When does entropy promote local organization? Electronic Supplementary Information

Andrei A. Klishin,[a,b,c] and Greg van Anders[a,b,d]

S1 Prior Results for Lattice Dimers

The lattice dimer model that we chose for our example has a long history of prior study across different disciplines. Physicists know it as "dimers", mathematicians as "tilings", and computer scientists as "perfect matchings".[1] The results from all these disciplines inform our study, as we show in this section.

The lattice dimer model was first introduced to model adsorption of diatomic molecules on surfaces in 1937 by Fowler and Rushbrooke.[2] Fowler and Rushbrooke quickly identified one of the main questions of the model: how many dimer configurations $\mathscr{Z}$ are there in a filled rectangular domain of $m \times n$ cells? Since the system is extensive, it is easy to show that for a large domain the scaling is exponential:

$$\mathscr{Z}(m,n) \sim \mu^{mn/2}, \tag{S1}$$

where $mn/2$ is the number of dimers, and μ is so-called *molecular freedom*, or the effective number of orientations per dimer. If the dimers did not constrain each other, they would independently take vertical or horizontal orientations, resulting in $\mu = 2$. Since dimers do constrain each other, $\mu < 2$, and Fowler and Rushbrooke aimed to explicitly compute this number.

They constructed their computation as an enumeration of the number of dimer tilings in a long narrow strip via constructing recursion relations. For width 2, the recursion produces the Fibonacci sequence with the estimate of $\mu \approx \phi \approx 1.618$, the golden ratio, but for wider strips the recursions had to be worked out by hand and took form of bulky multilinear expressions. Remarkably, these expressions could be solved numerically using the Mallock machine,[3] an early analog computer, up to width 6, roughly converging on $\mu \approx 1.8$. Molecular freedom calculations thus put the lattice dimer model at the dawn of computational physics.

After several improvements on the computation of μ,[4] its exact value was finally computed in early 1960s a series of papers by Fisher, Temperley, and Kasteleyn, thus "solving" the model.[5–7] The full solution bears considerable similarity to the Onsager expression for the critical 2D Ising model,[8] prompting further interest from the statistical physics community. Dimer tilings inspired the Resonating Valence Bond theory of superconductivity,[9,10] and brane tilings in string theory.[11] Other work focused on extensions of the dimer model to different domain geometries, computation of correlation functions, and finite-density dimers.

Dimers tilings don't have long-range order in *rectangular* domains,[12] but the tiling structure can be radically different in domains of a different shapes.[1] For instance, in the "Aztec diamond" domain (a square rotated by 45 degrees), the dimer configurations demonstrate the striking "Arctic circle" effect, with the dimers freezing into a brick pattern outside of the central circle.[13,14] This large-scale effect of boundary conditions was later explained within a much more general variational framework.[15]

Structure of dimer tilings is further elucidated with correlation functions, computed with coarse grained field theory mappings,[16] earlier Pfaffian methods,[12] or later Grassmanian algebra.[17,18] These techniques compute the correlations at both short and long distances, in the bulk as well as near the borders and corners of the domain. They also allow introducing a finite number of *monomers*, or unpaired/empty sites, thus discussing tilings at slightly less than 100% density.

Finite density tilings also attracted considerable interest. Rigorous mathematical proofs show that the dimer free energy does not exhibit singularities as a function of density, i.e. there are no ordering phase transitions.[19,20] As well, a variety of expansions have been developed to study finite-density free energy of the dimer system,[21–24] but they did not focus on short distance correlations.

Aggregating the lessons from prior studies, we can more precisely formulate the question for the present study, the method to address it, and the way to validate the answer. We are interested in the *local structures* (motifs) formed by dimers at *finite density*. The dimer correlations can be converted into DEFs via Landau-Gibbs free energy (see section below). We compute the correlations with an automated combinatorial method of tensor networks (see section below). In order to validate the tensor network method, we use it to compute the value of molecular freedom μ and compare it to the exact analytical result (see section below). As well, at full packing we compare our values of DEFs with the exact results from correlation functions.

S2 Tensor Network Construction

We cast the lattice dimer model into a spin-like form, in which each lattice site can be in one of five states s: empty e, left l, right r, top t or bottom b. The latter four states correspond to site occupation with half a dimer. Each half of a dimer needs to match with the corresponding other half, fulfilling a local constraint. Apart from these constraints, the site is biased to be empty or non-empty by regulation of the fugacity z, turning the computation into a weighted constrained summation problem (#SAT) which we solve numerically via tensor network contraction.

