RL Boltzmann Generators for Conformer Generation in Data-Sparse Environments

Yash Patel
Department of Statistics
University of Michigan
Ann Arbor, MI 48104
yppatel@umich.edu

Ambuj Tewari
Department of Statistics
University of Michigan
Ann Arbor, MI 48104
tewaria@umich.edu

Abstract

The generation of conformers has been a long-standing interest to structural chemists and biologists alike. A subset of proteins known as intrinsically disordered proteins (IDPs) fail to exhibit a fixed structure and, therefore, must also be studied in this light of conformer generation. Unlike in the small molecule setting, ground truth data are sparse in the IDP setting, undermining many existing conformer generation methods that rely on such data for training. Boltzmann generators, trained solely on the energy function, serve as an alternative but display a mode collapse that similarly preclude their direct application to IDPs. We investigate the potential of training an RL Boltzmann generator against a closely related "Gibbs score," and demonstrate that conformer coverage does not track well with such training. This suggests that the inadequacy of solely training against the energy is independent of the modeling modality.

1 Introduction

Protein structure prediction has been a long-standing interest for physical chemists and structural biologists alike, with implications for drug discovery and environmental biopolymers. With recent advancements from AlphaFold2, previously insurmountable challenges have emerged into the realm of possibility [1, 2, 3], highlighted by the introduction of new categories in the latest CASP competition, regarded as the world-stage competition in structural biology [4]. One such category is the modeling of "protein conformational ensembles." AlphaFold2 has achieved unparalleled accuracy for proteins that take on fixed structures; however, many proteins of the human proteome take on multiple shapes, or "conformations." Such "intrinsically disordered proteins" (IDPs) are not captured by current state-of-the-art protein folding solutions [5, 6, 7].

Recent work has investigated the use of machine learning for such conformer generation, albeit for relatively small molecules, largely inspired by techniques from generative modeling. Towards that end, works in molecular conformation generation have explored the use of variational autoencoders [8, 9], normalizing flows [10], and diffusion models [11, 12, 13]. Other works have explored more intricate, physically-inspired modeling [14]. Despite performing well on small molecules, specifically on the GEOM dataset [15], these models all suffer from the drawback that they *require* substantial training data, which ultimately come from MD simulations or experiments. That is, to train their respective models, these approaches all required feeding in draws from the "true" conformer ensembles for a subset of the dataset. This, however, makes such approaches fundamentally unfit for the IDP setting, where MD simulations are completely intractable and traditional experimental methods, such as X-ray crystallography, NMR, and SAXS, fail [16]. Cryo-EM has recently emerged as a potentially viable option, with recent work investigating techniques inspired by neural radiance fields to reconstruct proteins from single-particle cryo-EM images [17, 18, 19, 20]. Such reconstruction, however,

currently fails to handle the large-scale structural dynamism characteristic of IDP conformers. Therefore, neither computational nor experimental methods exist for generating substantial datasets of IDP conformers.

A potential work-around that has been explored in parallel is to train directly against the unnormalized Boltzmann distribution, explored by a subset of generative models known as "Boltzmann generators" [21, 22, 13]. However, training solely against the energy function has been observed to result in concentrated sampling on stable conformers, making the isolated use of the energy function sample inefficient and therefore unable to scale to IDPs [21]. Such an empirical finding, however, has only been replicated in the context of Boltzmann generators, all of which employ normalizing flows. It is, therefore, difficult to disentangle whether such mode collapse is a function of the modeling technique or is more fundamentally linked to the isolated training against the energy.

Another approach to avoid the direct use of the ground truth ensemble is to use reinforcement learning, as was done for simple molecules in [23]. This approach revolves around training an agent (herein referred to as an "RL Boltzmann generator") with a bespoke "Gibbs score" reward function, which closely parallels training directly against the energy function. This initial work, however, failed to investigate the interplay between training the RL Boltzmann generator against the Gibbs score and its final conformer coverage. Our main contribution, therefore, is to demonstrate that RL Boltzmann generators solely training on the Gibbs score also exhibit symptoms of mode collapse, suggesting that this issue is more fundamentally linked to the isolated use of the energy function than to the modeling modality. Code and accompanying documentation for the project is openly available at `https://github.com/yashpatel5400/clean_idp_rl`.

