Predicting ground state configuration of energy landscape ensemble using graph neural network

Seong Ho Pahng¹,∗ and Michael P. Brenner²,³

¹Department of Chemistry and Chemical Biology, Harvard University, Cambridge, MA 02138
²School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138
³Google Research, Mountain View, CA 94043

(Dated: August 20, 2020)

Many scientific problems seek to find the ground state in a rugged energy landscape, a task that becomes prohibitively difficult for large systems. Within a particular class of problems, however, the short-range correlations within energy minima might be independent of system size. Can these correlations be inferred from small problems with known ground states to accelerate the search for the ground states of larger problems? Here, we demonstrate the strategy on Ising spin glasses, where the interaction matrices are drawn from protein contact maps. We use graph neural network to learn the mapping from an interaction matrix J to a ground state configuration, yielding guesses for the set of most probable configurations. Given these guesses, we show that ground state configurations can be searched much faster than in vanilla simulated annealing. For large problems, a model trained on small J matrices predicts a configurations whose energy is much lower than those obtained by simulated annealing, indicating the size generalizability of the strategy.

Finding the ground state configurations of a complex energy landscape is a long standing computational challenge [1]. Short of brute-force enumeration, random search algorithms such as simulated annealing can anneal Markov chains to the global minimum as a simulation temperature approaches zero [2]. However, in cases where many interacting degrees of freedom result in a highly rugged energy landscapes, conventional methods suffer from low probability of overcoming energy barriers and the chain may get stuck in local minima [3–5].

The classical methods for searching energy landscapes are devised to work for general problems. Yet, many scientific problems often present themselves via an ensemble of energy landscapes with similar underlying patterns, with interactions arising from a single or handful of governing equations. Examples include the energy landscapes of organic molecules built out of chemical building blocks, where potential energies are obtained by solving Schrödinger equation, or the space of protein structures from interactions of individual amino acids. In an ensemble setting, we hypothesize that there exist system-specific sampling rules [6–8] that make it possible to traverse these particular energy landscapes more efficiently than classical methods. These rules can be learned from examples of energy minima calculated with classical methods for small problems.

Here we demonstrate this approach in the context of a model problem that defines a natural ensemble. We construct Ising spin glasses [9, 10], where the interaction matrix J is a structured random matrix, chosen from protein contact maps. Given the large database of natural proteins [11] and the distinctive contact pattern of a folded protein [12], protein contact map data gives an ideal ensemble for testing whether interaction rules encoded in J’s are consistent across varying system sizes.

In recent years, several machine learning techniques have been applied to sample spin configurations of Ising model. The list includes but is not limited to simple regression [13], restricted Boltzmann machine [14], reinforcement learning [15], autoregressive model [16, 17], and normalizing flow model [18]. The goal of these works is to estimate the Boltzmann distribution of a given problem so that the learned model can either completely replace or be used as a proposal distribution for Markov Chain Monte Carlo simulation. However, these schemes do not consider learning with J instances of varying sizes.

Instead we recast spin glass energy minimization, a well known NP-hard problem [19], as a node classification problem in graph theory, and employ a graph neural network (GNN) [20] to parametrize the mapping from a J to the corresponding ground state configuration. We generate the set of most probable configurations from the GNN model to predict low-lying configurations of an energy landscape. If this configuration set misses a ground state configuration, we show that simulated annealing starting from a configuration in this set can search for the ground state configuration more efficiently. The schematic of this strategy is described in Fig. 1.

We further test the utility of the GNN model by constraining the size of J—where size refers to the number of amino acid—in a training set and testing the trained model on larger J’s. As we increase the size limit of training set J from 30 to 500, model’s test performances quickly reach the level comparable to those obtained with the size limit of 800. We also show that the model trained on J with size less than 800 can predict configurations whose energies are much lower than those found by simulated annealing for J with size around 3000.