[a] Department of Physics, University of Michigan, Ann Arbor, MI 48109, USA
[b] Center for the Study of Complex Systems, University of Michigan, Ann Arbor, MI 48109, USA
[c] John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA
[d] Department of Physics, Engineering Physics, and Astronomy, Queen's University, Kingston ON K7L 3N6, Canada, E-mail: gva@queensu.ca

In a tensor network each node is a tensor of rank equal to the node degree, and each link is a contraction of two tensors by the corresponding indices. Main text Fig. 6 shows an example tensor network for computations in this paper that is based on the structure proposed in Ref.[39]. Our network consists of four types of tensors: site tensors, vertical and horizontal coupling tensors, and fugacity tensors. Each lattice site corresponds to a site tensor $\delta_{s_1 s_2 s_3 \dots}$ that is a high-rank generalization of Kronecker delta: its entries are 1 when all indices are identical and 0 otherwise. The purpose of this unit tensor is to ensure that couplings from all directions see the site in the same state, while simultaneously summing over all possible states. Because of simplicity of its construction, the rank of site tensor can be adjusted for different computations as explained below.

Each site is connected to its four nearest neighbors on the square lattice via the coupling tensors. A coupling tensor is equivalent to a transfer matrix used in solving other lattice models.[40] The possible configurations of lattice site states are either valid or invalid. For valid configurations, the coupling tensor has an entry 1, for invalid an entry 0. The coupling tensors are not symmetric, since exchanging the states of two adjacent sites does not preserve the validity. Moreover, the validity check is different for horizontally and vertically adjacent sites, so we construct the two different coupling tensors explicitly elementwise.

For a horizontal coupling $T^h_{s_1 s_2}$, if the left site is in "left" state, then the right site needs to be in the "right" state, and vice versa. This can be enforced by all other entries of the coupling in the corresponding row and column being 0. All other combinations of left and right state are allowed, with no relative preference between them. The resulting coupling can be written down as following:

$$T^h_{s_1 s_2} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix}, \qquad \text{(S2)}$$

where the index s_1 (rows) is contracted with the site to the left from coupling, and the index s_2 (columns) with the site to the right. Both indices run, in order, over site states $\{e, l, r, t, b\}$. The vertical coupling tensor $T^v_{s_1 s_2}$ is constructed analogously and can be written down as:

$$T^v_{s_1 s_2} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \qquad \text{(S3)}$$

where the index s_1 (rows) is contracted with the site to the bottom of the coupling, and the index s_2 (columns) with the site to the top.

Apart from the site and coupling tensors, we also include the chemical potential (external field), in form of rank-1 tensors f_s attached to each site. A generic site in grand canonical ensemble contributes a multiplicative factor z to the grand partition function if it is occupied and 1 if it's empty. Here $z \equiv e^{\beta\mu}$ is the fugacity associated with the particle bath and is the only thermodynamic free parameter used in the model. The corresponding fugacity tensor can be written as:

$$f_s = \begin{pmatrix} 1 & z & z & z & z \end{pmatrix}. \qquad \text{(S4)}$$

Having specified the tensors and their pattern of connections in the bulk of the lattice, we need to additionally specify the boundary conditions. We consider rectangular lattices with two types of boundary conditions: open and periodic. For open boundary conditions, the edge sites are only subject to three constraints, and corner sites are only subject to two constraints, allowing unpaired dimer halves on the domain boundary, effectively "sticking out" beyond the domain. For periodic boundary conditions, the sites on the right edge are coupled to the ones on the left, and the sites on the top edge are coupled to the bottom. For the validation computation we compare both types of boundary conditions (see section below), whereas for the free energy computation we only consider the periodic boundary condition. Boundary conditions conclude the specification of the tensor network so it can be contracted.

Contracting the tensor network results in a tensor of rank equal to the number of unpaired links. There are no unpaired links in the network of main text Fig. 6, so it contracts to a rank-0 tensor (i.e., a scalar number) equal to the partition function $\mathscr{Z}$. Computing other quantities from the tensor network requires modifying it slightly as described below.

We construct tensor networks on computer via automated Python scripts and then contract them with `PyTNR` package.[39,41] `PyTNR` automatically determines the order of contractions and combines them with Singular Value Decomposition (SVD) to reduce the tensor rank and dimensionality by discarding eigenvalues below a chosen threshold. We choose the threshold of 10^{-5} for most computations, thus making our method approximate. Our method is also non-deterministic since the order of contractions in `PyTNR` is somewhat stochastic. We compensate for the stochasticity by performing replica computations and computing error bars.