2   Statistical Framework

We first summarize how this problem is posited as a reinforcement learning problem, as was originally formulated in [23]. The ability to learn the true conformer ensemble without training data is feasible because we know the probability distribution of said ensemble follows the Boltzmann distribution,

$$p(x) = \frac{1}{Z} e^{-U(x)/kT} \tag{1}$$

where Z is a normalizing constant, k the Boltzmann constant, T the temperature, and $U : \mathbb{R}^n \to \mathbb{R}$ the energy of the "state" of the system. For the following discussion, we fix a molecule of interest. States $\mathcal{S}$ and actions $\mathcal{A}$ are characterized by the dihedral angles of the molecule backbone. That is, $s, a \in [0, 2\pi]^N$ where N is the number of dihedral angles in the molecule. Actions are further discretized to be multiples of $2\pi/M$ with $M = 6$ to improve training performance.

Dynamics through the environment evolve through sampling of the action space from a policy $a_t \sim \pi(s_{t-1})$. A temporary state $\widehat{s_t}$ is produced by directly acting upon state s_{t-1} with the torsion angles a_t. This $\widehat{s_t}$ is then relaxed using a force field $\mathcal{F}$, specifically MMFF [24], to produce the next state s_t. This procedure is then repeated T times, ultimately producing a sequence of conformers $s := \{s_0, s_1, s_2, ..., s_T\}$. After training the RL Boltzmann generator, we expect this set of conformers s to adequately cover the conformer landscape. Note that the T horizon will vary from molecule to molecule depending on the expected number of conformers.

To reward the agent, a "Gibbs Score" is initially formulated as follows, with $U(x)$ computed using the same classical force field used for relaxation, namely MMFF:

$$\text{Gibbs}(s_t) = \frac{1}{Z_0} \exp\{-(U(s_t) - U_0)/k\tau\}, \tag{2}$$

where Z_0 and U_0 are normalization factors that are empirically estimated at the beginning of simulations. To encourage conformer diversity, however, the reward given is not simply the Gibbs score for the current state, but rather it after undergoing a pruning operation. If the current conformer is similar to one that has been previously observed, we do not wish to "double reward" the agent, else this would directly result in mode collapse. To combat this, we prune similar states before computing the reward. Denoting the current "history" of configurations the molecule has been in as $s := \{s_i\}_{i=1}^{t-1}$, the current conformer is pruned if

$$\exists \, k \text{ s.t. } D_{\text{TFD}}(s_t, s_k) \leq \epsilon, \tag{3}$$

for some pruning threshold ϵ. Here, we use the Torsion Fingerprint Distance (TFD) [25] to compute the similarity of two conformers, although another distance metric such as RMSD may equivalently be employed. The final reward, therefore, is

$$r(s_t) = \text{Gibbs}(s_t) \mathbb{1}[\nexists \, k \text{ s.t. } D_{\text{TFD}}(s_t, s_k) \leq \epsilon]. \tag{4}$$

3 Training Procedure

To solve this learning RL problem, we employ the techniques described in [23], which we highlight here. The overall policy π consists of a Graph Neural Network, which learns a set of embeddings for the molecule, followed by pooling and fully connected layers to map such an embedding to a probability vector over the action space $\mathcal{A}$. π is then trained iteratively using PPO [26, 27].

In greater detail, the employed GNN is an edge-network MPNN [28], which learns a set of embeddings through iterative updating using

$$x_i^{t+1} = \Theta x_i^t + \sum_{j \in \mathcal{N}(i)} h(x_j^t, e_{i,j}), \tag{5}$$

where $\{x_i\}_{i=1}^N$ are the embeddings for the N atoms in the molecule, $e_{i,j}$ is the edge information between atoms (i, j), Θ is a GRU, and h is an MLP. We initialize the atom embeddings as encodings of their locations and types.

After each iteration t, the output embeddings are aggregated using a set-to-set pooling operation [29], which gives us an aggregated embedding of the overall molecule y^t. The above series of iterative transformations therefore means we end up with a history of such molecular embeddings $\{y^i\}_{i=1}^t$ at time step t. Since the goal is for this conformer generator to explore the landscape, taking into account previously visited conformations is critical in assessing future actions. Therefore, these $\{y^i\}$ are further passed through an LSTM to give a current "aggregated" state g^t.