We begin by constructing an ensemble of Hamiltonians for which the underlying potential energy landscapes have similar patterns. For simplicity, we consider Ising
Hamiltonians of the form

\[
\mathcal{H}(\sigma) = -\frac{1}{2} \sum_{i,j}^N J_{ij} \sigma_i \sigma_j + h \sum_i^N \sigma_i, \quad h = \frac{\sum_J J_{ij}}{2N}
\]  

(1)

where both coupling and field terms depend on an interaction matrix \( J \) and \( J_{ij}, \sigma_i \in \{0,1\} \). The field is chosen to prevent all ground state configurations from collapsing to a trivial ground state of all 1’s. Within this formulation, to obtain an energy landscape ensemble, we need to specify an ensemble of \( J \) matrices, whose \( J \) is random yet with distinct shared patterns. Due to binary \( J_{ij} \), this structured randomness of \( J \) shall be encoded in spin connectivity.

In this work, we obtain this ensemble using protein contact maps to calculate the \( J \) ensemble. Since proteins are characterized by distinct secondary structures, together with non-local contacts, protein contact maps define a set of structured random connectivity matrices. We downloaded first subunit of all protein structure files deposited in the Protein Data Bank [21] to ensure \( J \)'s have the patterns of connectivity from natural protein folds. We train graph neural network with \( \sigma_{\text{min}} \) found from simulated annealing. The model aggregates the nearest-neighbor information for all spins at each layer. Thus an \( L \)-layer model can account for \( L \)-hop neighborhood information. As the model learns the rule of local interaction, it predicts a configuration which, if not the ground state already, can be improved by simple configuration enumeration and Monte Carlo sampling.

Functions Eq. 1 have no relation to amino sequences; our intent here is not to make predictions about proteins per se, but instead to use the regularity of protein structures to define a natural ensemble.

For all \( J \)'s except the two largest, we ran simulated annealing for each \( J \) starting from 100 random initial configurations, and selected an annealed configuration with the lowest energy as its purported ground state configuration \( \sigma_{\text{min}} \). The annealing schedule was optimized such that simulated annealing always find the ground state configurations on \( J \)'s with size smaller than 30, which we identified from brute-force enumeration of all configurations. For the two largest \( J \)'s, we decreased a cooling rate and increased an equilibration steps at each temperature to account for an enlarged configurational space and ran 30 randomly initialized simulated annealing. Further simulation details are discussed in the Supplemental Material [23]. Since finding the global energy minimum of an Ising spin glass in \( 2^N \) configuration space is NP-hard, we settled for this repeated annealing scheme and assume \( \sigma_{\text{min}} \) closely approximates the actual ground state configuration. From all pairs of \( J \) and \( \sigma_{\text{min}} \), 6400 pairs were randomly selected as validation set, another 6400 pairs as test set, and remaining 51763 pairs as training set. For the first size generalizability experiment, we used the same test set but sub-select from the training set the pairs whose \( J \) are smaller than certain size cutoffs to make small-\( J \) training sets. For the second size generalizability experiment, we used the entire training set and test on the two large \( J \)'s.

Our prediction task is to learn the mapping from \( J \mapsto \sigma_{\text{min}} \). This can be cast as a node classification problem in graph theory and hence we parametrize the mapping with a graph neural network. Given a graph, the \( L \)-layer model generates an expressive feature embedding for node \( \sigma_i \) by aggregating the features of all \( L \)-hop neighbor nodes of \( \sigma_i \) as shown in Fig. 1, and uses this em-
bedding to classify globally whether each node shall be
turned on or off. To allow generalization of the mapping
across J's with different size and structure, we chose a
message passing framework [24, 25] with attention mech-
anism [26, 27], instead of Laplacian-based convolution
method [28, 29] which requires a constant graph struc-
ture.

The inputs to the graph neural network are the ad-
jacency matrix J and node features, which are initially
a node degree the field strength \( h \) from Eq. 1. At each
layer, the network updates node features by first apply-
ing standard nonlinear transformation—expanding fea-
ture dimension from 2 to \( F \), then calculating attention
coefficient \( \alpha_{ij} \) to find relative importance of a neigh-
bor node \( j \) to node \( i \), and taking weighted sum of neighbor
nodes’ features using these coefficients. To capture more
information from neighbors, this process is repeated \( K \)
times with different set of weights and newly computed
\( \alpha_{ij}^K \) to produce \( K \times F \) features for a node. The features
of node \( i \) are then reduced to a probability \( P(\sigma_i=1) \) in the
final layer for node classification. The functional forms of
the operations are detailed in the Supplemental Material
[23]. Notably, the final model used in this work consists
of six layers.

