Applying Deep Reinforcement Learning to the HP Model for Protein Structure Prediction*

Kaiyuan Yang
Department of Computer Science
National University of Singapore
117417, Singapore

Houjing Huang
Institute of Automation
Chinese Academy of Sciences
Beijing, 100190, China

Olafs Vandans
EXN SIA
Jurmala, Latvia

Adithya Murali
NVIDIA Seattle Robotics Lab
Redmond, 98052, WA, USA

Fujia Tian
Department of Physics
City University of Hong Kong
83 Tat Chee Avenue, HK, China

Roland H.C. Yap
Department of Computer Science
National University of Singapore
117417, Singapore
ryap@comp.nus.edu.sg

Liang Dai
Department of Physics
City University of Hong Kong
83 Tat Chee Avenue, HK, China
liangdai@cityu.edu.hk

Abstract

A central problem in computational biophysics is protein structure prediction, i.e., finding the optimal folding of a given amino acid sequence. This problem has been studied in a classical abstract model, the HP model, where the protein is modeled as a sequence of H (hydrophobic) and P (polar) amino acids on a lattice. The objective is to find conformations maximizing H-H contacts. It is known that even in this reduced setting, the problem is intractable (NP-hard). In this work, we apply deep reinforcement learning (DRL) to the two-dimensional HP model. We can obtain the best known conformations for benchmark HP sequences with lengths from 20 to 50. Our DRL is based on a deep Q-network (DQN). We find that a DQN based on long short-term memory (LSTM) architecture greatly enhances the RL learning ability and significantly improves the search process. DRL can sample the state space efficiently, without the need of manual heuristics. Experimentally we show that it can find multiple distinct best-known solutions per trial. This study demonstrates the effectiveness of deep reinforcement learning in the HP model for protein folding.

1 Introduction

Predicting protein structure from a sequence of amino acids is one of the central problems in computational biophysics research [1, 2]. From the viewpoint of statistical physics, proteins usually fold into the optimal structures with minimum free energies [3]. Following similar approaches to address other complicated problems, physicists have developed an abstract model, the HP model, to simplify the protein structure prediction problem [4]. In the HP model, a protein is represented as a chain of monomers on a 2D or 3D lattice. Each monomer can be either H, standing for hydrophobic, or P, standing for polar. The task is to find the optimal structure for a given sequence of H and P, such as HPPHPH in Fig. 1. The optimal structure is defined as the structure with the maximum number

*The full manuscript is in revision with journal Physica A: Statistical Mechanics and its Applications.

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

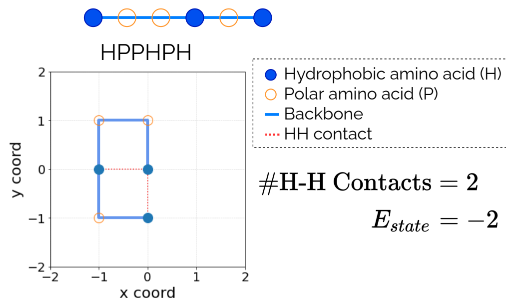

Figure 1: HP model on a 2D square lattice. Example HP sequence, HPPHPH, folds into a conformation with two H-H contacts. The energy of this conformation is optimal, $E_{\text{state}} = -2$.

of H-H contacts. The physical reasoning behind the HP model is that protein structural stability is contributed by the attraction between hydrophobic residues to a large extent [5–7].

Even though the HP model is already a simplified model for protein folding, finding the optimal structure in the HP model is NP-hard [8, 9, 1]. We highlight that the HP model represents the *ab initio* paradigm heavily rooted in biophysics, which is different from the AlphaFold [10, 11] and the academia counterpart 'RoseTTaFold' [12] methods that rely heavily on protein structure data. The value of *ab initio* approaches is that they may help to better understand the problem. In this study, we are firstly motivated to apply reinforcement learning (RL) to the HP model and evaluate its effectiveness. Secondly, in recent years, a couple of studies have attempted various RL approaches, and thus we are also motivated to compare our performance with related work. We investigate what "ingredients" are best needed for a simple "recipe" to solve the HP model using a DRL setup. Here, we will start from a basic prototypical version of the original DQN [13, 14], then we gradually modify/add/subtract components as needed, and finally construct a DRL setup as shown in Fig. 2. Our DRL setup for the HP model can achieve best-known conformations for test benchmark sequences with lengths up to 50, outperforming previous RL approaches on the HP model.

