1. **Project Title:** Training Transformer Network to Find Ground State of Ising Models

2. **Investigator List:** Levy Lin, Holden Mac Entee, Yanglet Xiao-Yang Liu

3. **Project Description**

   (a) **Motivation**
   The Ising Model has been a prominent problem in physics since 1920. The model represents a mathematical explanation of ferromagnetism in statistical mechanics. The ground state, or alignment of spins in an Ising model to minimize energy, has many applications such as determining physical properties; for instance, magnetism or phase transitions. A related model, known as **spin glass**, is a magnetic state characterized by randomness. More specifically, the Edwards-Anderson model of spin glass organizes spins on a $d$-dimensional lattice considering only nearest neighbor interactions. The Hamiltonian for this spin system is given by: $H = -\sum_{<ij>} J_{ij}S_iS_j$, as the external magnetic field is assumed 0. This simplified Hamiltonian is related to graph MaxCut, defined as: $H(\sigma) = -\sum_{ij \in E(G)} J_{ij}\sigma_i\sigma_j$. MaxCut, and therefore the ground state of a desired Ising model, can be found using combinatorial optimization.

   (b) **Key Methodology**
   Finding the ground state of Ising models is an NP-Hard problem. We seek to employ a transformer network aided by reinforcement learning that will find the ground state of an Ising model in a quicker time than current state-of-the-art implementations, and improve the scalability of the problem.

   i. **Task 1: Read 3 $\sim$ 5 references and find the associated datasets of Ising models.** Currently we have a dataset from Stanford that can be used to benchmark algorithms. We aim to find several more references containing datasets representing Ising models that we will test current implementations on and compare our solution to.

   ii. **Task 2: Construct Datasets based on Graph Distributions.** We will consider graph distributions such as Barabási-Albert (BA) [11], Erdős-Rényi (ER) [8], Power Law (PL) [9], and Gset to serve as the benchmark data to compare algorithms. The graph distributions are described as follows:

   A. BA: Random scale-free networks. Graphs contain nodes with unusally high degrees compared to other nodes in Graph.

   B. ER: All graphs on a fixed vertex set with a fixed number of edges are equally likely. The probability of generating each graph with $n$ nodes and $M$ edges is: $p^M(1-p)^{\binom{n}{2}-M}$

   C. PL: Any graph whose degree distribution follows a power law. For example, the number of vertices in graph $G$ with degree $i$, $y_i$, is proportional to $i^{-\beta}$ : $y_i \propto i^{-\beta}$. even: evenly, randomly distributed entries tor: 2D torus. All nodes have degree 4. Wrapped in both directions skew: randomly distributed entries, but the avg degree of nodes 1:n appears to be a decaying function, from high (node 1) to low (node n)

   D. Gset: Contains graphs of the following properties:
   1: Even: evenly randomly distributed entries
   2: Tor: All nodes have degree 4, wrapped in both directions
   3: Skew: Randomly distributed entries, but the avg degree of nodes 1:$n$ appears to be a decaying function, from high (node 1) to low (node $n$)

   iii. **Task 3: Reproduce Professor Liu's codes and pipeline for baseline.** We will replicate competitive Ising model solvers such as ECO-DQN [6], S2V-DQN [7], Gurobi, and MCPG [10] to create a benchmark to compare Professor Liu's codes to. These algorithms can be quickly summarized as the following:

   A. ECO-DQN:

   B. S2V-DQN:

   C. Gurobi:

   D. MCPG:

   iv. **Task 4: Implement a variant of transformer network and use RL algorithm to train it under Professor Liu's guidance.** We will implement Professor Liu's transformer network variant and leverage reinforcement learning to generate results on the same datasets used with the aforementioned algorithms to compare the ability of Professor Liu's algorithm.

   (c) **Expected Outcomes**

   i. Beat existing solvers by 0.5% on graphs ranging from 300 to 500 nodes.

   ii. Scale our algorithm to large graph instances ranging from 1,000 to 3,000 nodes and outperform existing solvers by 2-3%.

4. **Estimated Project Timeline**

   (a) 12/1/2024 ∼ 12/31/2024: Read papers; identify datasets to benchmark; set up solver and pipeline

   (b) 1/1/2025 ∼ 2/28/2025: Outperform existing solvers on medium and large datasets

   (c) 3/1/2025 ∼ 4/30/2025: Write paper to submit to NeurIPS 2025

## References

[1] Fan, C., Shen, M., Nussinov, Z. et al. Searching for spin glass ground states through deep reinforcement learning. Nat Commun 14, 725 (2023). https://doi.org/10.1038/s41467-023-36363-w

[2] Kwon, H.Y., Yoon, H.G., Park, S.M. et al. Searching for the ground state of complex spin-ice systems using deep learning techniques. Sci Rep 12, 15026 (2022). https://doi.org/10.1038/s41598-022-19312-3

[3] Jiang, Z., Chen, G., Qiao, R. et al. Point convolutional neural network algorithm for Ising model ground state research based on spring vibration. Sci Rep 14, 2643 (2024). https://doi.org/10.1038/s41598-023-49559-3

[4] Laydevant, J., Marković, D. & Grollier, J. Training an Ising machine with equilibrium propagation. Nat Commun 15, 3671 (2024). https://doi.org/10.1038/s41467-024-46879-4

[5] Mills, K., Ronagh, P. & Tamblyn, I. Finding the ground state of spin Hamiltonians with reinforcement learning. Nat Mach Intell 2, 509–517 (2020). https://doi.org/10.1038/s42256-020-0226-x

[6] Barrett, T. D., Clements, W. R., Foerster, J. N., & Lvovsky, A. I. (2020). Exploratory combinatorial optimization with reinforcement learning. arXiv. https://arxiv.org/abs/1909.04063

[7] Dai, H., Khalil, E. B., Zhang, Y., Dilkina, B., & Song, L. (2018). Learning combinatorial optimization algorithms over graphs. arXiv. https://arxiv.org/abs/1704.01665

[8] Alarfaj, B., Taylor, C., & Bogachev, L. (2023). The joint node degree distribution in the Erdős-Rényi network. arXiv. https://arxiv.org/abs/2303.05138

[9] Tandon, R., & Ravikumar, P. (2013). On the difficulty of learning power law graphical models. In 2013 IEEE International Symposium on Information Theory (pp. 2493–2497). https://doi.org/10.1109/ISIT.2013.6620675

[10] Chen, C., Chen, R., Li, T., Ao, R., & Wen, Z. (2023). Monte Carlo policy gradient method for binary optimization. arXiv. https://arxiv.org/abs/2307.00783

[11] Albert, R., & Barabási, A.-L. (2002). Statistical mechanics of complex networks. Reviews of Modern Physics, 74(1), 47–97. https://doi.org/10.1103/RevModPhys.74.47