The model’s predicted configuration, \( \hat{\sigma} \), are then ob-
tained by choosing the greater of the two node classifica-
tion probabilities, \( \text{argmax}_{\sigma} [P(\sigma_i=0) \ P(\sigma_i=1)] \). However,
this point estimation does not take a full advantage of the
learned embedding. This scheme is especially problem-
atic for nodes with probabilities around 0.5 because the
non-argmax configurations would have been just as likely.
Therefore, we generate a set of top most probable config-
urations from the configuration probability output of the
model, giving a broader coverage of the low-lying region
in the energy landscape. To obtain \( M \) such configura-
tions, we pick top \( \log_2 M \) nodes whose \( P(\sigma_i=1) \) are close
to 0.5, and order all permuted configurations according
to their corresponding sum combination of probabilities.

A GNN model trained on the entire training set cor-
rectly predicted \( \sigma_{min} \) for 1700 of 6400 J's in the held-out
test set. To further quantify the model’s performance, we
investigate following two metrics. Define accuracy as the
ratio between the number of correctly predicted nodes in
\( \hat{\sigma} \) and total number of nodes and energy difference \( \Delta E \) as
the energy gap between a predicted configuration and the
the true ground state. Fig. 2(a) shows the prediction accu-
ry decreases, while the energy difference increases with
increasing \( J \). The average accuracy and \( \Delta E \) across the
entire ensemble are 0.978 and 2.79 respectively, due to
the size distribution of \( J \) skewed towards small \( J \)'s (Sup-
plemental Material Fig. S1 [23]). Since energy histograms
of small \( J \)'s obtained via complete configuration enum-
eration are peaked at positive energy and negative energy
configurations occur in far-left tail region (Supple-
mental Material Fig. S2 [23]), predicting configurations with
such small energy differences is surprising. We emphasize
again that the model does not evaluate the energy func-
tion of Eq. 1 to optimize a configuration. This suggests
the GNN model has learned a generalizable node feature
transformation for this particular class of energy land-
scape simply by comparing its predicted configurations
to known ground state configurations.

Fig. 2(b) shows the averaged histogram of node classi-
fication probability \( P(\sigma=1) \) from high accuracy configu-
trations in blue, and that of low accuracy configurations
in orange. A striking feature is that most nodes in both
cases are predicted with high certainty as evinced by the
peaks at both ends. In addition, the histogram of low
accuracy configurations shows more nodes in the mid-
dle, indicating that the model’s prediction accuracy may
be directly related to the node classification probability
\( P(\sigma) \). We thus set a threshold probability, \( P_{thr} \), to se-
lect nodes with low uncertainty where \( P(\sigma_i=1) \geq P_{thr} \) or
\( P(\sigma_i=1) < 1-P_{thr} \) and calculated an error rate among
these nodes as \( P_{thr} \) is varied. As shown in the inset of
Fig. 2, the number of misclassified nodes among such
nodes goes down as we increase the threshold. This re-
sult in turn confirms that most misclassifications indeed
occur among uncertain nodes in the middle region of the
histogram.

Given there are only a handful of uncertain nodes, the
set of top most probable configurations can account for
most of permutations of their node configurations be-
cause the first few nodes to be changed are those with
\( P(\sigma_i=1) \approx 0.5 \). This enumerated set allows for coverage
of configuration space around the model's initial predic-
tion. We enumerated top 1000 most probable configura-
tions for each \( J \) in the test set to cover 10 most uncertain
nodes since 1000≈2^{10}. We then calculated the energy of
these configurations and picked the lowest energy con-
figuration as an improved prediction of the model, \( \hat{\sigma}_{\text{top}} \).
TABLE I. Summary of the size generalizability experiment on test set. Accuracy, energy offset, and the number of ground state match in 6400 test set $J$’s using the GNN’s prediction $\hat{\sigma}$, the lowest-energy configuration of top most probable set $\sigma_{\text{top}}$, and seeded annealing $\sigma_{\text{anneal}}$ are reported.