2 Methodology

RL for SAW: The HP model folding process is set up as a self-avoiding walk (SAW). Each successive HP unit is placed onto the lattice site following the square-grid constraint and the self-avoiding constraint. The SAW uses a relative direction scheme consisting of $\mathcal{A}_3 = \{L, F, R\}$. A *contact* is formed when two amino acid units are adjacent on the lattice grid sites, but not adjacent in the HP

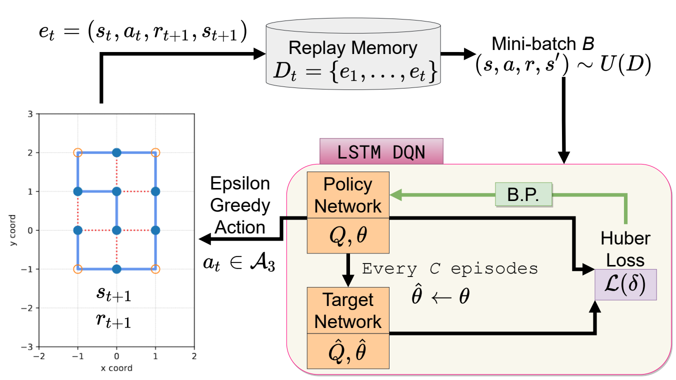

Figure 2: Overview of our DRL method with LSTM-based DQN. Enclosed in the purple border are two NNs for the learning, the Policy Network and the Target Network. The TD error is used to calculate the loss. Back propagation (B.P. in green) then tunes the learnable parameters θ in the Policy Network. The HP model SAW agent interacts with the RL environment at each time step t by taking actions based on the ϵ-greedy method. Action-state-reward experiences e_t are stored in replay memory D_t. Mini-batches B of experiences are uniformly sampled for DQN training.

sequence string, i.e.,not linked via the peptide backbone. Contacts between two hydrophobic H units are called *H-H contacts*. The free energy of a conformation or state, denoted as E_{state}, is the negative value of the number of H-H contacts. Fig. 1 shows a conformation of an HP sequence of length $N = 6$ with $E_{\text{state}} = -2$. In this study, we view solving the HP model as optimal SAW path finding to optimize the HP model score or number of H-H contacts. For each HP sequence, the 'folding' is a SAW path as the HP backbone and its comprising units are embedded in the lattice grid in a non-overlapping fashion. Note the sequential decision making nature of the SAW — the optimal HP model folding is a sequence of walk-steps that traces out the optimal SAW path. Thus, the problem can be framed and potentially solved by RL/DRL.

DQN Setup: Fig. 2 gives an overview of our DRL method using DQN. For RL based experiments, we adopt an ϵ-greedy approach, where ϵ is the probability of the agent selecting an action at random and it decays following an exponential schedule. The ϵ-greedy approach in RL is 'sanity-checked' by comparing with a baseline strategy that randomly explore the state space throughout, called 'RAND'. In 'RAND', ϵ is set to be a constant, $\epsilon = \epsilon_{max} = 100\%$ for all episodes. Our RL environment has a sparse-reward setting. The agent receives reward only at the end of a finished SAW episode, i.e., after the terminal time step T for an episodic walk. The discount factor is set to be 0.98. For an HP sequence, with each state-action transition to a new state s_{t+1}, the RL environment computes the number of H-H contacts or $|E_{\text{state}}|$ of s_{t+1} . The input to the neural networks (NN) in DQN is a one-hot encoded vector [15] representing the state, which captures both the actions performed so far as well as the sequence information of the whole chain as illustrated in Fig. 3. These one-hot encoded arrays are then fed sequentially into the RNN of the DQN. Through empirical testing, we found a two-layer 256-hidden-state stacked LSTM architecture is sufficient for HP sequences shorter than $N \leq 36$. For longer HP sequences ($N = 48, 50$), a three-layer LSTM with 512 hidden states can achieve better results. We store the last $\min(50000, \psi/10)$ number of time steps of transitions in the replay memory buffer, where ψ is the total number of episodes per trial. Our mini-batch size used in DRL training is 32. The optimizer used for learning is Adam [16], with a learning rate of 0.0005. NNs are implemented with the PyTorch framework version 1.10.1, and run on CUDA version 11.3. We use the open source Python library OpenAI Gym to develop the RL environment [17]. Our HP model on a 2D square lattice RL environment is extended and refactored based on the open source repository 'gym-lattice' [18]. All experiments are repeated four times with four random seeds to ensure reproducibility.

