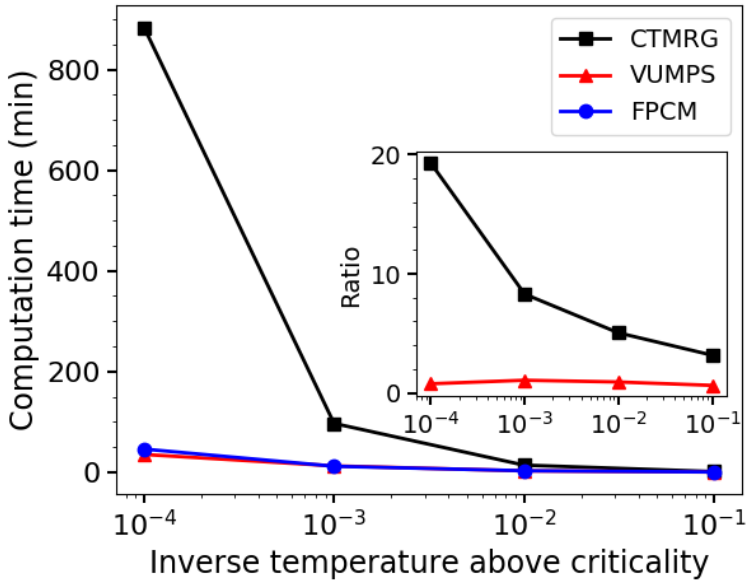
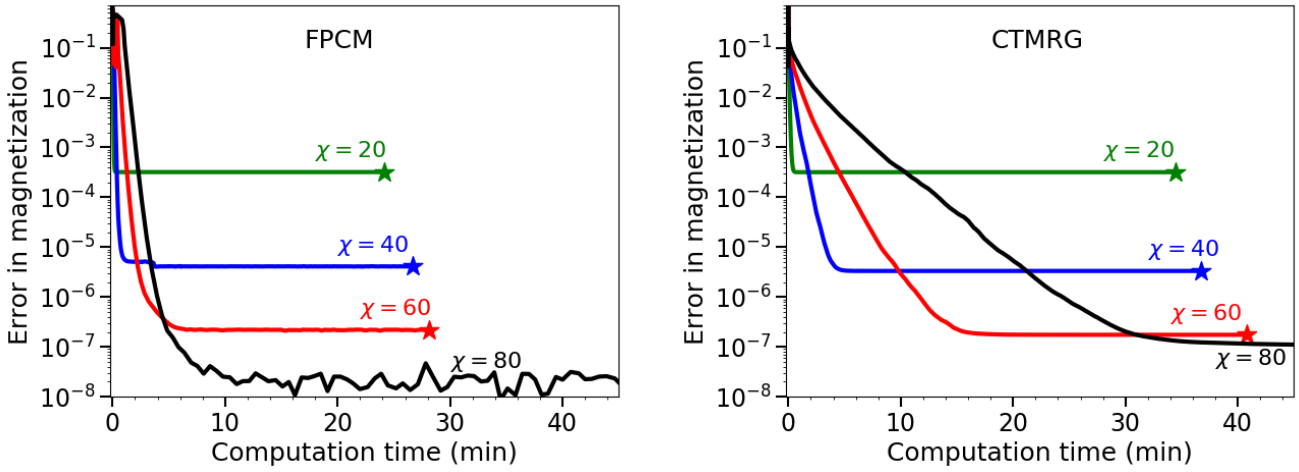



Figure 2. Convergence time as a function of inverse temperature above criticality, $\beta/\beta_c - 1$, for the 2D classical Ising model. For all data points, a boundary MPS bond dimension of $\chi = 600$ is used. All data is converged to an error in the magnetization of $< 2 \times 10^{-9}$. The inset shows the ratio of the convergence time of CTMRG and VUMPS with respect to the FPCM convergence time.