| Size cutoff | $\# J$’s | $\hat{\sigma}$ | $\sigma_{\text{top}}$ | $\sigma_{\text{anneal}}$ | $\# \sigma_{\text{min}}$ found |
|-------------|-----------|----------------|---------------------|---------------------|---------------------|
| 30          | 560       | 14 0.866 23.84 | 150 0.880 17.22     | 2824 0.926 3.68     | 2988                 |
| 40          | 1301      | 65 0.901 11.05 | 260 0.909 8.68      | 2756 0.923 3.64     | 3081                 |
| 50          | 1839      | 388 0.944 5.15 | 1079 0.954 3.00     | 2383 0.932 2.68     | 3850                 |
| 100         | 7319      | 511 0.953 4.12 | 1265 0.962 2.23     | 2272 0.932 2.74     | 4048                 |
| 200         | 24589     | 784 0.964 3.20 | 1398 0.972 1.65     | 2222 0.942 1.85     | 4404                 |
| 300         | 36130     | 1442 0.972 2.79 | 1638 0.977 1.32     | 1454 0.938 1.93     | 4534                 |
| 400         | 43190     | 1497 0.974 2.47 | 1687 0.980 1.18     | 1503 0.943 1.45     | 4687                 |
| 500         | 47631     | 1519 0.976 2.36 | 1738 0.981 1.12     | 1463 0.947 1.28     | 4720                 |
| 800         | 51763     | 1700 0.978 2.31 | 1673 0.983 1.13     | 1417 0.953 1.07     | 4790                 |

FIG. 3. Number of sampling steps taken to reach ground state configuration for simulated annealing launched from a random configuration (blue) and from the lowest-energy configuration of top most probable configurations (orange). Each point reports the averaged value from 10 trials for random annealing and 5 trials for seeded annealing. We also include the minimum number from the 5 trials for seeded annealing experiment (green).

From this procedure, we additionally found the ground state configurations in 1673 $J$’s. This improvement of $\hat{\sigma}$ by configuration enumeration suggests that the uncertain nodes contain frustrated nodes to which the configuration energy is highly sensitive and, thus, that the GNN model has an implicit representation of energy in the node embedding.

For the half of test set where our model missed the ground state configurations, all predicted configurations still have small energy differences relative to the ground state configurations. We exploited this by running 5 simulated annealings with $\hat{\sigma}_{\text{top}}$ as a starting configuration for each remaining $J$. Since we are now annealing from a low-lying point in energy landscape, the starting temperature of the annealing should concurrently decreased to prevent the chain from sampling arbitrarily high energy states. We used the temperature value at which the energy trajectory of sampled states drifts up to a bit higher energy at the beginning to allow for initial exploration of energy landscape [23]. The $\hat{\sigma}_{\text{top}}$-seeded simulated annealing found the ground state configuration for additional 1417 $J$’s with about two orders of magnitude reduction in the averaged number of sampling steps as shown in Fig. 3. In about 20% cases, the minimum number of sampling steps from 5 trials were only few hundreds as only one or two node were misclassified in $\hat{\sigma}_{\text{top}}$. In total, we found ground state configurations for 75% of test set $J$’s. This seeded simulated annealing result shows that the predicted configuration falls in the vicinity of $\sigma_{\text{min}}$, which is often close enough that simulated annealing can locate $\sigma_{\text{min}}$. Given the top most probable configurations, we could also run simulated annealing from other configurations or perform parallel tempering [30, 31] with multiple configurations to account for the possibility of $\hat{\sigma}_{\text{top}}$ falling in a basin that is too far away from the one containing $\sigma_{\text{min}}$.

