Identifying Hamiltonian Manifold in Neural Networks

Yeongwoo Song[1], Hawoong Jeong[1 2]*
[1]Department of Physics, KAIST [2]Center for Complex Systems, KAIST
ywsong1025@kaist.ac.kr, hjeong@kaist.edu

Abstract

Recent studies to learn physical laws via deep learning attempt to find the shared representation of the given system by introducing physics priors or inductive biases to the neural network. However, most of these approaches tackle the problem in a system-specific manner, in which one neural network trained to one particular physical system cannot be easily adapted to another system governed by a different physical law. In this work, we use a meta-learning algorithm to identify the general manifold in neural networks that represents Hamilton's equation. We meta-trained the model with the dataset composed of five dynamical systems each governed by different physical laws. We show that with only a few gradient steps, the meta-trained model adapts well to the physical system which was unseen during the meta-training phase. Our results suggest that the meta-trained model can craft the representation of Hamilton's equation in neural networks which is shared across various dynamical systems with each governed by different physical laws.

1 Introduction

Deep learning has succeeded in many application areas such as image classification, language translation, and so on. One of the major roles of such accomplishment was capable of parameterizing useful representations from sufficient data with neural networks. However, grafting deep learning onto physics is yet another problem. They struggle to learn conservation laws or implicit physical geometries or symmetries. Exposing the hidden physical interpretation in neural networks has drawn the attention of many researchers since various tasks solvable with deep learning share several physical priors (e.g. training a robot agent with reinforcement learning needs to take gravity into account).

Although various studies make the model learn the conservative quantities or symmetries inside the system, they still suffer from several drawbacks. Previous works that try to induce physical biases into neural networks are system-specific, which means that if a model is trained for one system, it cannot easily adapt to another system with different physical laws. Furthermore, systems whose physics is unknown have more sparse data. These flaws make the standard supervised learning methods harder to learn the physics of the system.

Here, we focus our objectives on systems that can be formulated by Hamiltonian mechanics. Systems governed by Hamiltonian mechanics inhere symmetries, and their state trajectories are lain on a certain manifold. Moreover, almost all of the physics in our nature have its own conservation laws and Hamiltonian mechanics relates the state of the system to its corresponding conservative quantities (usually energy). In this work, we search for overall representations that share the essence of Hamiltonian mechanics (which is generalized by Hamilton's equations) in the form of neural networks. Thereby, making no further assumptions such as physical priors to the model, we can learn the physics of the system through a data-driven approach and add more physical interpretability to neural networks.

*Correspondence to Hawoong Jeong.

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

2 Preliminaries

2.1 Hamiltonian mechanics

Hamiltonian mechanics is a tool to describe the state of a dynamical system by a vector composed of canonical coordinates $\boldsymbol{x} = (\boldsymbol{q}, \boldsymbol{p})$. Here $\boldsymbol{q}$ and $\boldsymbol{p}$ denotes generalized coordinates and its canonical momentum respectively; $\boldsymbol{q} = (q_1, q_2, \ldots, q_N)$, $\boldsymbol{p} = (p_1, p_2, \ldots, p_N)$, where N is the degree of freedom of the given system. Then a specific scalar, which we call a Hamiltonian ($\mathcal{H}$) is defined such that Hamilton's equation (Equation. 1) is held. In fact, for almost all classical systems, the Hamiltonian of the system corresponds to its energy where it is to be conserved.

$$\frac{d\boldsymbol{q}}{dt} = \frac{\partial \mathcal{H}}{\partial \boldsymbol{p}}, \frac{d\boldsymbol{p}}{dt} = -\frac{\partial \mathcal{H}}{\partial \boldsymbol{q}} \tag{1}$$

From Equation. 1, we can see that the vector field $\boldsymbol{X}_{\mathcal{H}} = \left(\frac{\partial \mathcal{H}}{\partial \boldsymbol{p}}, -\frac{\partial \mathcal{H}}{\partial \boldsymbol{q}}\right)$ gives the time evolution of the system. Here, $\boldsymbol{X}_{\mathcal{H}}$ is called a Hamiltonian vector field which is known as symplectic, and thus allows the Hamiltonian $\mathcal{H}$ to lie on a symplectic manifold. Symplectic manifold is one of the essences of differential geometry, which naturally arouse from the formulations of classical mechanics. It provides the generalization of the phase space of a closed system, thus allowing one to probe the time evolution of a system with the aid of Equation. 1 (Arnol'd [1], Libermann and Marle [2]).

