Decay-aware neural network for event classification in collider physics

Tomoe Kishimoto
Computing Research Center, High Energy Accelerator Research Organization
1-1 Oho, Tsukuba, Ibaraki, Japan
International Center for Elementary Particle Physics, The University of Tokyo
Institute for AI and Beyond, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
tomoe.kishimoto@kek.jp

Masahiro Morinaga
International Center for Elementary Particle Physics, The University of Tokyo
Institute for AI and Beyond, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
morinaga@icepp.s.u-tokyo.ac.jp

Masahiko Saito
International Center for Elementary Particle Physics, The University of Tokyo
Institute for AI and Beyond, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
saito@icepp.s.u-tokyo.ac.jp

Junichi Tanaka
International Center for Elementary Particle Physics, The University of Tokyo
Institute for AI and Beyond, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
jtanaka@icepp.s.u-tokyo.ac.jp

Abstract

The goal of event classification in collider physics is to distinguish signal events of interest from background events to the extent possible to search for new phenomena in nature. We propose a decay-aware neural network based on a multi-task learning technique to effectively address this event classification. The proposed model is designed to learn the domain knowledge of particle decays as an auxiliary task, which is a novel approach to improving learning efficiency in the event classification. Our experiments using simulation data confirmed that an inductive bias was successfully introduced by adding the auxiliary task, and significant improvements in the event classification were achieved compared with boosted decision tree and simple multi-layer perceptron models.

1 Introduction

In collider physics experiments, a large number of events[1] are produced from particle collisions using high-energy accelerators, such as the Large Hadron Collider [1]. For data analysis, event

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

[1] The term "event" corresponds to "image" in image classification.

classification becomes crucial as it enables an effective separation of interesting signal events from background events. Machine learning, such as boosted decision trees (BDT), has a long history in collider physics [2, 3]; deep learning (DL) is also widely used to enhance the performance of event classification. DL can provide significant discrimination power by utilizing its huge parameter space; however, a large amount of data is required to maximize its performance. A multi-task learning technique is a feasible approach to improving learning efficiency by introducing an inductive bias. Multiple related tasks are trained simultaneously while using a shared representation in multi-task learning, which may result in better performance than training the tasks individually.

A decay-aware neural network based on the multi-task learning technique is proposed in this paper to efficiently address the event classification problem. In the proposed model, the main event classification task and an auxiliary task are trained simultaneously, where the domain knowledge of particle decays, as shown in Figure 1, is learned as the auxiliary task. In collider physics experiments, heavy particles of interest, such as Higgs boson (h), immediately decay to other particles and they are not observed by the detectors. Therefore, the auxiliary task based on particle decay knowledge is expected to provide a good inductive bias because the features of these heavy particles can be extracted efficiently from the observed particles.

In this study, we perform experiments using simulation data to evaluate the decay-aware neural network. Furthermore, we also present the details of datasets and the proposed model structure. The remainder of this paper is organized as follows. Section 2 describes the related works, including our novelty. Section 3 summarizes the datasets used in this study. Section 4 provides details of the proposed model. Section 5 presents the experimental results. Finally, Section 6 concludes the paper.

2 Related work

A previous study reported that a multi-layer perceptron (MLP) model outperformed traditional BDT by identifying powerful features for the event classification [4]. Graph neural network (GNN) [5] architecture has also been reported to enhance the event classification performance [6, 7]. Although we use this GNN architecture in our proposed model to extract graph-level and edge-level features, in contrast to these previous studies, our study has the following contributions:

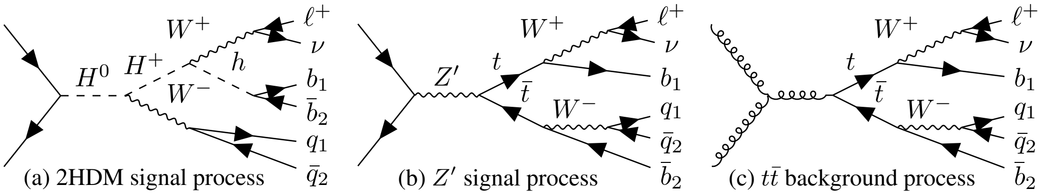
- The multi-task learning is introduced to improve event classification performance. The rich structure of GNN enables solving multiple tasks simultaneously. The event classification and auxiliary tasks are implemented as the graph-level and edge-level classifications, respectively.
- Domain knowledge about particle decay is learned in the auxiliary task. The labels of auxiliary tasks are prepared by counting the decay points between the observed objects.