To test for size generalizability of GNN model, we first trained models on eight small-$J$ training sets with increasing size cutoffs and test them on the existing test set. As shown in Table I, test set accuracy and energy difference roughly reach those of the original model trained on the entire training set when the size cutoff for $J$ is above 300. Note that the number of the match between the annealed configurations, $\sigma_{\text{anneal}}$, and $\sigma_{\text{min}}$ decreases as the size constraint increases because most of $\sigma$’s are already recovered through model predictions. In all cases, the configuration enumeration and seeded simulated annealing improve upon initial model predictions. If $\hat{\sigma}$, $\sigma_{\text{top}}$, and $\sigma_{\text{anneal}}$ are considered together, GNN model provides comparable performance even at the size cutoff of 200. The local interaction pattern of 6-hop neighbor networks in a protein shorter than 100 amino acids should be similar enough to that of much longer chain. It is thus likely that relatively poor performances with training sets with size cutoff less than 100 are simply due to a limited amount of available data.
To test this hypothesis in more practical use case setting, we tested the original model on $J$’s of size 3661 and 2814. The model predicted $\hat{\sigma}$ with energy -443 and configuration enumeration further improved the energy to -447 for $J$ of size 3661, whereas the lowest energy configuration found from 30 randomly initialized simulated annealing runs was -432. On $J$ with size 2814, we obtained energy values of -366 for $\hat{\sigma}$ and -370 for $\sigma_{\text{top}}$ whereas randomly initialized simulated annealing only reached down to -362. As in previous analysis, we launched simulated annealing from $\sigma_{\text{top}}$ and obtain annealed configurations with energy -463 and -391 for $J$ with size 3661 and 2814, respectively. Fig. 4 highlights the efficiency of the GNN model over randomly initialized simulated annealing.

Our work shows that it is indeed possible to use an ensemble of energy landscapes with known ground state configurations to train a neural network to deduce the ground state configuration of similar energy landscapes. On our model problem, we deterministically found the ground state configurations on 50% of the held out test set $J$’s and stochastically on additional 25% through a graph neural network, top configuration enumeration, and seeded simulated annealing. Although this number may appear modest, we emphasize that all configurations predicted by the model were extremely low-lying configurations, often in the vicinity of the ground state configurations. Since the loss function does not include other local minima—or, for that matter, the energy function itself, we believe that such an informed prediction is possible only if the learned node feature embedding of GNN correctly captures the local interaction rule encoded in $J$ interaction matrices, and hence the topological undulation in configuration space.

In addition, we showcased the practical utility of the GNN model with size generalizability experiments. The GNN model predicted the configurations that could not be reached by naive simulated annealing with random initial guess, and we were able to further improve it by combining the enumeration scheme and a seeded simulated annealing. Therefore, the GNN model presents an appealing method to produce extremely good initial guesses for a class of energy landscape problem where governing physics is local.

In future work, we will apply this framework to a variety of problems where the discovery of global minima would have technological consequences.

We thank Lucy Colwell for suggesting protein contact maps as a model system for ensembles of energy landscapes, and for helpful discussions. We also thank Yohai Bar-Sinai, Carl Goodrich, Mor Nitzan, Mobolaji Williams and Jong Yeon Lee for helpful discussions. This work is supported by the Office of Naval Research through the grant N00014-17-1-3029, as well as the Simons Foundation.

To test this hypothesis in more practical use case setting, we tested the original model on $J$’s of size 3661 and 2814. The model predicted $\hat{\sigma}$ with energy -443 and configuration enumeration further improved the energy to -447 for $J$ of size 3661, whereas the lowest energy configuration found from 30 randomly initialized simulated annealing runs was -432. On $J$ with size 2814, we obtained energy values of -366 for $\hat{\sigma}$ and -370 for $\sigma_{\text{top}}$ whereas randomly initialized simulated annealing only reached down to -362. As in previous analysis, we launched simulated annealing from $\sigma_{\text{top}}$ and obtain annealed configurations with energy -463 and -391 for $J$ with size 3661 and 2814, respectively. Fig. 4 highlights the efficiency of the GNN model over randomly initialized simulated annealing.