2.2 Learning dynamics from neural networks

There are various works that use neural networks to analyze the time evolution or the conservation law of the physical system. Greydanus et al. [3] proposed Hamiltonian Neural Networks (HNNs) to parameterize the Hamiltonian of the system with a neural network. HNN learns the dynamics of the given system by inducing a physical bias to the loss function. Making use of Hamilton's equation (Equation. 1), HNN predicts the dynamics using the symplectic gradient.

As an alternative way to encode physical bias to the objective function, Saemundsson et al. [4] introduced a new type of network called Variational Integrator Networks (VINs) that preserve the geometrical structure of physical systems. The architecture of VIN is designed to match the discrete-time equations of motion of the given dynamical system. Thus it provides interpretability and more efficient learning by preserving the manifold geometry ingrained in physical systems directly to the model architecture.

Furthermore, recent studies including Yin et al. [5], Kirchmeyer et al. [6] suggest a data-driven approach to learn the dynamics of a system generalized across different environments. These focus on dynamical systems that are governed in the form of the following differential equation;

$$\frac{d\boldsymbol{x}(t)}{dt} = f_e(\boldsymbol{x}(t)) \tag{2}$$

where $\boldsymbol{x}(t)$ is a time-dependent state in a phase space $\mathcal{X}$ and $f_e : \mathcal{X} \to T\mathcal{X}$ maps $\boldsymbol{x}(t)$ to its time-derivative state $\frac{d\boldsymbol{x}(t)}{dt}$ which lies in the tangent space $T\mathcal{X}$. Although Hamilton's equation (Equation. 1) implies the temporal dynamics as similar as in Equation. 2, the dissimilarity in the right-hand side of each equation discerns the difference which is more complex to reveal the characteristics that are generalized across various systems.

2.3 Meta-learning

The goal of meta-learning is to train a model well generalized on various data such that the model trained with a meta-learning algorithm can easily adapt to new tasks from only a small amount of unseen samples. There are various kinds of meta-learning; model-based, metric-based, and optimization-based (Hospedales et al. [7]). Among these strategies, optimization-based methods are readily compatible and applicable with any differentiable model. Thus we adopt a fairly general optimization-based meta-learning algorithm; Model-Agnostic Meta-Learning (MAML) introduced by Finn et al. [8]. More detail about the MAML algorithm is described in Section. 3.2, and 3.3.

Lee et al. [9] proposed that the MAML algorithm can be used for finding the physical law governed by Hamiltonian mechanics. However, it was restricted to systems that share the same functional form of Hamiltonians (e.g. different systems governed by the same physical law but with various experiment settings or physical parameters).

Bringing the proposed question more generally, can we find the representation that is shared between various Hamiltonians? In this work, we search for the portrayal of a symplectic manifold governed by Hamilton's equation in neural networks.

3 Methods

3.1 Preparing the dataset for meta-learning

We prepared a task distribution composed of 6 types of physical systems (mass-spring, pendulum, double-pendulum, two-body, three-body, magnetic-mirror). For all of our system, we generated a dataset with $N = 10,000$ trajectories confined to two-dimensional space, where the canonical coordinates $\boldsymbol{x} = (\boldsymbol{q}, \boldsymbol{p})$ as input, and their time derivatives $\dot{\boldsymbol{x}} = (\dot{\boldsymbol{q}}, \dot{\boldsymbol{p}})$ as output for each trajectory. We provide the system specification in the Appendix. A.1.