### B. 2D Classical XY Model

In Figure 4 we present results for contracting the partition function for the 2D classical XY model. Because the lattice degree of freedom is continuous for this model, the MPO tensor comprising the partition function can only be constructed approximately, though to high accuracy. The construction we use is described in previous work, where HOTRG was used to contract the partition function[45,46]. We refer readers to those previous references for details on constructing the MPO for this model. The link bond dimension we use is $d = 25$. We use an inverse




(a) (b)

Figure 3. Plots of error in magnetization for the isotropic ferromagnetic 2D classical Ising model at $\beta/\beta_c - 1 = 10^{-3}$ with random nonunitary gauge transformations introduced on the horizontal and vertical links, as shown in (23). This artificially breaks the lattice symmetry in order to test each method on an asymmetric network. (a) shows results for the FPCM introduced in this work, and (b) shows results for the most recently proposed nonsymmetric CTMRG algorithm[33]. Note that for this example the CTMRG algorithm is limited to an accuracy of $O(10^{-7})$. This is because the algorithm involves inverting the (square root) of the singular values of a matrix whose singular values become arbitrarily small at large bond dimension. Therefore, a pseudo-inverse with a cutoff of $10^{-7}$ was required in order to obtain convergence.

critical temperature 10% below the critical point estimated in that reference and zero applied magnetic field, where the model is known to be gapped. Since the U(1) symmetry cannot be broken at any finite temperature, we expect the magnetization to be zero.

The MPO tensor comprising the partition function is real and symmetric about reflections about the diagonals of the network, but not about the $x$ and $y$ axes. The environment we use for all methods is restricted to being real. For CTMRG and the FPCM, in the ansatz in (3) we impose $A_1 = A_2^T = A_3 = A_4^T = A$, $C_1 = C_2$, and $C_3 = C_4$, and additionally impose $C_1 = C_1^T$ and $C_3 = C_3^T$. For VUMPS, in (9) we don't impose any symmetries, but when we calculate observables we set the bottom fixed point uMPS equal to the transpose of the top fixed point uMPS obtained (such that the environment is invariant under a rotation about $\pi$). Again for CTMRG we can obtain the projectors using a symmetric diagonalization of the grown corner. We see that VUMPS performs noticeably worse than the FPCM, likely because the ansatz used for VUMPS cannot exploit the lattice symmetry as well as the CTM ansatz. We perform a few steps of CTMRG per step of the FPCM, which noticeably improves the convergence time for this example. We expect the improvement of the FPCM compared to CTMRG to become even more pronounced closer to criticality.

### C. 2D Quantum Heisenberg Model

In Figure 5 we present results for contracting a PEPS approximation to the ground state of the 2D quantum Heisenberg model. The PEPS tensor was optimized using the conjugate gradient method described in Ref. 26. We plot the error in the energy relative to the energy obtained from Monte Carlo simulations[47].

The PEPS tensor is complex and symmetric (not Hermitian) about all rotations and reflections, therefore so is the double layer MPO tensor that comprises the tensor network for the norm of the PEPS. The environment we use for all methods are necessarily complex. For CTMRG and the FPCM, in the ansatz in (3) we impose $A_1 = A_2 = A_3 = A_4 = A$, $C_1 = C_2 = C_3 = C_4 = C$, and additionally impose $C = C^T$. For VUMPS, in (9) we don't impose any symmetries[48], but when we calculate observables we set the bottom fixed point uMPS equal to the top fixed point uMPS obtained (not the conjugate of the top fixed point, as we would do if the MPO was Hermitian as opposed to complex symmetric). The CTMRG algorithm we use is a modification of the one from Ref. 33, where the symmetry of the network is exploited wherever possible. We perform a few steps of CTMRG per step of the FPCM, which we find improves the convergence time.

