Symmetry Discovery with Deep Learning

Krish Desai
Department of Physics
University of California
Berkeley, CA 94720
krish.desai@berkeley.edu

Benjamin Nachman
Physics Division
Lawrence Berkeley National Laboratory
Berkeley, CA 94720
bpnachman@lbl.gov

Jesse Thaler
Center for Theoretical Physics
Massachusetts Institute of Technology
Cambridge, MA 02139
jthaler@mit.edu

Abstract

Symmetries are a fundamental property of functions associated with data. A key function for any dataset is its probability density, and the symmetries thereof are referred to as the symmetries of the dataset itself. We provide a rigorous statistical notion of symmetry for a dataset, which involves reference datasets that we call "inertial" in analogy to inertial frames in classical mechanics. Then, we construct a novel approach to automatically discover symmetries from a dataset using a deep learning method based on an adversarial neural network. We test our method on the LHC Olympics dataset. Symmetry discovery may lead to new insights and can reduce the effective dimensionality of a dataset to increase its effective statistics.

1 Introduction

The equations of motion for physical systems are closely tied to their symmetries. Often these symmetries are known from fundamental principles, but there are also systems with unknown or emergent symmetries. Discovering and characterizing these symmetries is an essential component of research in physics. Beyond their inherent interest, symmetries are also practically useful for increasing the statistical power of a dataset for a variety of analysis goals. For example, a dataset can be augmented with pseudodata generated by applying symmetry operations to the existing data in order to create a larger training sample for various machine learning tasks. Neural network architectures can be constructed that respect the symmetries (e.g. Convolutional Neural Networks and translation symmetries [1]) in order to improve generalization and reduce the number of model parameters. Futhermore, symmetries can significantly increase the size of a useful synthetic dataset created from a generative model trained on a limited set of examples [2].

Deep learning is a powerful tool for identifying patterns in high-dimensional data and is therefore a promising tool for identifying symmetries. A variety of deep learning methods have been proposed for symmetry discovery and related problems. Neural networks can parametrize the equations of motion for physical systems, which have various conserved quantities resulting from symmetries [3, 4]. Generic neural networks targeting classification tasks can encode symmetries in their hidden layers [5, 6]. This possibility can be used to actively learn symmetries by encoding a shared equivariance in hidden layers across learning tasks [7]. Directly learning symmetries can be framed as an inference problem given access to parameteric symmetric translations of the same dataset [8]. Another class of targeted approaches can be found in the domain of automatic data augmentation. If a dataset can be

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).

augmented without changing its statistical properties, then one has learned a symmetry. Significant advances in this area have used reinforcement learning [9, 10].

An alternative symmetry discovery approach that is flexible, fully differentiable, and simple uses generative models [11, 12]. A typical generative model is a function that maps random numbers to structured data. A deep generative surrogate model is trained so that the resulting probability density matches that of a target dataset. For symmetry discovery, the random numbers are replaced with the target dataset itself and the well-trained generator will implement the symmetry. In this paper, we extend the generative model framework for symmetry discovery and introduce it to the physics community. In particular, we build a rigorous statistical framework for describing the symmetries of a dataset and construct a learning paradigm for automatically detecting generic symmetries. Our deep learning setup is simpler than existing approaches and we develop an analytic understanding of the algorithm performance in simple cases. This allows us to understand some of the dynamics of the machine learning.