In the language of meta-learning, five types of systems will be used for meta-training, and the remaining one for meta-testing (evaluation). We will denote each trajectory as $\mathcal{T}_{\mathbb{S},i}$, which is sampled from the task distribution $p(\mathcal{T}_{\mathbb{S}})$ (where $S \in$ {mass-spring, pendulum, double-pendulum, two-body, three-body, magnetic-mirror}, and $i \in [\![1..N]\!]$). Then tasks from $\mathcal{T}_{\mathbb{S}^\complement}$ will compose the task distribution for meta-training, and tasks from $\mathcal{T}_{\mathbb{S}}$ will form the data distribution for meta-testing. For example, we will use the data distribution $\mathcal{T}_{\mathbb{S}^\complement}$; $\mathbb{S}^\complement = \{$mass-spring, pendulum, double-pendulum, two-body, three-body$\}$ for meta-training, $\mathcal{T}_{\mathbb{S}}$; $S = \{$magnetic-mirror$\}$ for evaluating the magnetic-mirror system.

3.2 Meta-training the neural network

We adopt an approach from Sanchez-Gonzalez et al. [10] for the model architecture, which uses a graph network to parameterize a system with a number of particles. Using graph neural networks to describe a system of several particles mediates the model to handle various degree of freedom inputs.

Upon the framework that we described so far, we represent our model by f_θ parameterized by neural network parameters θ. Here, considering the various input ($\boldsymbol{x}^i$), output ($\dot{\boldsymbol{x}}^i$) scale across different systems, we use the log-cosh loss for the loss function $\mathcal{L}$:

$$\mathcal{L}_{\mathcal{T}_i}(f_\theta) = \sum_{(\boldsymbol{x}^i, \dot{\boldsymbol{x}}^i) \sim \mathcal{T}_i} \log\cosh\big(f_\theta(\boldsymbol{x}^i) - \dot{\boldsymbol{x}}^i\big)$$

In the inner loop of our variation of MAML, the model adapts to a new task $\mathcal{T}_{\mathbb{S}^\complement,i}$ by updating the parameters with one gradient step using $K = 50$ phase space points for each task.

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_{\mathbb{S}^\complement,i}}(f_\theta)$$

In the outer loop using the Adam optimizer (Kingma and Ba [11]), meta-optimization across systems is done by updating the parameters as below.

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_i \mathcal{L}_{\mathcal{T}_{\mathbb{S}^\complement,i}}(f_{\theta_i'})$$

3.3 Evaluating the meta-trained model

We evaluate the performance by making a few-step gradient descent with the same K phase space points from the unseen system and compared the performance between the meta-trained model and the random initialized HNN model. We tested The initial condition for the HNN model such as; the number of layers, the number of hidden units and the type of activation were set as same as the original work. The model adaptation to the novel system is done by updating the parameters as follows.

$$\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_{\mathbb{S},i}}(f_\theta)$$

Using the adapted model at each step, we integrate our model output with the fourth-order Runge-Kutta integrator to achieve the system dynamics (Runge [12]). Then, the performance was achieved by calculating the mean squared error between the predicted and ground truth trajectory points.

4 Results

4.1 Quantitative analysis

For each of the systems, we evaluated the performance of the meta-learned model which was meta-trained with the data composed by the rest of the system. We took the gradient step up to 100 and evaluated the mean squared error between the predicted and the ground truth trajectory points at every adaptation step. To corroborate the stability of the meta-trained model, we repeated the adaptation task 100 times using different trajectories randomly sampled at each time. In the tasks performed on magnetic-mirror, double-pendulum, pendulum, mass-spring system, the meta-trained model showed a lower mean squared error (Figure 1). During the adaptation task, the number of given points was the same as during the meta-training phase.