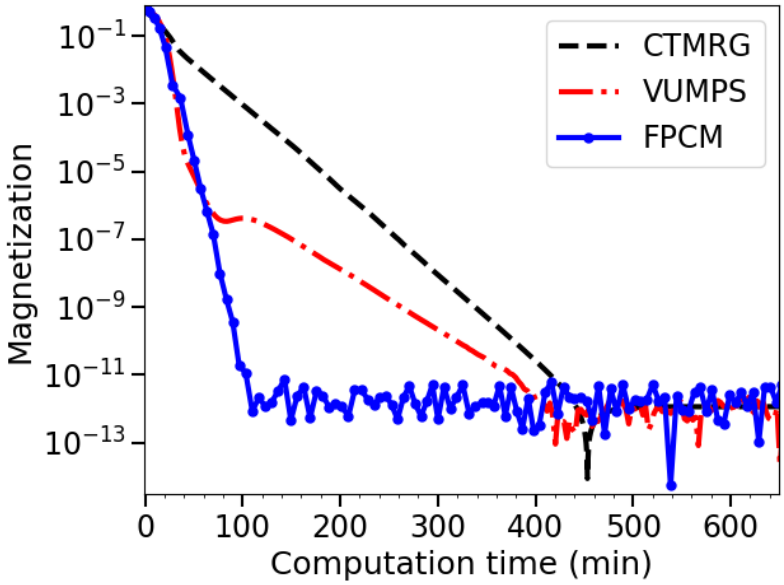



Figure 4. Plot of magnetization for the 2D classical XY model, for network bond dimension $d = 25$ and boundary MPS bond dimension $\chi = 50$.

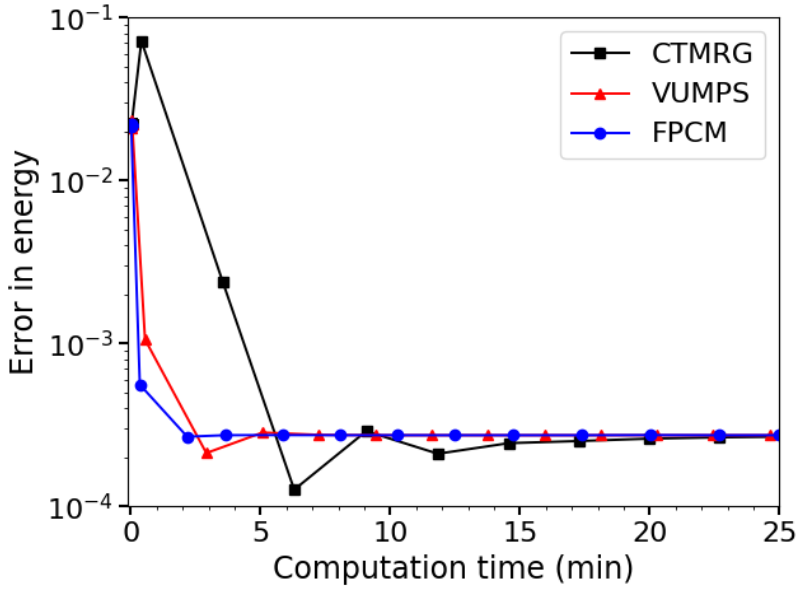



Figure 5. Plot of error in energy (compared to Monte Carlo results) for the 2D quantum Heisenberg model. The network bond dimension is $d = 25$ (or PEPS bond dimension $\sqrt{d} = 5$), and the MPS boundary bond dimension $\chi = 100$.

### D. Chiral Resonating Valence Bond PEPS

In Figure 6 we present results for contracting a chiral Resonating Valence Bond (RVB) PEPS, which was introduced as a chiral extension of the traditional nearest neighbor RVB PEPS[49,50]. As in the previous works on this model, we choose $\lambda_1 = \lambda_2 = \lambda_{\text{chiral}} = 1$, where $\lambda_2 = \lambda_{\text{chiral}} = 0$ would correspond to the non-chiral nearest neighbor RVB state. We refer to the previous works on this model for details on its derivation and physics.

The PEPS tensor (and therefore double layer MPO tensor) for this model is complex and Hermitian about reflections about the $x$ axis, $y$ axis, and diagonals. For CTMRG and the FPCM, in the ansatz in (3) we impose $A_1 = \bar{A}_2 = \bar{A}_3 = A_4 = A$ and $C_1 = \bar{C}_2 = C_3 = \bar{C}_4 = C$, and additionally impose $A^s = (A^s)^\dagger$ and $C = C^\dagger$. For VUMPS, in (9) we do not impose any symmetries[51]. When we calculate observables we set the bottom fixed

point uMPS obtained from VUMPS equal to the complex conjugate of the top fixed point uMPS obtained. For CTMRG, the projectors are obtained with a Hermitian diagonalization of the grown corner.