S3 Model Validation

We validate our tensor network approach by computing the molecular freedom μ. Molecular freedom is the scaling base of the partition function, related to the system free energy and entropy per site, and it has been computed exactly in Refs.[5–7] for an infinite rectangular domain. The exact result is $\mu = e^{2G/\pi} \approx 1.792$, where G is the Catalan constant. We compute a finite-size approximation from the tensor network model at full packing. We enforce full packing by setting the fugacity tensor to be $f_s = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \end{pmatrix}$ and thus prohibit empty sites. Since avoiding empty sites also requires the domain to have an even number of dimers, we consider rectangular domains of dimensions $(k) \times (k+1)$. For each domain size we compute the partition function $\mathscr{Z}(k, k+1)$ and extract a finite-size approximation to molecular freedom as following:

$$\mu_k^{BC} = \exp\left(\frac{\ln \mathscr{Z}(k, k+1)}{k(k+1)/2}\right), \qquad \text{(S5)}$$

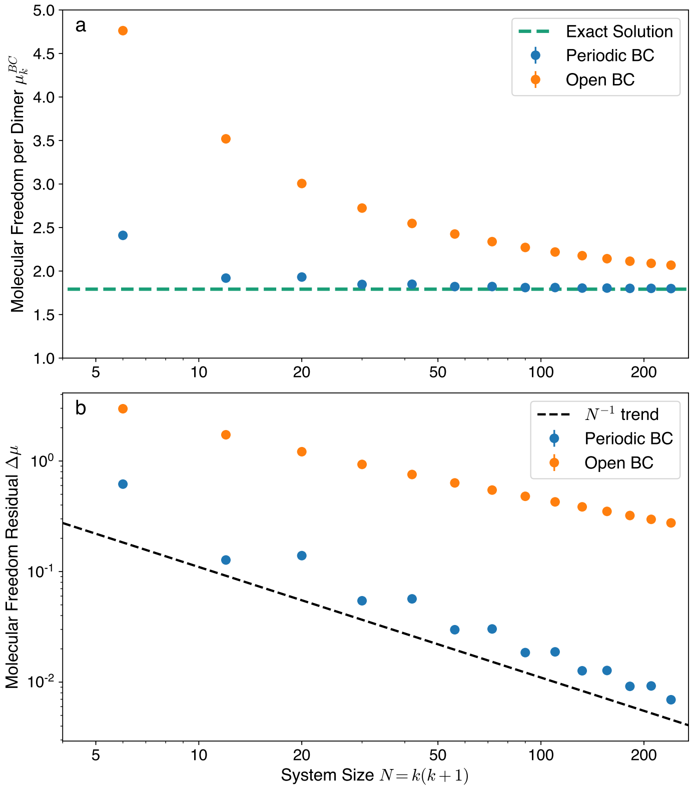

Fig. S1 Validation of the tensor network model via computation of the finite-size molecular freedom. (a) Scatter plots represent the tensor network computation of μ_k^{BC} with periodic (blue) and open (orange) boundary conditions, while the green horizontal dashed line denotes the analytic value $\mu \approx 1.792$. (b) Scatter plots represent the residual of the molecular freedom $\Delta\mu = (\mu_k^{BC} - \mu)$ on logarithmic scale, while the black dashed line denotes the N^{-1} trend line for visual comparison. The computation is performed on rectangular domains of size $(k) \times (k+1)$ for $k = 2..15$. All error bars are much smaller than marker sizes, at about 10^{-4} for open boundary conditions and 10^{-5} for periodic.

where *BC* denotes the boundary condition type (open or periodic), as described above.

Fig. S1a presents the validation computation results. The results for open boundary conditions consistently and significantly overcount the dimer configurations, since the dimers can "stick out" outside the domain. On the other hand, the results for periodic boundary conditions indicate that the molecular freedom nearly saturates the exact result for systems of size $N = k(k+1)$ of as small as 30.

Finiteness of the system implies that the computed quantities are subject to systematic finite-size errors of varying magnitude. We expect the error magnitude to increase for observables that marginalize over more variables, thus having the smallest error for molecular freedom μ (effectively a zero-point function), slightly higher for the density ρ (extracted from the one-point function), and the highest for the Landau-Gibbs free energies βG (extracted from the two-point function, see below). The scaling of the error of μ is shown to decay as $\mathcal{O}(N^{-1})$ in Fig. S1b. The slight step effect of the trend likely shows that μ is more sensitive to the even dimension of the computation domain than the odd dimension. We do not have analytic results for finite-size density ρ for comparison. The Landau-Gibbs free energies at the highest density of $\rho = 100\%$ can be compared to the exact analytic calculation of Fisher and Stephenson (horizontal arrows in Fig. 3b of the main text, see below on converting the notation conventions and finite-size scaling).[12]

The computation of molecular freedom at the largest system size of $15 \times 16 = 240$ sites involves tensor network enumeration of $5^{240} \approx \mathcal{O}(10^{167})$ distinct states. Since most of these states involve unmatched dimer halves, the number of configurations with nonzero weight is roughly $\mu^{240/2} \approx \mathcal{O}(10^{30})$. However, the computation at 100% density is actually not the hardest from the point of view of tensor network contraction. Since the states where any of the sites are empty can be easily dismissed in SVD, this regime is less taxing on memory than slightly smaller density. In order to balance the needs for computation tractability and reduction of finite size effects, we perform the computations at finite density for system size of $9 \times 10 = 90$ sites, the results of which are presented in the main text.