Greydanus et al. [3] trained the HNN model up to the order of 10^3 gradient steps to provide baseline results. However, we can see that the meta-trained model doesn't need the same gradient steps used in the HNN baseline. Within a gradient step of less than 100, the meta-learned model adapts faster and better than the HNN baseline.

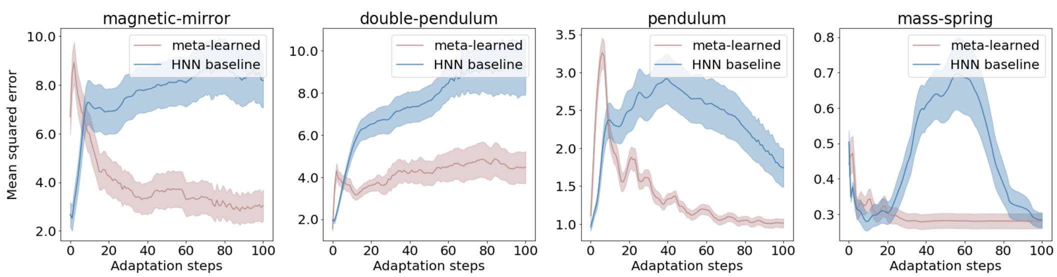

Figure 1: Evaluation result of the meta-trained model for magnetic-mirror, double-pendulum, pendulum, mass-spring system. For each system, we performed the adaptation task with 100 trajectories and calculated the average of the mean squared error between the ground-truth and the predicted trajectories at each gradient step.

4.2 Qualitative analysis

From the previous analysis, we observed that the meta-trained model showed a lower mean squared error between the ground-truth and the predicted dynamics. However, a small trajectory error does not guarantee correct system dynamics prediction. To provide a more concrete result, we randomly selected a trajectory from an unseen system and examined the predicted dynamics performed with the meta-trained model after 100 adaptation steps.

First, for the case of the magnetic-mirror and the double-pendulum system, although there exists a definite difference between the predicted dynamics and the ground truth, the meta-trained model adapted faster than the HNN baseline. Next, we observed that the adaptation to the pendulum system seems to be finished for the meta-learned model, but not for the HNN baseline. Finally, for the mass-spring adaptation task, both the meta-learned model and the HNN baseline adapted well to the ground-truth dynamics.

Furthermore, it is worth noting the behaviour of the prediction of the surplus coordinates (r in the pendulum system, and y in the mass-spring system), since, in the real world, we cannot readily distinguish what coordinates remain physically meaningful among different phase variables. Here, we confined the dynamics of the systems to a two-dimensional space, thus r and y remain constant