2 Statistics of Symmetries

Let X be a random variable on $\mathbb{R}^n$ governed by probability density function (PDF) p, and let x be an instantiation of X. Naively, a *symmetry* is a function $g : \mathbb{R}^n \to \mathbb{R}^n$ which preserves the PDF: $p(X = x) = p(X = g(x))|g'(x)|$, where $|g'(x)|$ is the Jacobian determinant. While this is a necessary condition, it is not sufficient. It is easiest to see this when $n = 1$. Let F be the cumulative distribution function (CDF) of X. Then, $F(X) \sim \mathcal{U}(0, 1)$, where $\mathcal{U}(0, 1)$ is the uniform random variable on the interval $[0, 1]$. Conversely, $F^{-1}(\mathcal{U}(0, 1))$ is a random variable governed by the PDF p (for technical details and proofs, see [13]). The uniform distribution on the interval $[0, 1]$ has many PDF-preserving functions such as the percentile inversion map: $\widetilde{g}(x) = 1 - x$. This map has the additional property that $\widetilde{g}^2(x) = x$, so it appears to represent a $\mathbb{Z}/2\mathbb{Z}$ symmetry. However, the CDF map from above means that every probability density p admits a $\mathbb{Z}/2\mathbb{Z}$ PDF-preserving map: $g = F^{-1} \circ \widetilde{g} \circ F$. Clearly, not all one-dimensional random variables have a $\mathbb{Z}/2\mathbb{Z}$ symmetry and so the definition of symmetry must be stricter than simply PDF-preserving.

While there is a map as above for every one-dimensional random variable, each such variable requires a different map. This suggests a sharper definition of symmetry that makes use of a reference distribution. Given two PDFs $p : \mathbb{R}^n \to \mathbb{R}$ and $p_I : \mathbb{R}^n \to \mathbb{R}$, a map $g : \mathbb{R}^n \to \mathbb{R}^n$ is defined to be a symmetry of p relative to p_I if it is PDF-preserving for both p and p_I. The reference or "inertial" density p_I is the analogue of an inertial reference frame in classical mechanics. This new definition of symmetry will typically exclude quantile maps, like $\widetilde{g}$ above, because the $\widetilde{g}$ that works for one random variable will typically not work for another (e.g Gaussian and exponential random variables). While this new definition solves the problem of "fake" symmetries, it also introduces a dependence on the inertial distribution. However, there is often a canonical choice for p_I which reduces the number of possibilities in practice. In particular, a natural choice for many physics datasets is to pick the uniform distribution on $\mathbb{R}^n$, where n is the dimension of the dataset.

3 Machine Learning Approach

Our general strategy for discovering symmetries in a dataset is to simultaneously learn two functions. The generator, $g : \mathbb{R}^n \to \mathbb{R}^n$ represents the symmetry map and the classifier, $f : \mathbb{R}^n \to [0, 1]$ tries to distinguish the input data $\{x_i\}$ from the transformed data $\{g(x_i)\}$. When the classifier cannot distinguish the original data from the transformed data, then g is a symmetry. The technical details of this statement are provided in Sec. 4. As described in Sec. 2, it is insufficient to require that g preserve the PDF of the input data. There are several methods to implement the inertial distribution condition into the learning strategy introduced above. One possibility is that the classifier f could be applied to both the input dataset and data drawn from the inertial probability density. The training procedure could penalize any map g which does not fool f for both datasets. Alternatively, one could find all PDF preserving maps g and then *post hoc* select the ones that also preserve samples from the inertial dataset. If the PDF preserving maps of p_I are known, then one could restrict the set of maps g and then perform an unconstrained optimization on the restricted search space.

Each of these methods has advantages and disadvantages. A key challenge with the first two options is that they require sampling from the inertial density. When p_I is uniform on $\mathbb{R}^n$, this is not feasible.