S4 Landau-Gibbs Free Energy

If a certain local motif of two dimers is more likely to be found than the two dimers separately, it is said to be stabilized by the DEFs. We can quantify the DEFs by computing the Landau-Gibbs (LG) free energy from the two free energies:

$$\beta G(s;x,y) = \beta G_{\mathsf{motif}}(s;x,y) - \beta G_{\mathsf{bare}}(s), \qquad \text{(S6)}$$

where $\beta G_{\mathsf{bare}}(s)$ is the *bare* free energy of a site in state s (site independent), and $G_{\mathsf{motif}}(s;x,y)$ is the *motif* free energy of the site (x,y) in state s, *given* a fixed reference dimer. Both free energies depend on the fugacity z. The prefactor β fixes units to convert between the energy scale and thermodynamic temperature.

Thermodynamically, we term the expression (S6) the Landau-*Gibbs* free energy to highlight the grand canonical nature of the dimer ensemble. In a grand canonical ensemble, we cannot move

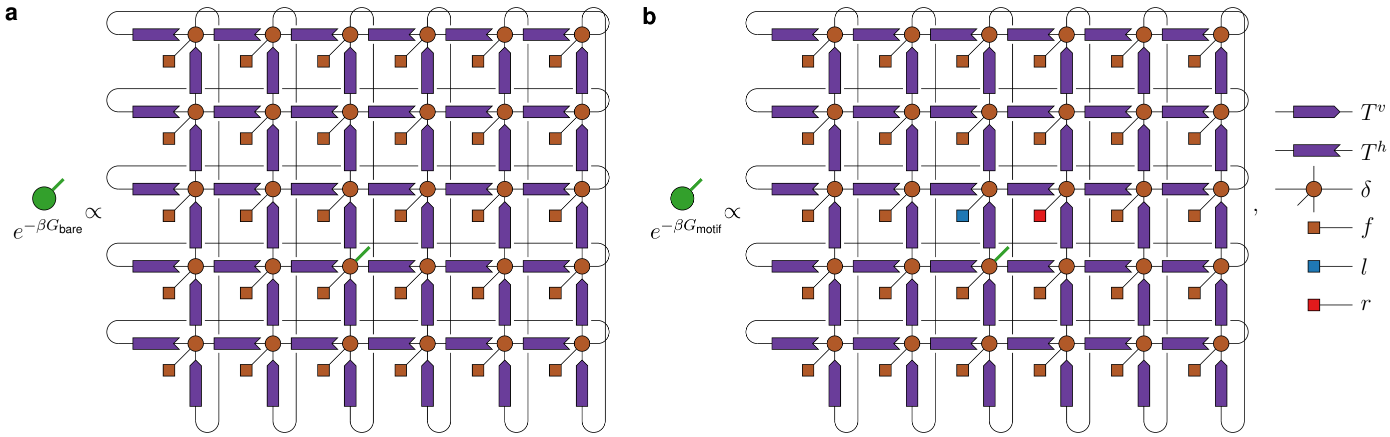
a
$e^{-\beta G_{\text{bare}}} \propto$
b
$e^{-\beta G_{\text{motif}}} \propto$
,
T^v
T^h
δ
f
l
r

Fig. S2 Schematic representation of tensor networks used for computations of free energies. (a) In order to compute the *bare* free energy, we add an external leg on an arbitrary site (green). Contracting this network results in the rank-1 tensor of $e^{-\beta G_{\mathbf{bare}}(s)}$ up to a normalization. (b) In order to compute the *motif* free energy, we replace the fugacity tensors on two central sites with the alternative tensors f^l and f^r (blue and red, respectively), and add external legs on various sites (x,y), one at a time. Contracting this network results in the rank-1 tensor of $e^{-\beta G_{\mathbf{motif}}(s;x,y)}$, up to a normalization.

a dimer to a different location; instead, we destroy it at an old location and create it anew at the new one. If it is easy to create a dimer at a certain location, given the placed reference dimer, the LG free energy would be negative. If it is hard to create a dimer, the LG free energy would be positive. If the difficulty of creating a dimer does not depend on the reference dimer, the LG free energy would be zero. Low free energy (high sea entropy) regions of space show the effective interaction potential of the two dimers. To better illustrate this potential and highlight the symmetries, we color-code the positions of the *centers of the dimers* rather than lattice sites in Fig. 3 of the main text. We choose the color map to be gray at the center value $\beta G = 0$, implying no attractive or repulsive effective dimer interactions. The dimer interactions disappear at either low density or large distance between the two dimers, implying that the placement of those dimers would become independent.