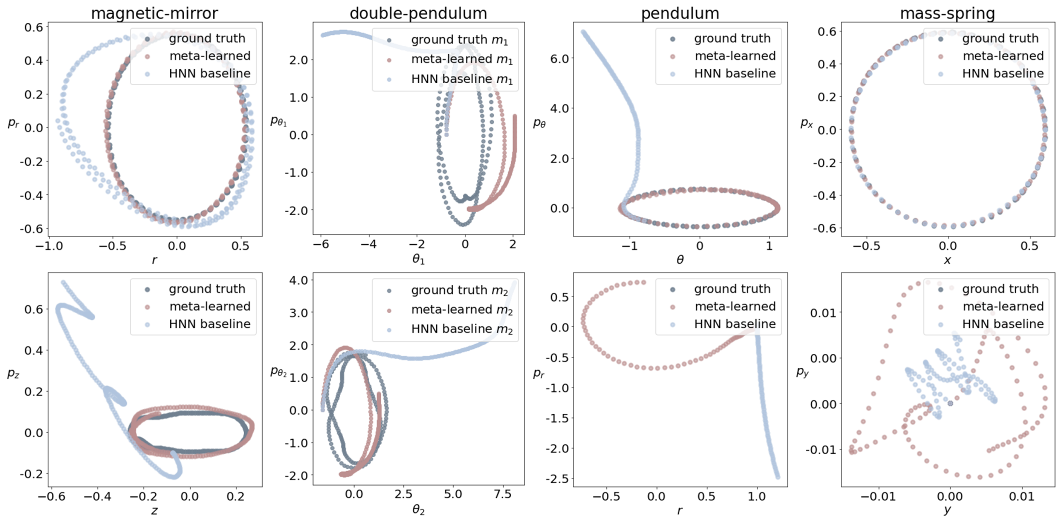

Figure 2: Magnetic-mirror, double-pendulum, pendulum, mass-spring system dynamics prediction with the model after 15 gradient steps for qualitative investigation. For all of the corresponding systems, the meta-trained model tends to adapt faster than the HNN baseline.

for the pendulum and mass-spring system each. Inspecting the predicted trajectories of the redundant coordinates, we can see that the scale of the unnecessary coordinates (r and y) is much smaller than the necessary ones (θ and x).

5 Discussion

While previous studies worked on the problem of finding a generalization of the system dynamics across different environments, we extended the problem to identifying a shared representation of a physical system across different physical laws. We exploit meta-learning algorithm to pursue the representation of Hamilton's equation across distinct physical systems in neural networks. Our comparison result between the meta-learned model and the HNN baseline advocate that the neural network learned the Hamiltonian of an unseen system by identifying the representation for Hamilton's equation itself rather than crafting the Hamiltonian directly.

6 Broader impact

In nature, every physical system has its governing law with an underlying principle above them. For example, systems with different Hamiltonians can be expressed with the same Hamilton's equation. We proposed a method to embed such general physical fundamentals into neural networks using a meta-learning algorithm. This provides experimental evidence of the manifold hypothesis (Fefferman et al. [13]), therefore facilitating a physical interpretation of neural networks. Furthermore, this approach does not entail physical priors or inductive biases to the model and provides a data-driven way to resolve questions in physics using deep learning. However, this works lacks the investigation for the two-body, and the three-body system despite they were used for the meta-training. Including this issue, experiments on a more realistic system should be made in further studies.

Acknowledgments and Disclosure of Funding

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government NRF-2018S1A3A2075175 (YS), and NRF-2022R1A2B5B02001752 (HJ).

References

[1] Vladimir Igorevich Arnol'd. *Mathematical methods of classical mechanics*, volume 60. Springer Science & Business Media, 2013.

[2] Paulette Libermann and Charles-Michel Marle. *Symplectic geometry and analytical mechanics*, volume 35. Springer Science & Business Media, 2012.

[3] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[4] Steindor Saemundsson, Alexander Terenin, Katja Hofmann, and Marc Deisenroth. Variational integrator networks for physically structured embeddings. In *International Conference on Artificial Intelligence and Statistics*, pages 3078–3087. PMLR, 2020.

[5] Yuan Yin, Ibrahim Ayed, Emmanuel de Bézenac, Nicolas Baskiotis, and Patrick Gallinari. Leads: Learning dynamical systems that generalize across environments. *Advances in Neural Information Processing Systems*, 34:7561–7573, 2021.

[6] Matthieu Kirchmeyer, Yuan Yin, Jérémie Donà, Nicolas Baskiotis, Alain Rakotomamonjy, and Patrick Gallinari. Generalizing to new physical systems via context-informed dynamics model. *arXiv preprint arXiv:2202.01889*, 2022.

[7] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey, 2020.

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[9] Seungjun Lee, Haesang Yang, and Woojae Seong. Identifying physical law of hamiltonian systems via meta-learning. In *International Conference on Learning Representations*, 2021.

[10] Alvaro Sanchez-Gonzalez, Victor Bapst, Kyle Cranmer, and Peter Battaglia. Hamiltonian graph networks with ode integrators. *arXiv preprint arXiv:1909.12790*, 2019.

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

[12] Carl Runge. Über die numerische auflösung von differentialgleichungen. *Mathematische Annalen*, 46(2):167–178, 1895.