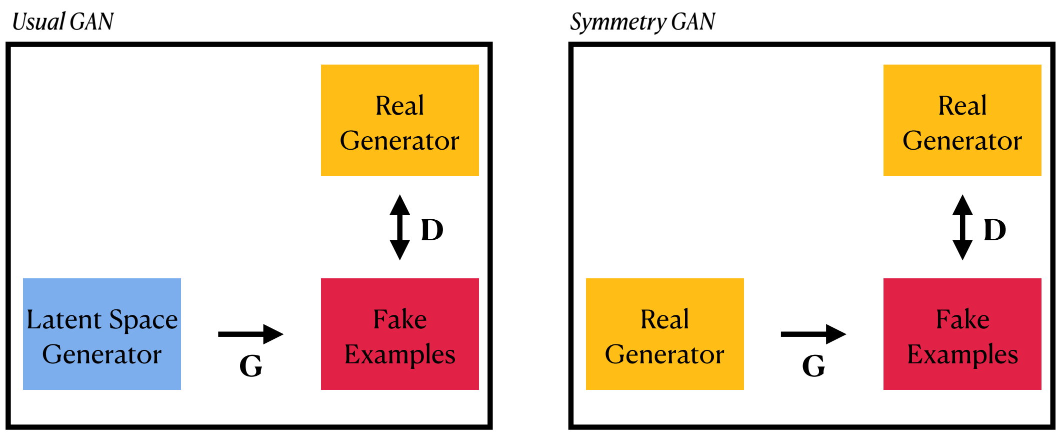


Figure 1: A diagram of the usual training setup for a Generative Adversarial Network (GAN) (left) and the variation proposed in this paper for automatically discovering symmetries (right).

Approximate methods are possible, such as cutting off the support of p_I a few standard deviations away from the mean of p. The second method is computationally wasteful, as the space of PDF preserving maps is generally much larger than the space of symmetry maps. This paper will focus on the third option: restricting the set of functions g to be PDF preserving for p_I.

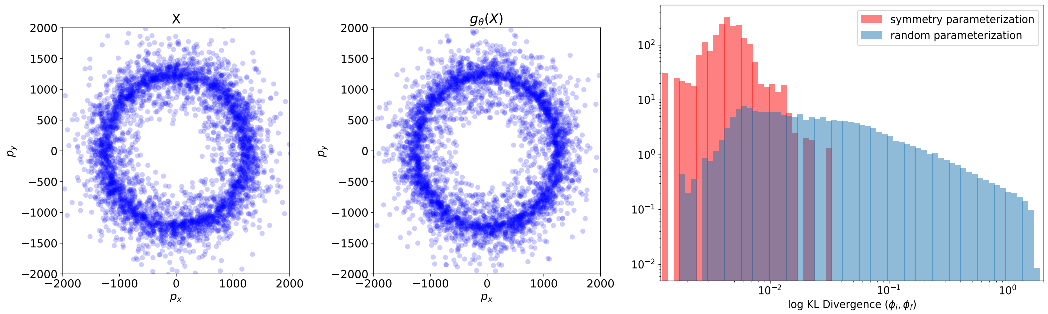
For any open set $\mathcal{O} \subseteq \mathbb{R}^n$, a differentiable function $g : \mathcal{O} \to \mathcal{O}$ preserves the PDF of the uniform distribution $\mathcal{U}(\mathcal{O})$[1] if and only if g is an equiareal map. The Jacobian determinant of a differentiable equiareal map is $|g'| = 1$ and the probability density function is $p(x) = \frac{1}{\mathrm{Vol}(\mathcal{O})}$. Hence the PDF preserving condition $p = p \circ g \cdot |g'|$ is met if and only if $|g'| = 1$, i.e. g is equiareal. Ergo, our search space to discover symmetries of physics datasets should be the space equiareal maps of appropriate dimension. The set of equiareal maps for $n > 1$ is not well characterized. For example, even for $n = 2$ not all equiareal maps are linear. A simple example of a non-linear area-preserving map is the Hénon map [14]: $f(x, y) = (x, y - x^2)$. This makes the space of equiareal maps difficult to directly encode in the learning. While the general set of equiareal maps is difficult to parameterize, the set of area preserving linear maps on $\mathbb{R}^n$, a subgroup of the general affine group $\mathrm{Aff}_n(\mathbb{R})$, are well understood and characterized as topological groups of dimension $n(n+1) - 1$. These maps even have complete parameterizations such as the *Iwasawa decomposition* [15] which significantly aid the symmetry discovery process. We leave the discovery of non-linear symmetries to future work.

4 Deep Learning Implementation

The procedure described in Sec. 3 is implemented by modifying the learning setup of a Generative Adversarial Network (GAN) [16]. In a typical GAN setup, a generator function g surjects a latent space onto a data space[2] and an auxiliary classifier distinguishes generated examples from target examples. For symmetry discovery, the latent probability density is the same as the target probability density. This setup is illustrated in Fig. 1. The GAN generator g and discriminator f are paramterized as neural networks and trained simultaneously to optimize the binary cross entropy loss functional: $L[g, f] = -\sum_{x \in \{x_i\}} [\log(f(x)) + \log(1 - f(g(x)))]$, which differs from the usual binary cross entropy in that the same samples appear in the first and second terms. This is similar to neural resampler in Ref. [19] or step 2 of the OmniFold algorithm [20].