Again, we see an improvement in performance of the new fixed point corner method and VUMPS over CTMRG, but the new fixed point corner method performs better than VUMPS because the symmetry of the network can be exploited better in the CTM ansatz. We perform a few steps of CTMRG per step of the FPCM, which we find improves the convergence time.

## V. CONCLUSION AND OUTLOOK

We presented two new approaches for contracting 2D tensor networks, such as 2D classical partition functions and 2D quantum states represented as IPEPS. One approach uses the recently proposed VUMPS algorithm. The other approach uses the CTM ansatz like CTMRG, but improves upon CTMRG by solving for the boundary tensors with fixed point equations, which we refer to as the fixed point corner method (FPCM). With careful benchmarking we compared these new approaches to CTMRG for a variety of systems, which is currently the most widely used method for contracting 2D tensor networks in IPEPS calculations. We found that both methods improve upon the performance of CTMRG, however for certain models the improvement is more pronounced for our new fixed point corner method. We believe this is because, by using the CTM ansatz for the boundary, lattice symmetries can be better exploited by this method than can be exploited by VUMPS.

We showed that the improvement upon CTMRG is particularly pronounced as models approach criticality, as exemplified by our benchmarking of the 2D classical Ising model. This can be explained by the fact that, as the gap of the model closes, so too does the gap of the transfer operator. By solving for the boundary tensors with fixed point equations, methods like Arnoldi and Lanzos can be used, which are known to perform better than power methods for finding extremal eigenvectors of matrices with small gaps. Even though each step of the new approaches we present are often slower than each step of CTMRG, substantially fewer steps are required to reach fixed points leading to an overall improvement in the performance. Additionally, our new fixed point corner method is able to improve upon the accuracy of CTMRG for anisotropic 2D tensor networks because the inverse used for obtaining the projector is better conditioned than the one used by current CTMRG methods.

We are convinced that these new methods directly improve the performance of current state of the art IPEPS optimization techniques, where the contraction of the network is the most computationally expensive step. When combined with the variational methods for optimizing IPEPS that were recently introduced[25,26], we expect that significant improvements can still be made to existing IPEPS algorithms.

## ACKNOWLEDGMENTS

M.F. acknowledges helpful feedback on the presentation of the results from S.R. White and E.M. Stoudenmire. The authors gratefully acknowledge support from the National Science Foundation Graduate Research Fellowship Program (NSF GRFP) under Grant No. DGE-1144469 (M.F.), the Austrian Science Fund (FWF): F4104 SFB ViCoM and F4014 SFB FoQuS (V.Z.-S. and F.V.), and the European Research Council (ERC) under Grant No. 715861 (J.H.). J.H. and L.V. are supported by the Research Foundation Flanders (FWO). This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No 647905). This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562.

[1] V. Zauner-Stauber, L. Vanderstraeten, M. Fishman, F. Verstraete, and J. Haegeman, "Variational optimization algorithms for uniform matrix product states," arXiv:1701.07035.
[2] J. Haegeman and F. Verstraete, Annual Review of Condensed Matter Physics **8**, 355 (2017).
[3] R. J. Baxter, Journal of Mathematical Physics **9**, 650 (1968).
[4] M. Suzuki, Physics Letters A **146**, 319 (1990).
[5] M. Suzuki, Journal of Mathematical Physics **32**, 400 (1991).
[6] G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003).
[7] G. Vidal, Phys. Rev. Lett. **98**, 070201 (2007).
[8] T. Nishino, K. Okunishi, Y. Hieida, N. Maeshima, and Y. Akutsu, Nuclear Physics B **575**, 504 (2000).
[9] T. Nishino, Y. Hieida, K. Okunishi, N. Maeshima, Y. Akutsu, and A. Gendiar, Progress of Theoretical Physics **105**, 409 (2001).
[10] Y. Nishio, N. Maeshima, A. Gendiar, and T. Nishino, "Tensor Product Variational Formulation for Quantum Systems," arXiv:cond-mat/0401115.
[11] F. Verstraete and J. J. I. Cirac, "Renormalization algorithms for Quantum-Many Body Systems in two and higher dimensions," arXiv:cond-mat/0407066.
[12] T. Nishino, Journal of the Physical Society of Japan **64**, 3598 (1995).
[13] T. Nishino and K. Okunishi, Journal of the Physical Society of Japan **65**, 891 (1996).
[14] R. Orús and G. Vidal, Phys. Rev. B **78**, 155117 (2008).