Our work shows that it is indeed possible to use an ensemble of energy landscapes with known ground state configurations to train a neural network to deduce the ground state configuration of similar energy landscapes. On our model problem, we deterministically found the ground state configurations on 50% of the held out test set $J$’s and stochastically on additional 25% through a graph neural network, top configuration enumeration, and seeded simulated annealing. Although this number may appear modest, we emphasize that all configurations predicted by the model were extremely low-lying configurations, often in the vicinity of the ground state configurations. Since the loss function does not include other local minima—or, for that matter, the energy function itself, we believe that such an informed prediction is possible only if the learned node feature embedding of GNN correctly captures the local interaction rule encoded in $J$ interaction matrices, and hence the topological undulation in configuration space.

In addition, we showcased the practical utility of the GNN model with size generalizability experiments. The GNN model predicted the configurations that could not be reached by naive simulated annealing with random initial guess, and we were able to further improve it by combining the enumeration scheme and a seeded simulated annealing. Therefore, the GNN model presents an appealing method to produce extremely good initial guesses for a class of energy landscape problem where governing physics is local.

In future work, we will apply this framework to a variety of problems where the discovery of global minima would have technological consequences.

We thank Lucy Colwell for suggesting protein contact maps as a model system for ensembles of energy landscapes, and for helpful discussions. We also thank Yohai Bar-Sinai, Carl Goodrich, Mor Nitzan, Mobolaji Williams and Jong Yeon Lee for helpful discussions. This work is supported by the Office of Naval Research through the grant N00014-17-1-3029, as well as the Simons Foundation.
[15] T. A. Bojesen, Phys. Rev. E 98, 063303 (2018).
[16] D. Wu, L. Wang, and P. Zhang, Phys. Rev. Lett. 122, 080602 (2019).
[17] B. McNaughton, M. Milošević, A. Perali, and S. Pilati, Phys. Rev. E 101, 053312 (2020).
[18] G. S. Hartnett and M. Mohseni, arXiv preprint arXiv:2001.00585 (2020).
[19] F. Barahona, J. Phys. A 15, 3241 (1982).
[20] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, IEEE Trans. Neural Netw. Learn. Syst. (2020), https://doi.org/10.1109/TNNLS.2020.2978386.
[21] “RCSB Protein Data Bank,” .
[22] B. Monastyrsky, D. d’Andrea, K. Fidelis, A. Tramontano, and A. Kryshtalovych, Proteins 82, 138 (2014).
[23] See supplemental material at .
[24] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, IEEE Trans. Neural Netw. 20, 61 (2008).
[25] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, (2017), arXiv:1704.01212.
[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, (2017), arXiv:1706.03762.
[27] P. Velikovi, G. Cucurull, A. Casanova, A. Romero, P. Li, and Y. Bengio, (2017), arXiv:1710.10903.
[28] M. Defferrard, X. Bresson, and P. Vandergheynst, (2016), arXiv:1606.09375.
[29] T. N. Kipf and M. Welling, (2016), arXiv:1609.02907.
[30] R. H. Swendsen and J.-S. Wang, Phys. Rev. Lett. 57, 2607 (1986).
[31] D. J. Earl and M. W. Deem, Phys. Chem. Chem. Phys. 7, 3910 (2005).
Supplemental Material for “Predicting ground state configuration of energy landscape ensemble using graph neural network”

SIMULATION DETAILS

For each sampling step of simulated annealing, we randomly selected a single node from a current node configuration vector $\sigma_{\text{cur}} \in \{0, 1\}^N$ and changed the node value to 1 if it is 0 or vice versa to generate a proposal configuration $\sigma_{\text{prop}}$, and accepted this configuration with probability $A$ given by the Metropolis criterion

$$A = \min\left(1, e^{-\beta(H(\sigma_{\text{prop}}) - H(\sigma_{\text{cur}}))}\right) \quad (S1)$$

where $\beta$ is an inverse temperature $1/T$ as we set Boltzmann constant to 1 and $H$ is the Ising Hamiltonian from the main text.