To finally produce the action a_t, we hold fixed the list of torsion tuples that are defined for the molecule of interest, namely tuples $T_i = (b_1^i, b_2^i, b_3^i, b_4^i)$, representing the four atoms involved in the torsion. For each torsion, we pass the g^t and the corresponding embeddings for the atoms through a final network, namely $p_i^t := f(x_{b_1^i}, x_{b_2^i}, x_{b_3^i}, x_{b_4^i}, g^t)$. This produces a probability vector over the action space of a *single* torsion. Sampling $a_i^t \sim p_i^t$ gives the action for the *single* torsion T_i. Repeating this for each of the torsions gives the complete action a_t.

To learn the dynamics, [23] found that curricula had to be employed for complicated molecules [30, 31], since the reward landscape encountered was too sparse. We restrict ourselves to simple molecules here and, therefore, avoid the need for such curriculum learning, but further work may wish to investigate the robustness of the mode collapse found here in light of such curriculum learning.

4 Results

We present results of the performance of RL agent on the traditional small-molecule GEOM-Drugs dataset, following the conventions set forth in [14, 32]. Experiments with GEOM-Drugs were conducted on a randomly sampled subset of 100 molecules, using the following evaluation metric:

$$\text{COV-R}(S_g, S_r) = \frac{1}{|S_g|} \left| \left\{ \mathcal{C} \in S_r \mid \text{RMSD}(\mathcal{C}, \widehat{\mathcal{C}}) \leq \delta, \widehat{\mathcal{C}} \in S_g \right\} \right| \tag{6}$$

where R represents the "recall" metric and S_r, S_g represent the reference and generated conformer ensembles. δ is an arbitrary threshold used for designated two conformers as being "equivalent" for the purposes of assessing discovery. We follow the precedent of [13] and report a graph of metrics over δ. A similar metric can be defined for P, the precision, simply by exchanging S_g and S_r.

We compare our results against the current state of the art methods, namely Torsional Diffusion, GeoMol, and FF-optimized runs of RDKit's ETKDG [33], but largely wish to focus our discussion on the mode collapse phenomenon observed in training the RL model. The Torsional Diffusion and GeoMol models were assessed using the pre-trained models provided in the corresponding repositories, confirmed to not to have been trained on any of the molecules chosen in the test set used here. Implementation of the above described GNN models used for RL is in PyTorch-Geometric [28, 34], with the RL environment interface provided by OpenAI's gym [35] and implementation of PPO by stable-baselines [36]. The model was trained using Adam [37] on an NVIDIA Tesla V100 GPU per molecule and completes in roughly two hours for 100,000 time steps.

4.1 GEOM-Drugs

We present the mean observed recalls and precisions of the RL agent on GEOM-Drugs in Figures 1 and 2 respectively. The RL agent performance, for both metrics, is plotted at three stages of the training: at step 0, step 50,000, and step 100,000, respectively referred to as "untrained," "midtrained,", and "trained." We, thus, wish to concentrate this discussion on trends visible comparing these stages.
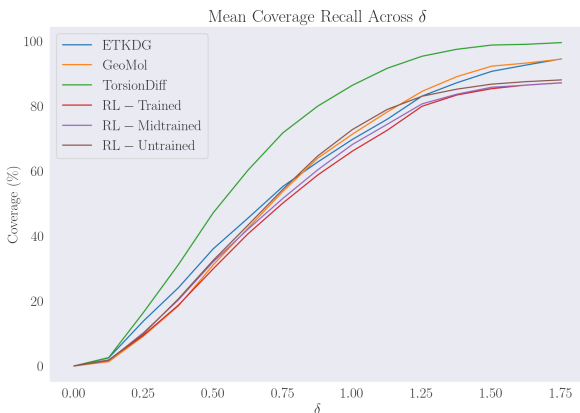

Figure 1: Assessment of the mean coverage recalls over coverage thresholds δ for the untrained, midtrained, and trained RL agent against other state-of-the-art methods.
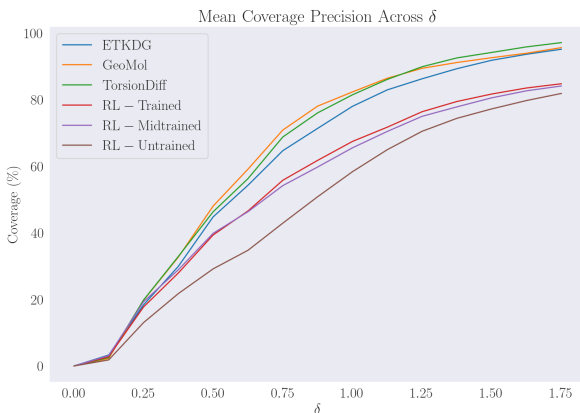

Figure 2: Assessment of the mean coverage precisions over coverage thresholds δ for the untrained, midtrained, and trained RL agent against other state-of-the-art methods.