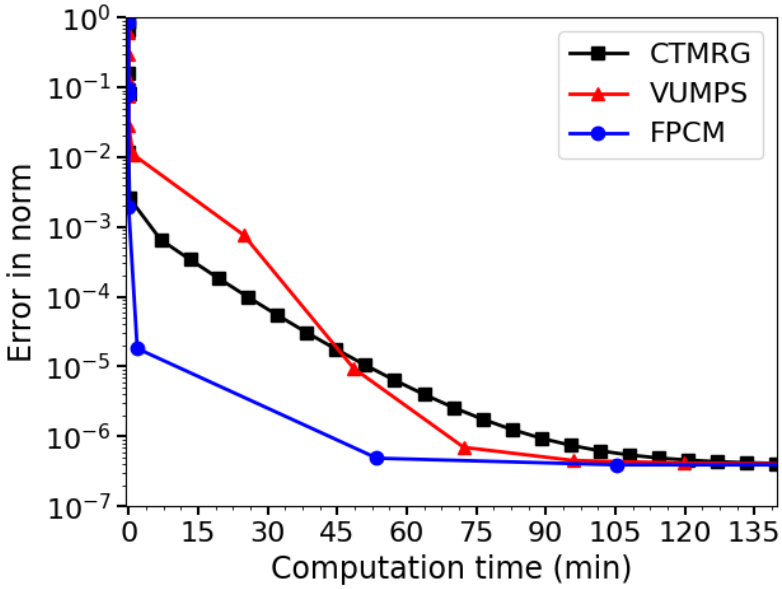



Figure 6. Plot of error in norm (extrapolated to large bond dimension) of the chiral RVB PEPS. The network bond dimension is $d = 9$ (or PEPS bond dimension $\sqrt{d} = 3$), and the boundary MPS bond dimension is $\chi = 800$.