Calibration of annealing schedule

We used $n$-bit Gray code algorithm to enumerate all $\{0, 1\}$-node configurations and identify ground state configurations for $J$’s with size ranging up to 30. Using an exponential cooling schedule, $T_K = T_0 \cdot 0.8^K$, with the initial temperature $T_0=10$ and exponentially increasing equilibration steps, $L_K = L_0 \cdot 1.2^K$, with the initial number of steps $L_0=1000$, 10 randomly initialized simulated annealing runs all reached those ground state configurations for $J$’s with size smaller than 30. We let the temperature cycle $K$ go up to 30. We used this annealing schedule to calculate purported ground state configuration $\sigma_{\text{min}}$ for remaining $J$’s with size ranging up to 800, repeating the simulation 100 times with different seed configurations for each $J$. Additionally, we ran several simulated annealing runs with slower cooling and longer equilibration on 50 randomly selected $J$’s with size larger than 500 but these did not improve over the lowest energy annealed configurations found from the original experiments.

Annealing schedule for large $J$

On the two largest $J$’s with size 3661 and 2814, all 100 simulations with the aforementioned schedule annealed to different configurations and some had relatively high energies compared to the rest, indicating a poor annealing. We thus modified the cooling schedule to $T_K = T_0 \cdot 0.85^K$ with the same initial temperature yet with $K$ going up to 40 and the equilibration schedule to $L_K = L_0 \cdot 1.15^K$ with $L_0=2000$. Using this schedule, we were able to anneal to low energy configurations in all 30 randomly initialized simulated annealing runs. The energies of initial configurations and annealed configurations are shown as blue and orange dots in Fig. 4 of the main text.

Annealing schedule for seeded simulated annealing

Since $\sigma_{\text{top}}$ has low energy and is thus likely in the vicinity of $\sigma_{\text{min}}$, we modified the cooling schedule and equilibration length to focus our search on the configurational space near $\sigma_{\text{top}}$—i.e., exploit rather than explore. We kept the factor of 0.8 in the exponential cooling but decreased the initial temperature $T_0$ to 0.5, which was high enough that configurations with energy higher than $\sigma_{\text{top}}$ are accepted for all $J$’s in the test set. For the equilibration, we decreased the initial number of steps $L_0$ to 100 while keeping the exponential form of the length schedule. Although it is certainly possible to optimize the schedule per individual $J$ basis and find $\sigma_{\text{min}}$ for more $J$’s and quicker, we did not pursue this further as it deviates from the scope of this work.

SIZE DISTRIBUTIONS OF J

Fig. S1 shows the size distributions of $J$ in the entire ensemble and test set. Note that the ensemble histogram resembles the length distribution of first subunit of all proteins in Protein Data Bank that have more than 20 but fewer than 800 amino acids. The histogram of the test set $J$ resembles the full histogram, as we would expect for a random split.

![FIG. S1. Size histogram of entire J ensemble (A) and test set J’s (B). They contain 64563 and 6400 J’s, respectively.](image)

ENERGY HISTOGRAMS ON SMALL J

Fig. S2 show the histogram of energy values of three randomly selected $J$’s with size 28, calculated using all configurations found via brute-force enumeration with 28-bit Gray code generator. This can be viewed as the density of states for the five instances of spin glass Hamiltonian of Eq. 1 of the main text. Note that negative energy values occur several standard deviations away from the center, which is at a positive energy. We expect this distributional form to hold in bigger $J$’s, yet with larger spread due to more contacts in $J$. In $J$ with size 3661, for example, the distribution seems to have a peak around...
times, each with different $\mathbf{W}$, $\mathbf{a}$, and newly computed $\alpha$. The $K$ sets of hidden features are concatenated in input and middle layers to generate $K \cdot M'$ features,

$$f'_i = \left\| \sum_{j \in \{i, \mathcal{N}_i\}} \frac{\mathbf{a}_i^T \mathbf{W} \mathbf{f}_j}{\| \mathbf{W} \mathbf{f}_j \|} \right\|$$  \hspace{1cm} (S4)

and averaged in the final layer to retain the feature dimension $F$ which is reduced to 2, representing $P(\sigma_i=0)$ and $P(\sigma_i=1)$,

$$\bar{f}_i^{\text{out}} = \frac{1}{K} \sum_{k=1}^{K} \sum_{j \in \{i, \mathcal{N}_i\}} \frac{\mathbf{a}_i^T \mathbf{W} \mathbf{f}_j}{\| \mathbf{W} \mathbf{f}_j \|}$$  \hspace{1cm} (S5)