[1] This may be an improper prior, but by carefully taking suitable limits the ideas go through, and the important takeaway is that uniform distributions are preserved by equiareal maps.

[2] While all the GANs discussed here are bijective, GANs in general need not be. Symmetry discovery requires the generator to be bijective. One may wish to use nomalizing flows [17, 18] instead in future work.



(i) (ii)

Figure 2: (i) The LHC Olympics data, as is (left) and transformed by the generator discovered by the neural net (right). (ii) The Kullback-Leibler divergence between ϕ_i and the randomly rotated and symmetry rotated ϕ_fs on a $\log-\log$ scale.

5 Particle Physics Example

We tested the above algorithm on one and two dimensional Gaussian data, both numerically and analytically, and the results are very promising. As an application to particle physics, we use background dijet events from the LHC Olympics dataset [21, 22]. These events are simulated using `Pythia 8` [23, 24] and `Delphes 3.4.1` [25–27]. The reconstructed particles of each event are clustered into $R = 1$ anti-k_T [28] jets using `FastJet` [29, 30]; all events are required to satisfy a single $p_T > 1.2$ TeV jet trigger.

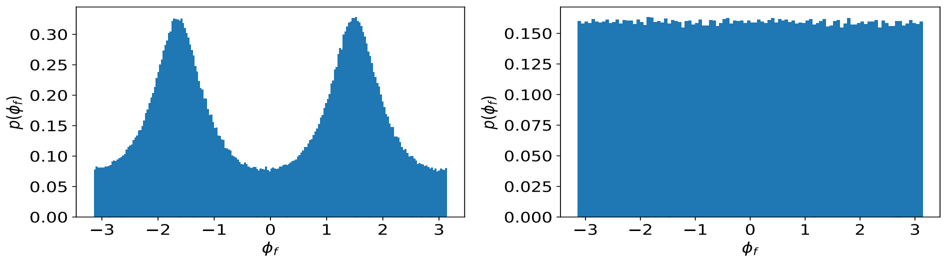
We represent each event as a four-dimensional vector $(p_{1x}, p_{1y}, p_{2x}, p_{2y})$, where p_1 refers to the momentum of the leading jet, p_2 represents the momentum of the subleading jet and x and y are the Cartesian coordinates in the transverse plane. We focus on the transverse plane because the jets are typically back-to-back in this plane as a result of momentum conservation. The longitudinal momentum of the parton-parton interaction is not known and so there is no corresponding conservation law for p_z. A natural search space would be $SO(4)$, the group of all rotations on $\mathbb{R}^4$. $SO(4)$ is a six parameter group, specified by $\{\theta_i\}_{i=1}^6$, which parametrize the six independent rotations, which rotate as $R_1 : p_{1x} \to p_{1y}, R_2 : p_{1x} \to p_{2x}, R_3 : p_{1x} \to p_{2y}, R_4 : p_{1y} \to p_{2x}, R_5 : p_{1y} \to p_{2y}, R_6 : p_{2x} \to p_{2y}$, where the notation $R : a \to b$ means $R(a) = a\cos\theta + b\sin\theta, R(b) = b\cos\theta - a\sin\theta$. The generator $g_{\boldsymbol{\theta}}$ is $g_{\boldsymbol{\theta}}(X) = R_1R_2R_3R_4R_5R_6\ X$.

It is difficult to graph six dimensional space, and the symmetries discovered do not lie in any single plane of $SO(4)$. Ergo, one must seek alternative methods to verify that the maps discovered by the neural net are indeed symmetries. One can do so in two ways. First, one plots the data, X and $g_{\boldsymbol{\theta}}(X)$ and compares their features. Fig. 2(i) demonstrates the similarity between X and $g_{\boldsymbol{\theta}}(X)$ suggesting that $g_{\boldsymbol{\theta}}$ is a symmetry.