[15] M. Levin and C. P. Nave, Phys. Rev. Lett. **99**, 120601 (2007).
[16] Z. Y. Xie, J. Chen, M. P. Qin, J. W. Zhu, L. P. Yang, and T. Xiang, Phys. Rev. B **86**, 045139 (2012).
[17] Z. Y. Xie, H. C. Jiang, Q. N. Chen, Z. Y. Weng, and T. Xiang, Phys. Rev. Lett. **103**, 160601 (2009).
[18] G. Evenbly and G. Vidal, Phys. Rev. Lett. **115**, 180405 (2015).
[19] S. Yang, Z.-C. Gu, and X.-G. Wen, Phys. Rev. Lett. **118**, 110504 (2017).
[20] M. Bal, M. Mariën, J. Haegeman, and F. Verstraete, Phys. Rev. Lett. **118**, 250602 (2017).
[21] S.-J. Ran, Phys. Rev. E **93**, 053310 (2016).
[22] S. R. White, Phys. Rev. Lett. **69**, 2863 (1992).
[23] S. R. White, Phys. Rev. B **48**, 10345 (1993).
[24] L. Vanderstraeten, M. Mariën, F. Verstraete, and J. Haegeman, Phys. Rev. B **92**, 201111 (2015).
[25] P. Corboz, Phys. Rev. B **94**, 035133 (2016).
[26] L. Vanderstraeten, J. Haegeman, P. Corboz, and F. Verstraete, Phys. Rev. B **94**, 155123 (2016).
[27] J. Jordan, R. Orús, G. Vidal, F. Verstraete, and J. I. Cirac, Phys. Rev. Lett. **101**, 250602 (2008).
[28] R. Orús and G. Vidal, Phys. Rev. B **80**, 094403 (2009).
[29] P. Corboz, J. Jordan, and G. Vidal, Phys. Rev. B **82**, 245119 (2010).
[30] P. Corboz, R. Orús, B. Bauer, and G. Vidal, Phys. Rev. B **81**, 165104 (2010).
[31] P. Corboz, S. R. White, G. Vidal, and M. Troyer, Phys. Rev. B **84**, 041108 (2011).
[32] R. Orús, Phys. Rev. B **85**, 205117 (2012).
[33] P. Corboz, T. M. Rice, and M. Troyer, Phys. Rev. Lett. **113**, 046402 (2014).
[34] H. N. Phien, J. A. Bengua, H. D. Tuan, P. Corboz, and R. Orús, Phys. Rev. B **92**, 035142 (2015).
[35] P. Corboz, Phys. Rev. B **93**, 045116 (2016).
[36] L. Wang, I. Pižorn, and F. Verstraete, Phys. Rev. B **83**, 134421 (2011).
[37] Y.-K. Huang, P. Chen, and Y.-J. Kao, Phys. Rev. B **86**, 235102 (2012).
[38] T. Nishino and K. Okunishi, Journal of the Physical Society of Japan **66**, 3040 (1997).
[39] V. Zauner, D. Draxler, L. Vanderstraeten, M. Degroote, J. Haegeman, M. Rams, V. Stojevic, N. Schuch, and F. Verstraete, New J. Phys. **17**, 053002 (2015).
[40] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr, Computing in Science & Engineering **16**, 62 (2014).
[41] L. Tagliacozzo, T. R. de Oliveira, S. Iblisdir, and J. Latorre, Physical review b **78**, 024410 (2008).
[42] F. Pollmann, S. Mukerjee, A. M. Turner, and J. E. Moore, Physical review letters **102**, 255701 (2009).
[43] B. Pirvu, G. Vidal, F. Verstraete, and L. Tagliacozzo, Physical review b **86**, 075117 (2012).
[44] V. Stojevic, J. Haegeman, I. McCulloch, L. Tagliacozzo, and F. Verstraete, Physical Review B **91**, 035120 (2015).
[45] Y. Liu, Y. Meurice, M. P. Qin, J. Unmuth-Yockey, T. Xiang, Z. Y. Xie, J. F. Yu, and H. Zou, Phys. Rev. D **88**, 056005 (2013).
[46] J. F. Yu, Z. Y. Xie, Y. Meurice, Y. Liu, A. Denbleyker, H. Zou, M. P. Qin, J. Chen, and T. Xiang, Phys. Rev. E **89**, 013308 (2014).
[47] A. W. Sandvik, Phys. Rev. B **56**, 11678 (1997).
[48] One may expect that we could set $A^s_R = (A^s_L)^T$ and $A^s_C = (A^s_C)^T$, however we were unable to get VUMPS to converge with these constraints imposed.
[49] D. Poilblanc, J. I. Cirac, and N. Schuch, Phys. Rev. B **91**, 224431 (2015).
[50] D. Poilblanc, N. Schuch, and I. Affleck, Phys. Rev. B **93**, 174414 (2016).

[51] One may expect that we could set $A^s_R = (A^s_L)^\dagger$ and $A^s_C = (A^s_C)^\dagger$, however we found in practice these relations only held up to diagonal phases, though this could possibly be fixed by some modification of the VUMPS algorithm.

[52] Y.-K. Huang, Journal of Statistical Mechanics: Theory and Experiment **2011**, P07003 (2011).

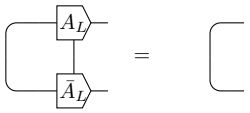
[53] Y.-K. Huang, P. Chen, and Y.-J. Kao, Phys. Rev. B **86**, 235102 (2012).