We then apply sigmoid activation to the output matrix $\mathbf{f}^{\text{out}}$ to obtain configuration probability matrix, $\mathbf{P} = \{\bar{P}_1, \bar{P}_2, \ldots, \bar{P}_N\}$ where $\bar{P}_i = [P(\sigma_i=0) \ P(\sigma_i=1)]^T$.

We trained the model to minimize the binary cross-entropy between the configuration probabilities and $\sigma_{\text{min}}$ found via simulated annealing, and optimized model hyperparameters on the 6400 validation set. We used Adam optimizer with an initial learning rate of 0.002 and a batch size of 16. The final model consisted of six graph attention layers—5 hidden layers implementing Eq. S4, each with $K=4$ attention heads and $F=128$ node features, and a final layer implementing Eq. S5.

**EFFECT OF LAYER DEPTH ON TEST SET PERFORMANCE**

As shown in the previous section, each graph attention layer aggregates information from 1-hop neighbors. Hence, a depth of model controls the field of view we employ in learning the local interaction rules. Fig. S3 summarizes how a model’s test set performance change as we increase its layer depth. We noticed that going past 6 layers give negligible improvement. From a network analysis perspective, it would be a worthwhile effort to probe whether this saturation of learnability at 6-hop neighborhood hints to a cluster property intrinsic to protein contact maps.
Fig. S3. Accuracy (a) and energy difference between \( \hat{\sigma} \) and \( \sigma_{\min} \) (b) over averaged size window of 100 on test set \( J \) at varying graph attention layer depths. The curves for 6-layer model are identical to those shown in Fig. 2 (a) of the main text. Error bars are omitted for comparison purpose.

**ADDITIONAL RESULTS FROM SIZE GENERALIZABILITY EXPERIMENT**

Table S1 reports detailed breakdowns of the test set accuracy and the energy difference of models trained on small-\( J \) training sets. *over* or *under* refers to a given metric averaged on all test set \( J \)'s whose sizes are over or under the size cutoff applied during model training.

| size cutoff | Accuracy | \( \Delta E \) |
|-------------|----------|----------------|
|             | under    | over          | under    | over          |
| 30          | 0.881199 | 0.871548      | 2.450821 | 21.263792     |
| 40          | 0.927850 | 0.899662      | 1.076943 | 11.178737     |
| 50          | 0.968899 | 0.944050      | 0.228221 | 5.177260      |
| 100         | 0.975699 | 0.950044      | 0.496598 | 4.471919      |
| 200         | 0.973282 | 0.946137      | 0.913523 | 5.005585      |
| 300         | 0.973081 | 0.955576      | 1.192279 | 4.803279      |
| 400         | 0.970069 | 0.955870      | 2.192192 | 8.932947      |
| 500         | 0.970202 | 0.951427      | 2.037306 | 8.844054      |

**TABLE S1.** Test set size generalizability results with additional criteria of over or under training set size cutoffs. The number of \( J \)'s at each size cutoff is reported in the table of main text.

Fig. S4 shows the two metrics at each size window of test set \( J \)'s, with finer size cutoff spacing from size 30 to 50. The degradation of model performance is less severe as a training set size cutoff increases. The model appears to generalize well starting from the size cutoff of 50. Note that there are only 1839 \( J \)'s in the training set at this cutoff, which is only about 3% of the whole \( J \) ensemble.

Fig. S4. Accuracy (a) and energy difference (b) of small-\( J \)-trained models over ten size windows. Each grid reports a value averaged over all \( J \)'s with size ±39 around size labels in the x-axis. The black line demarcates whether training set \( J \)'s are over or under the training set size cutoffs shown in the y-axis.