We can compute the effective dimer interactions by introducing modifications to the partition function expression (Eqn. 4 of the main text), or, equivalently, the tensor network form of partition function (Fig. 6 of the main text). In algebraic form, the two free energies can be computed via Eqns. 6-7 of the main text. The sums on the right hand side of the expressions evaluate to a vector, which we divide by the sum of its elements to ensure the normalization.

However, performing the computations in algebraic form is cumbersome, so we convert them into tensor network form. In order to *not* perform a sum in $s_{x,y}$ like in Eqn. 6 of the main text, we add an external leg to the corresponding site tensor in position (x,y) (green in Fig. S2a). The choice of the site (x,y) does not affect the values of G_{bare} because of the unbroken translational symmetry. Contracting that network results in a rank-1 tensor (vector) that contains the un-normalized, *unconditional* 1-point function. If we were to perform the remaining sum in $s_{x,y}$, we would recover the whole partition function $\mathscr{Z}$, which thus serves as the normalization.

In order to perform the conditional sum of Eqn. 7 of the main text, we need to implement $\delta(\text{ref})$ within the tensor network. In order to do that, we replace two fugacity tensors in the center of the network with, respectively, $f^l_s = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \end{pmatrix}$ and $f^r_s = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \end{pmatrix}$ (blue and red in Fig. S2b), and then add an external leg to different locations (x,y). The result of contraction is the un-normalized *conditional* 1-point function that depends on the chosen site (x,y). In order to reconstruct the full free energy landscape for a range of (x,y), we need to evaluate many of such 1-point functions, as detailed in Section S6 below. The normalization of the conditional 1-point function $\mathscr{Z}_{\text{cond}}$ is different from the original partition function $\mathscr{Z}$, but is independent of the site.

After computing the Landau-Gibbes free energies, we can compare them to the exact analytical results of Fisher and Stephenson at full packing.[12] Fisher and Stephenson used slightly different notation and reported the correlation function C_{ab} as arithmetic difference between the joint probability distribution of two dimers and the product of two marginal probabilities, normalized by the product of the latter. We convert their convention into ours as following:

$$\beta G^{FS}_{ab} = -\ln(1 + C_{ab}), \tag{S7}$$

where $a, b \in x, y$ represent the horizontal or vertical orientations of the dimers and the value of C_{ab} is taken directly from Table II in Ref.[12]. Fisher and Stephenson show the correlation function at the centers of the dimers, same as our convention in Fig. 3 of the main text, which allows us to identify the motifs for comparison to our Fig. 4 of the main text.

S5 Finite-Size Scaling

Fig. S3 shows the finite-size corrections to the LG free energy at 100% density. The LG free energies were largest by magnitude for the densely packed motif (gray, negative) and the horizontally shifted motif (green, positive), and the same free energies have the largest residuals. By making a linear fit and extrapolating it, we show that the LG free energies converge to the exact values of Fisher and Stephenson, and the finite-size correction is at the order of $\mathscr{O}(N^{-1})$, same as for the molecular freedom μ.

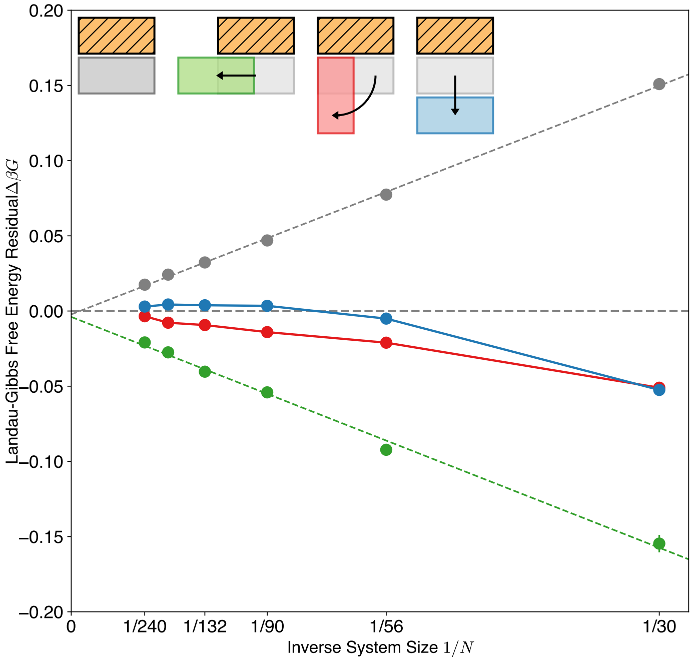

Fig. S3 Finite-size scaling of the residual of the LG free energy $\Delta\beta G = (\beta G - \beta G^{FS})$ with system size at 100% density. The inset shows the color coding of the four local motifs, same as in the Fig. 4 of main text. For the dense-packed (gray) and horizontally shifted (green) motifs linear fits in N^{-1} are shown, for the other two the data points are just connected with straight lines. Error bars are 10^{-3} or smaller and are mostly smaller than the markers.