## Appendix A: Gauging uMPS

### 1. New Algorithm for gauging a uMPS

Starting with a uniform MPS comprised of the MPS tensor $A$, we would like to find the gauge in which the left fixed point of the MPS transfer matrix is identity from bra to ket (the "canonical" gauge). In other words, we would like to find $A_L$ and $C$ which satisfy:

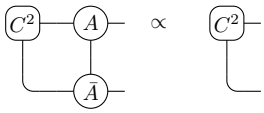

(A1)




(A2)

A method for finding the orthogonal gauge for a uMPS was first proposed in the context of the ITEBD algorithm[14] and involved an explicit inversion of the matrix $C$ to solve for $A_L$. Unfortunately this means that $A_L$ is generally only approximately isometric and the accuracy up to which this "pulling through" equation can be satisfied is limited for MPS with small singular values.

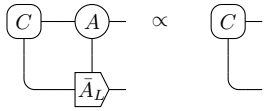
We now present a fast, robust and highly accurate alternative. Similar to previous methods, we start by finding the left fixed point which we suggestively call $C^2$ of the MPS transfer matrix:




(A3)

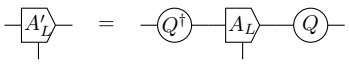
From properties of the transfer matrix we know that $C^2$ is a positive Hermitian matrix. We diagonalize it as $C^2 = U\Sigma^2 U^\dagger$ where $\Sigma^2$ are the singular values/positive eigenvalues of $C^2$. We therefore obtain $C = U\Sigma U^\dagger$, which is also a positive Hermitian matrix. We can now obtain $A_L$ from the uniqueness of the polar decomposition, by taking the polar decomposition of $CA^s = A^s_L P$. $|C-P|$ is our initial error. Because we took a square root of our singular values, our initial error is limited to the square root of machine precision, i.e. $O(10^{-8})$. If higher accuracy is required, we repeat the following steps until convergence:

1. Get the new corner matrix $C$ from the mixed transfer matrix of $A$ and $\bar{A}_L$ by approximately solving the fixed point equation:




(A4)

2. Get $A_L$ using the new $C$ from step 1. First take the (left) polar decomposition of $C = QC'$, where $C'$ is positive and Hermitian. Then we change the gauge of $A_L$ by $Q$, i.e.




(A5)

Then obtain the new $A_L$ from a polar decomposition of $C'A^s$ just as before.

### 2. New Algorithm for "bi-orthogonalizing" two uMPS

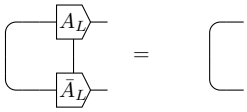
We now describe how to "bi-orthogonalize" two uMPS's (with single site unit cells) which are respectively comprised of MPS tensors $A$ and $B$. By bi-orthogonalize, we mean that we wish to find a gauge in which in one direction the fixed point of the mixed MPS transfer matrix formed from the two uMPS's are identity from bra to ket.

We begin by gauging $A$ and $B$ separately using the algorithm of the previous subsection, i.e. finding:

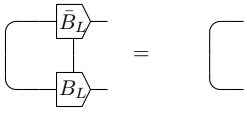

(A6)


(A7)

where $A_L$ and $B_L$ are isometric tensors satisfying

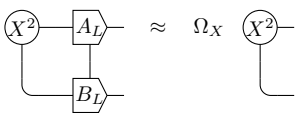



(A8)



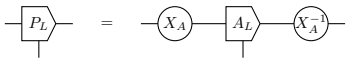

(A9)

Now we get the left fixed point $X^2$ of the mixed transfer matrix of $A_L$ and $B_L$:

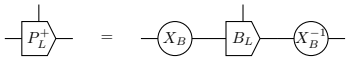



(A10)

We now take the SVD of $X^2 = U\Sigma^2 V^\dagger$. We define $X_A = \Sigma V^\dagger$ and $X_B = \Sigma U^T$ and define $P_L^{(+)}$ as follows:




(A11)




(A12)

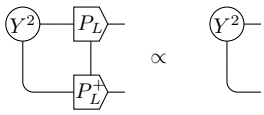
We also need to gauge transform $C_{A/B}$ with $X_{A/B}$, so we define $C^X_{A/B} = X_{A/B} C_{A/B}$. (This procedure can be viewed as a fixed point version of a similar bi-orthogonalization scheme proposed in previous works[52,53]).