For the second method, one observes that due to momentum conservation, the LHC Olympics data can be well-described by a single angle ϕ_i uniformly distributed over $[0, 2\pi)$. If one applies an arbitrary rotation, there is no reason the new internal angles $\phi_f = \arctan 2(g_{\boldsymbol{\theta}}(X)_2, g_{\boldsymbol{\theta}}(X)_1) = \arctan 2(g_{\boldsymbol{\theta}}(X)_4, g_{\boldsymbol{\theta}}(X)_3)$ should be uniformly distributed anymore. However, if one of the symmetry rotations discovered by the neural net is applied to X, ϕ_f must remain uniformly distributed, as shown in Fig. 3. This effect may be quantified by computing the Kullback-Leibler (KL) divergence of the two different kinds of ϕ_f againt ϕ_i. As one can see in Fig. 2(ii), the KL divergence of the symmetries is clearly less than the KL divergence of the random rotations. The KL divergence drops off linearly on the $\log-\log$ plot which implies a polynomial interpolation form for the histogram.

6 Conclusions and Outlook

This paper provides a rigorous statistical definition of the term *symmetry* and proposes a novel, flexible, and fully differentiable deep learning approach to symmetry discovery. The modified



(i) (ii)

Figure 3: Four examples of the internal angle distribution of the LHCO data rotated by a (i) randomly selected rotation, (ii) a symmetry in $SO(4)$.

GAN framework termed *Symmetry GAN* was applied to physical data sets with promising results, conforming with our predictions.

Moving forward, this area is rich with questions for further exploration. One may attempt to explore non-linear symmetries of data by better parametrizing the space of equiareal maps over $\mathbb{R}^n$. The process of learning the symmetry discovery map also offers scope for further work, in particular in the area of increasing the flexibility of the neural net, enabling it to learn the symmetry discovery map in more general situations than is currently possible.

The code for this paper can be found at this Github repository.

References

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989. DOI: 10.1162/neco.1989.1.4.541.

[2] A. Butter, S. Diefenbacher, G. Kasieczka, B. Nachman, and T. Plehn, "GANplifying Event Samples," 2020. arXiv: 2008.06545 [hep-ph].

[3] S. Greydanus, M. Dzamba, and J. Yosinski, *Hamiltonian neural networks*, 2019. arXiv: 1906.01563 [cs.NE].

[4] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho, *Lagrangian neural networks*, 2020. arXiv: 2003.04630 [cs.LG].

[5] G. Barenboim, J. Hirn, and V. Sanz, "Symmetry meets AI," Mar. 2021. arXiv: 2103.06115 [cs.LG].

[6] S. Krippendorf and M. Syvaeri, "Detecting Symmetries with Neural Networks," Mar. 2020. arXiv: 2003.13679 [physics.comp-ph].

[7] A. Zhou, T. Knowles, and C. Finn, *Meta-learning symmetries by reparameterization*, 2021. arXiv: 2007.02933 [cs.LG].

[8] G. Benton, M. Finzi, P. Izmailov, and A. G. Wilson, *Learning invariances in neural networks*, 2020. arXiv: 2010.11882 [cs.LG].

[9] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, *Autoaugment: Learning augmentation policies from data*, 2019. arXiv: 1805.09501 [cs.CV].

[10] S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim, *Fast autoaugment*, 2019. arXiv: 1905.00397 [cs.LG].

[11] R. Hataya, J. Zdenek, K. Yoshizoe, and H. Nakayama, *Faster autoaugment: Learning augmentation strategies using backpropagation*, 2019. arXiv: 1911.06987 [cs.CV].

[12] A. Antoniou, A. Storkey, and H. Edwards, *Data augmentation generative adversarial networks*, 2018. arXiv: 1711.04340 [stat.ML].

[13] J. E. Angus, "The probability integral transform and related results," *SIAM Review*, vol. 36, no. 4, pp. 652–654, 1994, ISSN: 00361445. [Online]. Available: http://www.jstor.org/stable/2132726.

[14] M. Hénon, "Numerical study of quadratic area-preserving mappings," *Quarterly of Applied Mathematics*, vol. 27, no. 3, pp. 291–312, 1969, ISSN: 0033569X, 15524485. [Online]. Available: http://www.jstor.org/stable/43635985.