Of particular note is that the recall of the RL agent *decreases* over the training run in exchange for its precision *increasing*, respectively seen in the rightward shift of the recall curves and upward shifts of the precision curves for progressively more trained RL agents. This implies the RL agent conformer generation becomes more highly concentrated on a smaller subset of the conformer space, symptomatic of a mode collapse. Such a finding is particularly noteworthy given that the Gibbs score was explicitly constructed to encourage conformer diversity through pruning. An immediate question of interest, therefore, is to investigate the source of misalignment between the posited Gibbs score and coverage and whether such misalignment can be rectified.

5   Discussion

We have, therefore, seen that conformer generation purely using the known energy function of the Boltzmann distribution even in the form of the Gibbs score seems insufficient to obtain a diverse set of conformers. This suggests a number of directions for further investigation beyond those discussed above. A unifying theme for such direct extensions is circumventing the seeming necessity to have training data to achieve comprehensive conformer coverage while being unable to directly produce it. A current line of investigation we are pursuing is motivated by the curriculum learning employed in the previous RL Boltzmann generator work: given that MD simulations can be tractably run on small proteins or subsets of large proteins, such data can be used either to train a traditional Boltzmann generator or as a curriculum in an RL Boltzmann generator. That is, for a protein of interest, progressively longer subsequences of its amino acid chain could be taken, some of which could be feasibly analyzed with MD. For such subsequences, the Boltzmann generator could be trained with *both* the energy and the generated MD ground truths, with the intention being that subsequences where no such truth is available would begin from an informed "prior" for energy-based sampling, resulting in enhanced conformer coverage. Pursuing this with traditional Boltzmann generators would involve a non-trivial extension to allow for such generators to be shared across different molecules, since current generators must be trained for each molecule of interest separately.

In this vein, future work should pursue characterizing the sample efficiency of the RL generator compared to traditional Boltzmann generators, since it will point to which approach is more likely to scale to the full IDP conformer problem. That is, comparing the improvement in coverage of RL vs. traditional Boltzmann generators with each additionally provided ground truth conformer would be of great interest. Further, as briefly mentioned, the IDP problem is actively being pursued in the Cryo-EM community. With concurrent efforts in the experimental and simulation spaces, a synergy between the two is highly desirable. Integrating the method found to have the optimal sample efficiency in an iterative active learning loop with a Cryo-EM data collection pipeline would further improve the reconstruction quality and efficiency.

6   Impact Statement

As stated, this work continues in the vein of using machine learning for structural biology and, more generally, biological understanding, where given the increasing complex nature of modern scientific inquiry, it seems likely that such coordinated efforts between human ingenuity and machine learning will play an ever-increasing role in scientific exploration going forward. We expect the methods proposed here to be directly relevant for a number of biological problems, specifically in the vein of ultimately being able to study cell dynamics in silico. In the nearer term, this work will assist in understanding IDP dynamics and ultimately how these relate to their signaling functions in ligand interactions. Such work on using RL for dynamics will likely also aid in protein interaction simulations, where interpretable dynamics are of great interest. An orthogonal yet related field in which such structural dynamics are of interest is material science, in which the same methodology can be directly applied. No clear ethical concerns are posed by this work.

References

[1] 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.

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

[3] Patrick Cramer. Alphafold2 and the future of structural biology. *Nature Structural & Molecular Biology*, 28(9):704–705, 2021.

[4] Marco Necci, Damiano Piovesan, and Silvio CE Tosatto. Critical assessment of protein intrinsic disorder prediction. *Nature methods*, 18(5):472–481, 2021.

[5] Paul CD Hawkins. Conformation generation: the state of the art. *Journal of chemical information and modeling*, 57(8):1747–1756, 2017.

[6] Christopher J Oldfield and A Keith Dunker. Intrinsically disordered proteins and intrinsically disordered protein regions. *Annu Rev Biochem*, 83(1):553–584, 2014.

[7] A Keith Dunker, J David Lawson, Celeste J Brown, Ryan M Williams, Pedro Romero, Jeong S Oh, Christopher J Oldfield, Andrew M Campen, Catherine M Ratliff, Kerry W Hipps, et al. Intrinsically disordered protein. *Journal of molecular graphics and modelling*, 19(1):26–59, 2001.