Benchmark HP Sequences: A popular benchmark sequence set (Istrail) with documented best-known lowest energy has been referred to by numerous publications in the field [19, 20], and is included in all the related work. For the DRL experiments, N-mers, where $N \in \{20, 24, 25, 36, 48, 50\}$, are selected from the Istrail benchmark for comparison with related work. Table 1 shows the best-known energies of selected lengths and their exact sequences.

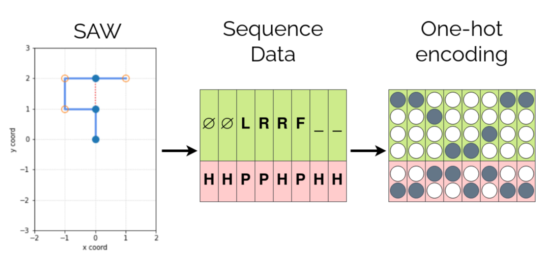

Figure 3: The SAW on the left is not yet complete as there are still two more HP units to be placed onto the 2D square lattice. The current SAW state is converted into a sequence vector of length N representing movement status and monomer type. The first two HP units are fixed, so the action associated is '$\varnothing$'. The remaining two HP units are not yet placed on the lattice and are indicated by place-holder '_' to mean actions to be determined. The action sequence for the current SAW state is L, R, R, F. Finally we encode the sequence data with one-hot encoding for 6 features to preserve their categorical properties.

Table 1: Selected benchmark HP sequences from the Istrail Benchmark with their best known energies.

N-mer ID	Length	HP Sequence	Best Known Energy
20mer-A	20	HPHPPHHPHPPHPHHPPHPH	-9
20mer-B	20	HHHPPHPHPHPPHPHPHPPH	-10
24mer	24	HHPPHPPHPPHPPHPPHPPHPPHH	-9
25mer	25	PPHPPHHPPPPHHPPPPHHPPPPHH	-8
36mer	36	PPPHHPPHHPPPPPHHHHHHHPPHHPPPPHHPPHPP	-14
48mer	48	PPHPPHHPPHHPPPPPHHHHHHHHHHPPPPPPHHPPHHPPHPPHHHHH	-23
50mer	50	HHPHPHPHPHHHHPHPPPHPPPHPPPPHPPPHPPPHPHHHHPHPHPHPHH	-21

3 Results & Discussions

Search Process with DRL: For each of the selected benchmark HP sequences from Table 1, we first establish the baseline performance with pure random explorations, called 'RAND' experiments. For the 'RAND' experiments, the HP sequence N-mer is allowed to randomly explore, pick valid actions at random during each time step, and grow the SAW. In contrast to 'RAND', in our DRL experiments, the N-mer chooses actions based on the ϵ-greedy algorithm and balances exploration with exploitation. Fig. 4 shows the search process in terms of learning curves of RAND and DRL. For all selected benchmark sequences, DRL displays an effective search process, as shown in the downward curve in red in Fig. 4, and consistently finds the best-known solutions in the exploitation phase in all of the four random seed trials. Table 2 shows the performance of the search process on the benchmark HP sequences from Table 1.

Performance Comparison with Related RL Work: We compare our DRL performance with related work's reported results in Table 2. Our DRL design follows the original DQN structure from DeepMind paper [14], and we show that the prototypical DQN algorithm is sufficient to be applied on optimal SAW path finding tasks. But different from previous work's attempt using DQN (and advanced variants of DQN), we design a suitable state representation and NN architecture to

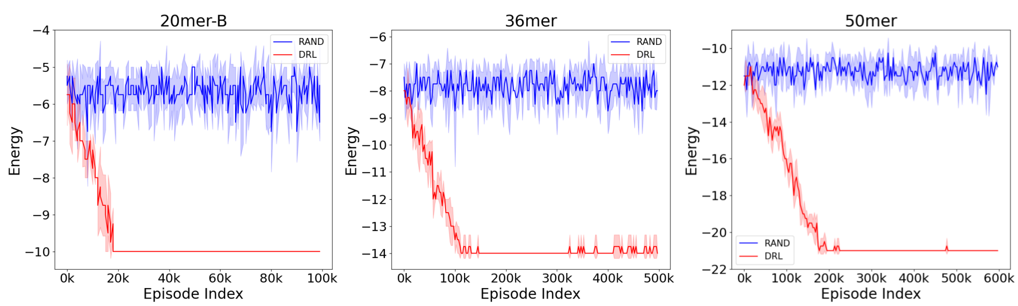

Figure 4: Learning curves of DRL (red) and RAND (blue) on the 100K-episode-trial of 20mer-B, 500K-episode-trial of 36mer, and 600K-episode-trial of 50mer. X-axis is the index of episodes in progression. Y-axis denotes the energy found by the agent. The curves are plotted with the moving minimum of 200 episodes. The shaded area represents the standard deviation. Note the best-known minimal energies for 20mer-B, 36mer, and 50mer are -10, -14, -21 respectively. Experiments are repeated four times on four random seeds for both RAND and DRL methods.