[13] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

[14] Ch Efthymiopoulos, M Harsoula, and G Contopoulos. Resonant normal form and asymptotic normal form behaviour in magnetic bottle hamiltonians. *Nonlinearity*, 28(4):851, 2015.

A Appendix

A.1 System specification used for meta-learning

Mass-Spring One of the most simplest physical system is one particle, frictionless mass-spring system.

$$\mathcal{H} = \frac{p_x^2}{2m} + \frac{kx^2}{2}$$

We sampled 100 points within [0s, 10s] per trajectory, with the constants as $m = k = 1$ for simplicity.

Pendulum Hamiltonian of a pendulum system is slightly more complex than mass-spring case.

$$\mathcal{H} = \frac{p_\theta^2}{2ml^2} + mgl(1 - \cos\theta)$$

We sampled 100 points within [0s, 10s] per trajectory, with the constants as $m = g = l = 1$ for simplicity.

Double-Pendulum Double pendulum system is a most common example of chaotic motion.

$$\mathcal{H} = \frac{m_2 l_2^2 p_{\theta_1}^2 + (m_1 + m_2) l_1^2 p_{\theta_2}^2 - 2 m_2 l_1 l_2 p_{\theta_1} p_{\theta_2} \cos(\theta_1 - \theta_2)}{2 m_2 l_1^2 l_2^2 [m_1 + m_2 \sin^2(\theta_1 - \theta_2)]} - (m_1 + m_2) g l_1 \cos\theta_1 - m_2 g l_2 \cos\theta_2$$

We sampled 200 points within [0s, 20s] per trajectory, with the constants as $m_1 = m_2 = g = l_1 = l_2 = 1$ for simplicity.

Two-Body In the two-body system case, we now consider the interaction between two particles. Then the Hamiltonian of the two-body system can be written as follows.

$$\mathcal{H} = \frac{\boldsymbol{p}_1^2}{2m_1} + \frac{\boldsymbol{p}_2^2}{2m_2} - \frac{G m_1 m_2}{|\boldsymbol{r}_1 - \boldsymbol{r}_2|}$$

We sampled 200 points within [0s, 20s] per trajectory, with the constants as $m_1 = m_2 = G = 1$ for simplicity.

Three-Body Adding a particle to the two-body system gives the three-body system. Although the Hamiltonian of the three-body system is a incidental extension of the two-body case, chaotic behaviour is observed.

$$\mathcal{H} = \frac{\boldsymbol{p}_1^2}{2m_1} + \frac{\boldsymbol{p}_2^2}{2m_2} + \frac{\boldsymbol{p}_3^2}{2m_3} - \frac{G m_1 m_2}{|\boldsymbol{r}_1 - \boldsymbol{r}_2|} - \frac{G m_2 m_3}{|\boldsymbol{r}_2 - \boldsymbol{r}_3|} - \frac{G m_3 m_1}{|\boldsymbol{r}_3 - \boldsymbol{r}_1|}$$

We sampled 100 points within [0s, 10s] per trajectory, with the constants as $m_1 = m_2 = m_3 = G = 1$ for simplicity.

Magnetic-Mirror Up to here, the given dynamical system was rather simple. Here we introduce a system that has a more complicated form of Hamiltonian. From the works of Efthymiopoulos et al. [14], the Hamiltonian of a magnetic bottle-type system is given below.

$$\mathcal{H} = \frac{1}{2}(\dot{\rho}^2 + \dot{z}^2) + \frac{1}{2}\rho^2 + \frac{1}{2}\rho^2 z^2 - \frac{1}{8}\rho^4 + \frac{1}{8}\rho^2 z^4 - \frac{1}{16}\rho^4 z^2 + \frac{1}{128}\rho^6$$

We sampled 200 points within [0s, 20s] per trajectory.

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? [No]
   (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)? **[TODO]**
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [No]
   (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]
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? [No]

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]

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]