S6 Multi-Marginalizing

While the tensor network contractions are highly optimized, they are the bulk of our computation. To make this computation efficient, it is desirable to extract maximal amount of useful information from the minimal number of contractions. In this section we explain how we minimize the number of contractions by taking advantage of the local and global symmetries of the Landau-Gibbs free energy field, as well as the *multi-marginalizing technique* to efficiently evaluate many conditional two-point functions, which then behave as one-point functions.

The motif free energies $G_{\text{motif}}(s;x,y)$ allow us to evaluate the Landau-Gibbs free energies for the placement of a dimer in position (x,y) with respect to the fixed dimer at any given fugacity z. The free energies of finding two halves of a dimer have to be equal due to the way the coupling tensors T^h and T^v are set up, forming local translational symmetries:

$$G_{\text{motif}}(l;x,y) = G_{\text{motif}}(r;x+1,y),$$
$$G_{\text{motif}}(b;x,y) = G_{\text{motif}}(t;x,y+1). \tag{S8}$$

Because of these symmetries, each site can supply information about the one-point functions on the site itself and two adjacent neighbors. By convention, on the site itself we use the values $G_{\text{motif}}(l)$ and $G_{\text{motif}}(b)$, whereas for the neighbors to the left and to the bottom we use the values $G_{\text{motif}}(r)$ and $G_{\text{motif}}(t)$, respectively. Exploiting this redundancy makes it sufficient to only com-

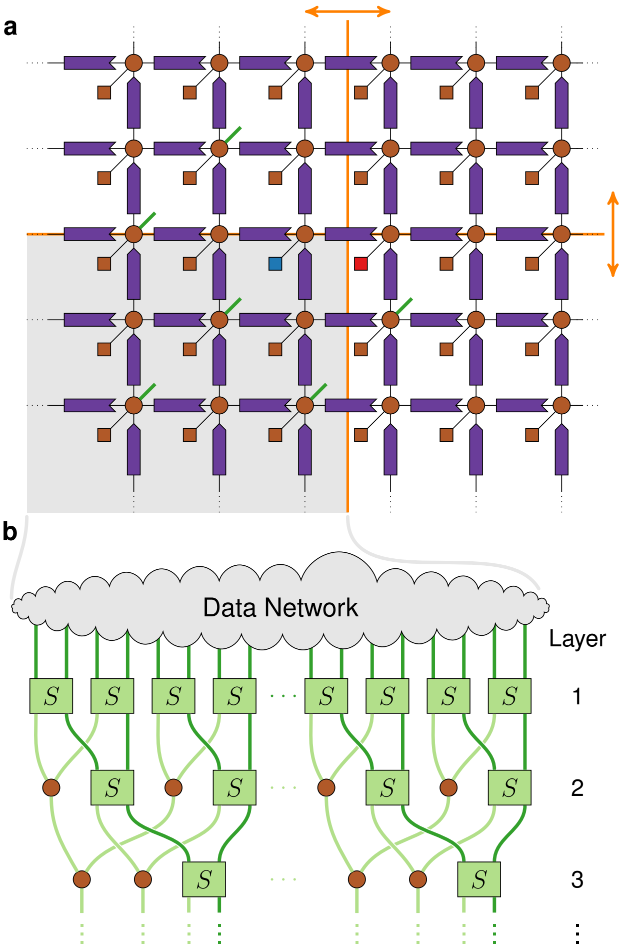

Fig. S4 Multi-marginalizing efficiently computes many one-point functions in a single tensor contraction. (a) The tensor network that encodes the dimer model, color coding is identical to that in Figs. 6 (main text) and S2. The system has global two-fold reflection symmetries with respect to the central vertical and horizontal axes (orange lines and arrows); therefore, it is only necessary to compute the one-point functions from one quadrant, here bottom-left (shaded gray). Site tensors in that quadrant have external legs attached in a checkerboard pattern (dark green lines) to take advantage of the symmetry (S8). (b) The tensor network that models the physical system can be abstracted as a "data network". We can be agnostic about the internal contents of the data network and only characterize it by the output "data legs" (dark green lines). Direct contraction of the data network gives an n-point function. In order to filter out the correlations from the n-point function, we connect the data legs to the multi-layer readout network. In layer 1, the data legs are connected to the tops of the S tensors (Eq. (S9)) in pairs. The data legs from the bottom of the S tensors are connected to the S tensors of the next layer, while the logic legs (light green lines) are connected to the Kronecker tensors of the next layer. The lowest layer exposes one output data leg and multiple logic legs that allow reading out marginalized 1-point functions from different input data legs.

pute one-point functions on half of all the sites in a checkerboard pattern. Apart from this pattern, we use the global four-fold symmetry of the domain, ensured by the central placement of the fixed dimer, to only explicitly compute one quadrant of the lattice (shaded gray in Fig. S4, top).