Table 2: Performance comparison among related RL work on Istrail Benchmark sequences. Table entries are lowest energies of E_{state} obtained. '-' indicates information not provided. Results matching the best known energies are highlighted in red bold. Second-closest results are highlighted in blue.

N-mer ID	Dogan-AntQ (2015) [21]	Li-FoldingZero (2018) [22]	Wu-QL (2019) [23]	Yu-DRL (2020) [24][a]	Random	**Ours**	Best Known
20mer-A	-	$\mathbf{-9}$	$\mathbf{-9}$	$-6 \mid -8$	$\mathbf{-9}$	$\mathbf{-9}$	-9
20mer-B	-	-	$\mathbf{-10}$	$-8 \mid -9$	-9	$\mathbf{-10}$	-10
24mer	$\mathbf{-9}$	-8	-	$-6 \mid -8$	$\mathbf{-9}$	$\mathbf{-9}$	-9
25mer	-	-7	-	$- \mid -7$	-7	$\mathbf{-8}$	-8
36mer	-13	-13	-	$- \mid -13$	-12	$\mathbf{-14}$	-14
48mer	-19	-18	-	-	-17	$\mathbf{-23}$	-23
50mer	-	-18	-	-	-15	$\mathbf{-21}$	-21

[a] Yu et al.[24] reported two classes of RL methods, DRL and AlphaGo Zero with Pretraining. Here the two results are separated by '|'.

better utilize the inherent capability of DQN. Our LSTM-based DQN is able to reach best-known solutions for all selected benchmark sequences, and achieves better performance on the HP model than previous results in the literature.

Search Effectiveness: For all four trials, the DRL agent is able to find many distinct best-known solutions. This indicates that our DRL method does not get stuck in a local optimum, but actively traverses the search space for better solutions. We present some of the best-known solutions found by our method for the 48mer in Fig. 5.

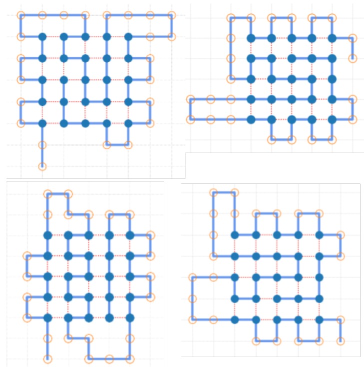

Figure 5: Examples of four best-known solutions found by DRL agent for 48mer with $E_{\text{state}} = -23$. Note these solutions are invariant to translational and rotational symmetries.

4 Conclusion

We demonstrate the effectiveness of applying DRL to the HP model for protein folding. Our "recipe" for DRL setup achieves best-known conformations of benchmark HP sequences ranging from length 20 to 50, which is better than previous results in the literature. We show that the prototypical DQN algorithm is sufficient to be applied to the HP model protein folding but a suitable state representation and LSTM architecture are needed. The LSTM architecture endows DRL with enhanced learning capacity, as LSTM's sequential representation ability captures long range interactions, which are key to protein folding. We present our considerations and design choices in this paper as a possible prototype for future RL application to HP model research, which should be extensible to incorporate more recent DRL developments.

Data Availability Statement

Source code available as a public open-source repository at GitHub Repo URL: `https://github.com/CompSoftMatterBiophysics-CityU-HK/Applying-DRL-to-HP-Model-for-Protein-Structure-Prediction`.

Conformation database showing the distinct best-known and next best conformations is available as a Zenodo open data repository via a link in our GitHub repository.

Broader Impact

The HP model is a classical and one of the most extensively studied physical model for protein structure prediction from sequences. While the HP model appears to be very simple, solving it is proven to be NP-hard. The value of this work lies in two aspects. First, the famous HP model is solved using an emerging approach of DRL, and good results are obtained. While there have been a few recent attempts of applying RL on the HP model, we have obtained better results than previous results in the literature. Second, this work expands the application areas of the machine learning (ML). It is often unclear whether ML can achieve good performance in a new problem without implementation. More importantly, to maximize the ML performance, many components need to be optimized, which is done in this work.