Gauge transforming (A6), (A7) by $X_{A/B}$ and plugging in our definitions for $P_L^{(+)}$, and $C^X_{A/B}$, we see that:


(A13)


(A14)

Additionally, if we compute the following fixed point:



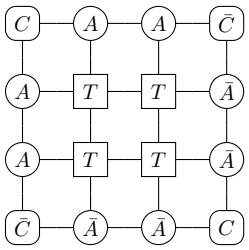

(A15)

we see that, by gauge transforming (A10) by $X_{A/B}$ and plugging in our definitions for $P_L^{(+)}$, then $Y^2 \approx I$ and $P_L$ and $P_L^+$ are approximate pseudo-inverses. We can improve the accuracy by taking the SVD of $Y^2 = U\Sigma^2 V^\dagger$, defining $Y_A = \Sigma V^\dagger$ and $Y_B = \Sigma U^T$, and gauge transforming $P_L$ by $Y_A$ and $P_L^+$ by $Y_B$ (and updating $C^X_{A/B} \to Y_{A/B} C^X_{A/B}$). This process can be repeated a number of times and is similar to the concept of "reorthogonalization" in standard Krylov subspace methods. Typically, a small number of repetitions (2 or 3) is advantageous for improving the accuracy of the orthogonalization, but a higher number of repetitions quickly leads to stagnation.

The accuracies of (A13) and (A14) are limited by the accuracy of calculating $X^{-1}_{A/B}$. If the singular values of $X^2$ are small, this process can be ill-conditioned and a pseudo-inverse should be used. In the limiting case where $B = \bar{A}$ (possibly up to a gauge transformation), $X^2$ is a unitary matrix, so the singular values are all unity and the bi-orthogonalization is well conditioned.

## Appendix B: Exploiting Symmetries

In this section we show how symmetries in the network can be exploited in the new FPCM presented in the main text. If the MPO tensor that composes the 2D tensor network of interest is Hermitian about reflections about the horizontal, vertical, and diagonals, the CTM ansatz can be simplified as follows:




(B1)

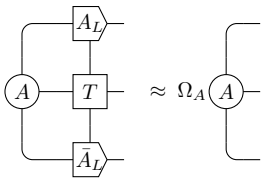
where we use the convention that the indices of non-conjugated tensors are ordered counterclockwise and the indices of conjugated tensors are ordered clockwise. We add the additional constraint that $C = C^\dagger$ and $A^s = (A^s)^\dagger$. With these constraints, the algorithm described in Section III C simplifies to the following:

1. Gauge the uMPS composed of tensor $A$. Using $A$, find isometric tensor $A_L$ and Hermitian matrix $C$ satisfying:


(B2)

   This is performed with the new uMPS gauging method described in Appendix A 1.

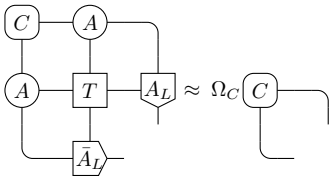
2. Obtain the new MPS boundary tensor $A$ using $A_L$ found in step 1. This is done by numerically solving the following fixed point equations (in practice using an iterative method like Arnoldi):




(B3)

Then, steps 1 and 2 are repeated until convergence. This is the simplest version of the algorithm, which we use for the isotropic 2D classical Ising model in Section IV A (which additionally is real) and the chiral RVB state in Section IV D. Again, as mentioned at the end of Section III C, performing a few steps of CTMRG per step of the FPCM can help improve convergence.

After convergence, a higher accuracy for observables can be obtained by using the corner transfer matrix $C$ calculated with the following fixed point equation:




(B4)

A similar fixed point equation for the corner transfer matrix was mentioned in Ref. 26. Corner transfer matrices for nonsymmetric networks can be obtained from generalizations of this equation.