Even with these symmetry reductions, we still need to compute many conditional one-point functions. Naively, we can compute the n-point function by creating n external legs on different lattice sites and further marginalize it. However, intermediately storing the n-point function requires 5^n units of memory, which quickly becomes prohibitively expensive. At the same time, all the extra information about site correlations in the full n-point function is unnecessary for the DEF analysis. For this analysis, we need to filter out the cross-correlations and instead compute *many one-point functions* in a single contraction.

We do this by using the method of multi-marginalizing, presented in Fig. S4.* In this method, the tensor network of sites, couplings, and external fields (Fig. S4a) is abstracted out as a *data network*, which can potentially represent any model. Apart from the data network, we create a separate *readout network* (Fig. S4b) as a superstructure on top of the model-driven data network that filters out the cross-correlations between the readout legs, and outputs the one-point functions in a more compact array. While the n-point function is an array with 5^n elements, the requisite n one-point functions only require $5 \cdot 2^m$ elements, where $m = \log_2 n$, thus saving lots of memory in filtering.

The main component of this filtering is the rank-4 switch tensor $S^{ab}_{c\alpha}$. a, b, c are called *data legs* and have dimension 5, while the new index $\alpha \in \{0, 1\}$ is a binary *logic leg* of dimension 2. The upper indices are presented in the direction of the data network, the lower indices in the direction of the output (see Figure). The logic leg regulates the behavior of the rest of the legs in the following way:

$$S^{ab}_{c\alpha} = \begin{cases} \delta^a_c \mathbf{1}^b, & \alpha = 0 \\ \delta^b_c \mathbf{1}^a, & \alpha = 1 \end{cases}, \tag{S9}$$

where δ is the regular Kronecker delta over the corresponding indices and $\mathbf{1}$ is a vector of all 1's. Depending on the value of the binary index α, either one or the other data leg is marginalized (contracted with $\mathbf{1}$). When the indices α and c are left free, the contraction of this tensor with two data legs gives a 5×2 array that stores the two marginal probability distributions, i.e. two one-point functions.

In order to extract more one-point functions, we build the whole readout network out of multiple switch tensors S and Kronecker tensors δ, arranged in multiple layers. Layer 1 consists only of S tensors (Fig. S4b) and reduces the number of data legs in half. Layer 2 connects the data legs from layer 1 to new S tensors, and the logic legs from layer 1 to a Kronecker tensor, thus reducing the number of data legs in half again. Layer 3 and so forth can be constructed in a similar fashion. The final layer m consists of one S tensor and $m-1$ Kronecker tensors that together expose one data leg and m logic legs. Choosing the binary index value of each logic leg specifies the binary address of the input data leg and thus allows us to read out the one-point function from a target site. A readout network of m layers thus allows addressing $n = 2^m$ sites. It is most convenient to extract one-point functions in number of a power of two. However, one can create a larger readout network and render some of the binary addresses mute by tracing out the unnecessary data legs (contracting them with $\mathbf{1}$).

Increasing the number of readout legs and the size of the readout network carries an overhead of computational time, so far poorly characterized. To minimize the total computational time, it is better to extract a finite number of one-point functions per contraction, and later stitch the multi-marginalizing results together. Multi-marginalizing doesn't have rigorous prescriptions on the optimal number of legs and layers, but for our system size we find heuristically that using $n = 8$ legs works the best.

S7 Supplementary Discussion

Our system of lattice dimers had unphysically low amounts of sea entropy, yet entropy still promoted local order. One might ask, is there a model that is even less physical with even less sea entropy in which entropy ceases to promote local order altogether?

One possible further reduction is to impose additional restrictions on the dimer orientations, such as the tatami constraint (no more than three dimers not allowed to meet at a single corner). However, introducing the tatami constraint makes the tilings extremely sensitive to boundary conditions, in many cases allowing no more than 1 or 2 configurations.[42] While this small number of configurations allows resolving deterministic logical expressions,[42] it is *subextensive* in system size, unlike in the simple dimer model, and thus gets rid of sea entropy. On the entropy axis of Fig. 1 of the main text, the tatami model would lie even further to the right, but it doesn't have DEFs anymore. The discovery of DEFs in lattice dimers thus on one side drastically expands and unifies the range of systems that can entropically order, and on the other side establishes a useful lower bound on this range, searching beyond which is not meaningful.