[15] K. Iwasawa, "On some types of topological groups," *Annals of Mathematics*, vol. 50, no. 3, pp. 507–558, 1949, ISSN: 0003486X. [Online]. Available: http://www.jstor.org/stable/1969548.

[16] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," 2014. arXiv: 1406.2661 [stat.ML].

[17] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," *International Conference on Machine Learning*, vol. 37, p. 1530, 2015.

[18] I. Kobyzev, S. Prince, and M. Brubaker, "Normalizing Flows: An Introduction and Review of Current Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1, 2020, ISSN: 1939-3539. DOI: 10.1109/tpami.2020.2992934. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2020.2992934.

[19] B. Nachman and J. Thaler, "A Neural Resampler for Monte Carlo Reweighting with Preserved Uncertainties," vol. 102, p. 076 004, 2020. DOI: 10.1103/PhysRevD.102.076004. arXiv: 2007.11586 [hep-ph].

[20] A. Andreassen, P. T. Komiske, E. M. Metodiev, B. Nachman, and J. Thaler, "OmniFold: A Method to Simultaneously Unfold All Observables," *Phys. Rev. Lett.*, vol. 124, no. 18, p. 182 001, 2020. DOI: 10.1103/PhysRevLett.124.182001. arXiv: 1911.09107 [hep-ph].

[21] G. Kasieczka, B. Nachman, and D. Shih, *R&D Dataset for LHC Olympics 2020 Anomaly Detection Challenge*, version v4, Zenodo, Apr. 2019. DOI: 10.5281/zenodo.4536377. [Online]. Available: https://doi.org/10.5281/zenodo.4536377.

[22] G. Kasieczka *et al.*, "The LHC Olympics 2020: A Community Challenge for Anomaly Detection in High Energy Physics," Jan. 2021. arXiv: 2101.08320 [hep-ph].

[23] T. Sjöstrand, S. Mrenna, and P. Z. Skands, "PYTHIA 6.4 Physics and Manual," *JHEP*, vol. 05, p. 026, 2006. DOI: 10.1088/1126-6708/2006/05/026. arXiv: hep-ph/0603175 [hep-ph].

[24] ——, "A Brief Introduction to PYTHIA 8.1," *Comput. Phys. Commun.*, vol. 178, p. 852, 2008. DOI: 10.1016/j.cpc.2008.01.036. arXiv: 0710.3820 [hep-ph].

[25] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaitre, A. Mertens, and M. Selvaggi, "DELPHES 3, A modular framework for fast simulation of a generic collider experiment," *JHEP*, vol. 02, p. 057, 2014. DOI: 10.1007/JHEP02(2014)057. arXiv: 1307.6346 [hep-ex].

[26] A. Mertens, "New features in Delphes 3," *J. Phys. Conf. Ser.*, vol. 608, p. 012 045, 2015. DOI: 10.1088/1742-6596/608/1/012045.

[27] M. Selvaggi, "DELPHES 3: A modular framework for fast-simulation of generic collider experiments," *J. Phys. Conf. Ser.*, vol. 523, p. 012 033, 2014. DOI: 10.1088/1742-6596/523/1/012033.

[28] M. Cacciari, G. P. Salam, and G. Soyez, "The anti-k_t jet clustering algorithm," *JHEP*, vol. 04, p. 063, 2008. DOI: 10.1088/1126-6708/2008/04/063. arXiv: 0802.1189 [hep-ph].

[29] ——, "FastJet User Manual," *Eur. Phys. J.*, vol. C72, p. 1896, 2012. DOI: 10.1140/epjc/s10052-012-1896-2. arXiv: 1111.6097 [hep-ph].

[30] M. Cacciari and G. P. Salam, "Dispelling the N^3 myth for the k_t jet-finder," *Phys. Lett.*, vol. B641, p. 57, 2006. DOI: 10.1016/j.physletb.2006.08.037. arXiv: hep-ph/0512210 [hep-ph].

Checklist

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? [N/A]

(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? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
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)? [N/A]
   (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] We did not require significant computing resources for the example shown in this extended abstract.
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]