[8] Aayush Gupta, Souvik Dey, Alan Hicks, and Huan-Xiang Zhou. Artificial intelligence guided conformational mining of intrinsically disordered proteins. *Communications biology*, 5(1):1–11, 2022.

[9] Minkai Xu, Wujie Wang, Shitong Luo, Chence Shi, Yoshua Bengio, Rafael Gómez-Bombarelli, and Jian Tang. An end-to-end framework for molecular conformation generation via bilevel programming. *CoRR*, abs/2105.07246, 2021.

[10] Elman Mansimov, Omar Mahmood, Seokho Kang, and Kyunghyun Cho. Molecular geometry prediction using a deep generative graph neural network. *CoRR*, abs/1904.00314, 2019.

[11] Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pages 8867–8887. PMLR, 2022.

[12] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

[13] Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and Tommi Jaakkola. Torsional diffusion for molecular conformer generation. *arXiv preprint arXiv:2206.01729*, 2022.

[14] Octavian Ganea, Lagnajit Pattanaik, Connor Coley, Regina Barzilay, Klavs Jensen, William Green, and Tommi Jaakkola. Geomol: Torsional geometric generation of molecular 3d conformer ensembles. *Advances in Neural Information Processing Systems*, 34:13757–13769, 2021.

[15] Simon Axelrod and Rafael Gomez-Bombarelli. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):1–14, 2022.

[16] Tamas Lazar, Elizabeth Martínez-Pérez, Federica Quaglia, András Hatos, Lucía B Chemes, Javier A Iserte, Nicolás A Méndez, Nicolás A Garrone, Tadeo E Saldaño, Julia Marchetti, et al. Ped in 2021: a major update of the protein ensemble database for intrinsically disordered proteins. *Nucleic acids research*, 49(D1):D404–D411, 2021.

[17] Julian Giraldo-Barreto, Sebastian Ortiz, Erik H Thiede, Karen Palacio-Rodriguez, Bob Carpenter, Alex H Barnett, and Pilar Cossio. A bayesian approach to extracting free-energy profiles from cryo-electron microscopy experiments. *Scientific Reports*, 11(1):1–15, 2021.

[18] Pilar Cossio and Gerhard Hummer. Likelihood-based structural analysis of electron microscopy images. *Current opinion in structural biology*, 49:162–168, 2018.

[19] Ali Punjani and David J Fleet. 3d variability analysis: Resolving continuous flexibility and discrete heterogeneity from single particle cryo-em. *Journal of structural biology*, 213(2):107702, 2021.

[20] Ellen D Zhong, Tristan Bepler, Bonnie Berger, and Joseph H Davis. Cryodrgn: reconstruction of heterogeneous cryo-em structures using neural networks. *Nature methods*, 18(2):176–185, 2021.

[21] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

[22] Jonas Köhler, Andreas Krämer, and Frank Noé. Smooth normalizing flows. *Advances in Neural Information Processing Systems*, 34:2796–2809, 2021.

[23] Tarun Gogineni, Ziping Xu, Exequiel Punzalan, Runxuan Jiang, Joshua Kammeraad, Ambuj Tewari, and Paul Zimmerman. Torsionnet: A reinforcement learning approach to sequential conformer search. *Advances in Neural Information Processing Systems*, 33:20142–20153, 2020.

[24] Thomas A Halgren and Robert B Nachbar. Merck molecular force field. iv. conformational energies and geometries for mmff94. *Journal of computational chemistry*, 17(5-6):587–615, 1996.

[25] Tanja Schulz-Gasch, Christin Scharfer, Wolfgang Guba, and Matthias Rarey. Tfd: torsion fingerprints as a new measure to compare small molecule conformations. *Journal of chemical information and modeling*, 52(6):1499–1512, 2012.

[26] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[27] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[28] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[29] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[30] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[31] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[32] Chence Shi, Shitong Luo, Minkai Xu, and Jian Tang. Learning gradient fields for molecular conformation generation. In *International Conference on Machine Learning*, pages 9558–9568. PMLR, 2021.

[33] Sereina Riniker and Gregory A Landrum. Better informed distance geometry: using what we know to improve conformation generation. *Journal of chemical information and modeling*, 55(12):2562–2574, 2015.

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[35] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[36] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (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)? [No]
   (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)? [Yes]

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? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

    (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]