Notes and references

1 R. Kenyon, *Directions in mathematical quasicrystals, CRM Monogr. Ser*, 2000, **13**, 307–328.

2 R. H. Fowler and G. S. Rushbrooke, *Trans. Faraday Soc.*, 1937, **33**, 1272–1294.

3 R. Mallock, *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 1933, **140**, 457–483.

4 T. S. Chang, *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 1939, **169**, 512–531.

5 P. Kasteleyn, *Physica*, 1961, **27**, 1209 – 1225.

6 H. N. V. Temperley and M. E. Fisher, *Philosophical Magazine*, 1961, **6**, 1061–1063.

7 M. E. Fisher, *Physical Review*, 1961, **124**, 1664.

8 L. Onsager, *Physical Review*, 1944, **65**, 117.

* Personal communication and implementation of A.S. Jermyn as part of PyTNR package.

9 D. S. Rokhsar and S. A. Kivelson, *Phys. Rev. Lett.*, 1988, **61**, 2376–2379.

10 S. Kivelson, D. Rokhsar and J. Sethna, *EPL (Europhysics Letters)*, 1988, **6**, 353.

11 K. D. Kennaway, *Int. J. Mod. Phys.*, 2007, **A22**, 2977–3038.

12 M. E. Fisher and J. Stephenson, *Physical Review*, 1963, **132**, 1411.

13 W. Jockusch, J. Propp and P. Shor, *arXiv preprint math/9801068*, 1998.

14 H. Cohn, N. Elkies and J. Propp, *arXiv preprint math/0008243*, 2000.

15 H. Cohn, R. Kenyon and J. Propp, *Journal of the American Mathematical Society*, 2001, **14**, 297–346.

16 R. Kenyon, *Annals of probability*, 2001, 1128–1137.

17 N. Allegra, *Nuclear Physics B*, 2015, **894**, 685–732.

18 N. Allegra and J.-Y. Fortin, *Physical Review E*, 2014, **89**, 062107.

19 O. J. Heilmann and E. H. Lieb, *Physical Review Letters*, 1970, **24**, 1412.

20 O. J. Heilmann and E. H. Lieb, *Statistical Mechanics*, Springer, 1972, pp. 45–87.

21 G. Rushbrooke, H. Scoins and A. Wakefield, *Discussions of the Faraday Society*, 1953, **15**, 57–65.

22 J. F. Nagle, *Physical Review*, 1966, **152**, 190.

23 R. J. Baxter, *Journal of Mathematical Physics*, 1968, **9**, 650–654.

24 D. S. Gaunt, *Physical Review*, 1969, **179**, 174.

25 N. Creignou and M. Hermann, *Information and computation*, 1996, **125**, 1–12.

26 J. D. Biamonte, J. Morton and J. Turner, *Journal of Statistical Physics*, 2015, **160**, 1389–1404.

27 S. Kourtis, C. Chamon, E. R. Mucciolo and A. E. Ruckenstein, *arXiv preprint arXiv:1805.00475*, 2018.

28 R. Penrose, *Combinatorial mathematics and its applications*, 1971, **1**, 221–244.

29 F. Verstraete, V. Murg and J. I. Cirac, *Advances in Physics*, 2008, **57**, 143–224.

30 U. Schollwöck, *Annals of Physics*, 2011, **326**, 96–192.

31 R. Orús, *Annals of Physics*, 2014, **349**, 117–158.

32 G. K.-L. Chan and S. Sharma, *Annual review of physical chemistry*, 2011, **62**, 465–481.

33 Z. Xie, H. Jiang, Q. Chen, Z. Weng and T. Xiang, *Physical review letters*, 2009, **103**, 160601.

34 M. Levin and C. P. Nave, *Physical review letters*, 2007, **99**, 120601.

35 E. Stoudenmire and D. J. Schwab, Advances in Neural Information Processing Systems, 2016, pp. 4799–4807.

36 A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, D. P. Mandic *et al.*, *Foundations and Trends® in Machine Learning*, 2016, **9**, 249–429.

37 A. Cichocki, A.-H. Phan, Q. Zhao, N. Lee, I. Oseledets, M. Sugiyama, D. P. Mandic *et al.*, *Foundations and Trends® in Machine Learning*, 2017, **9**, 431–673.

38 I. V. Oseledets, *SIAM Journal on Scientific Computing*, 2011, **33**, 2295–2317.

39 A. S. Jermyn, *arXiv preprint arXiv:1709.03080*, 2017.

40 N. Goldenfeld, *Lectures on phase transitions and the renormalization group*, Addison-Wesley, Reading MA, 1992.

41 A. S. Jermyn, *Journal of Computational Physics*, 2019, **377**, 142–154.

42 A. Erickson and F. Ruskey, International Workshop on Combinatorial Algorithms, 2013, pp. 140–149.