The HP model is also an *ab initio* paradigm to model and understand protein folding, which is an alternative to the recent popular trend of data-driven and learning approaches. Thus our study can help bring new or revisit classical perspectives into the current protein folding research discussions.

No confidential or private data were used in this study. And this work is likely not going to present any foreseeable short-term negative societal consequence.

Acknowledgments

We thank Ken Sung, Wong Limsoon, Tay Yong Chiang from the NUS Department of Computer Science for helpful comments and discussions. This research is financially supported by the National Natural Science Foundation of China (Project No. 21973080), the Research Grants Council of Hong Kong (Project No. 21302520), and City University of Hong Kong (Project No. 7005601).

References

[1] Sorin Istrail, Fumei Lam, et al. Combinatorial algorithms for protein folding in lattice models: a survey of mathematical results. *Communications in Information & Systems*, 9(4):303–346, 2009.

[2] WE Hart and Alantha Newman. Protein structure prediction with lattice models. *Handbook of Molecular Biology*, pages 1–24, 2006.

[3] Christian B Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.

[4] Ken A Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24(6):1501–1509, 1985.

[5] Ken A Dill, Sarina Bromberg, Kaizhi Yue, Hue Sun Chan, Klaus M Ftebig, David P Yee, and Paul D Thomas. Principles of protein folding—a perspective from simple exact models. *Protein science*, 4(4): 561–602, 1995.

[6] Rolf Backofen, Sebastian Will, and Peter Clote. Algorithmic approach to quantifying the hydrophobic force contribution in protein folding. In *Biocomputing 2000*, pages 95–106. World Scientific, 1999.

[7] Chao Tang. Simple models of the protein folding problem. *Physica A: Statistical Mechanics and its Applications*, 288(1):31–48, 2000. ISSN 0378-4371. doi: https://doi.org/10.1016/S0378-4371(00)00413-1. URL `https://www.sciencedirect.com/science/article/pii/S0378437100004131`. Dynamics Days Asia-Pacific: First International Conference on NonLinear Science.

[8] Pierluigi Crescenzi, Deborah Goldman, Christos Papadimitriou, Antonio Piccolboni, and Mihalis Yannakakis. On the complexity of protein folding. *Journal of computational biology*, 5(3):423–465, 1998.

[9] Oswin Aichholzer, David Bremner, Erik D Demaine, Henk Meijer, Vera Sacristán, and Michael Soss. Long proteins with unique optimal foldings in the hp model. *Computational geometry*, 25(1-2):139–159, 2003.

[10] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Bates, et al. Applying and improving alphafold at casp14. *Proteins: Structure, Function, and Bioinformatics*, 2021.

[11] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Bates, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596 (7873):583–589, 2021.

[12] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

[13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *Neural Information Processing Systems (NeurIPS): Deep Learning Workshop*, 2013.

[14] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[15] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[17] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[18] Lester James V. Miranda. gym-lattice: an hp 2d lattice environment with a gym-like api for the protein folding problem, April 2018. URL `https://doi.org/10.5281/zenodo.1214967`.

[19] Mao Chen and Wen-Qi Huang. A branch and bound algorithm for the protein folding problem in the hp lattice model. *Genomics, proteomics & bioinformatics*, 3(4):225–230, 2005.

[20] Chris Thachuk, Alena Shmygelska, and Holger H Hoos. A replica exchange monte carlo algorithm for protein folding in the hp model. *BMC bioinformatics*, 8(1):1–20, 2007.

[21] Berat Doğan and Tamer Ölmez. A novel state space representation for the solution of 2d-hp protein folding problem using reinforcement learning methods. *Applied Soft Computing*, 26:213–223, 2015.

[22] Yanjun Li, Hengtong Kang, Ketian Ye, Shuyu Yin, and Xiaolin Li. Foldingzero: Protein folding from scratch in hydrophobic-polar model. *Neural Information Processing Systems (NeurIPS): Deep Reinforcement Learning Workshop*, 2018.

[23] Hongjie Wu, Ru Yang, Qiming Fu, Jianping Chen, Weizhong Lu, and Haiou Li. Research on predicting 2d-hp protein folding using reinforcement learning with full state space. *BMC bioinformatics*, 20(25): 1–11, 2019.

[24] Haoyang Yu, John S Schreck, and Wann-Jiun Ma. Deep reinforcement learning for protein folding in the hydrophobic-polar model with pull moves. *Neural Information Processing Systems (NeurIPS): Workshop on Machine Learning and the Physical Sciences*, 2020.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [No]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Briefly in Broader Impact section
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] But mentioned the main software versions

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] License in their Github repositories
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]