3 Datasets

The training data were produced using particle physics simulations: proton–proton collision events were generated by MadGraph5_aMC@NLO [8] at a center of mass energy of 13 TeV, with showering and hadronization performed by Pythia8 [9] and detector response simulated by Delphes [10]. In this study, two types of datasets are defined as follows:

- 2HDM dataset: Two-Higgs-doublet model (2HDM) [11, 12], which introduces additional Higgs bosons, H^0, A and $H^{\pm}$, is used as the signal event. Top pair production ($t\bar{t}$) of the Standard Model is used as the background event. Figure 1 (a) and (c) show Feynman diagrams for the signal and background processes. The final state particles are one lepton (ℓ, electron or muon), one neutrino (ν), two b-quarks (b), and two light-quarks (q).
- Z' dataset: A heavy neutral particle (Z') decaying into top quark pair [13, 14] is used as the signal event. The background events are the $t\bar{t}$ process. The decay chains of the top pairs are the same between signal and background events, as shown in Figure 1 (b) and (c).

Neutrinos were reconstructed as a missing transverse momentum object. b-quark and light-quark were reconstructed as b-jet and light-jet objects, respectively. In this study, the object-type is indicated by an integer, and the four-momenta of each object (p_{T}, η, ϕ, mass) and object-type were employed as input variables. The log transformation was applied to p_{T} and mass to fit the values



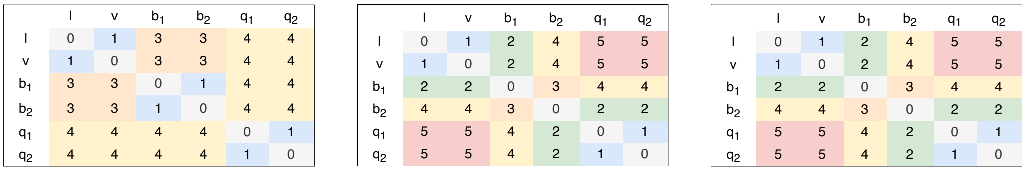
(a) 2HDM signal process (b) Z' signal process (c) $t\bar{t}$ background process

Figure 1: Examples of Feynman diagram for the signal and background processes.

within a reasonable range. 5×10^5, 5×10^4, and 5×10^4 events were independently generated for training, validation, and test phases for each signal and background process.

3.1 Definition of auxiliary task labels

The label of the auxiliary task is defined by counting the particle decay points, including gluon splitting points, between observed objects. Figure 2 presents examples of the label matrices corresponding to Figure 1. For instance, the label between ℓ and ν in the $t\bar{t}$ process is 1 because they originated from the same W decay; moreover, the label between ℓ and b_1 is 2 because there are two decay points: t and W decays. The information of truth particles in Delphes outputs is analyzed event by event to associate the observed objects with the truth particles to determine these labels. Although many diagrams, including the next-to-leading order diagram, can be considered for a given physics process, we specifically assume the tree-level diagrams in Figure 1 to prepare the label matrices in this study. The label matrices are represented by an edge-level class in the GNN.



	l	v	b_1	b_2	q_1	q_2
l	0	1	3	3	4	4
v	1	0	3	3	4	4
b_1	3	3	0	1	4	4
b_2	3	3	1	0	4	4
q_1	4	4	4	4	0	1
q_2	4	4	4	4	1	0

(a) 2HDM signal process

	l	v	b_1	b_2	q_1	q_2
l	0	1	2	4	5	5
v	1	0	2	4	5	5
b_1	2	2	0	3	4	4
b_2	4	4	3	0	2	2
q_1	5	5	4	2	0	1
q_2	5	5	4	2	1	0

(b) Z' signal process

	l	v	b_1	b_2	q_1	q_2
l	0	1	2	4	5	5
v	1	0	2	4	5	5
b_1	2	2	0	3	4	4
b_2	4	4	3	0	2	2
q_1	5	5	4	2	0	1
q_2	5	5	4	2	1	0

(c) $t\bar{t}$ background process

Figure 2: Examples of the label matrices for each process corresponding to Figure 1. There are s- and t-channels in the $t\bar{t}$ process; however, the t-channel uses the same label as the s-channel to fit the same class range with the other processes. The max class label is 5 in this study.

4 Proposed decay-aware neural network

Figure 3 shows an overview of the proposed decay-aware neural network. A graph structure composed of nodes and edges was built for the input data. In this study, each node in the graph corresponds to an object. Thus, object features: four-momentum and object-type are assigned to the node attribute. Edges are prepared as a fully-connected bidirectional graph, including a self-loop.

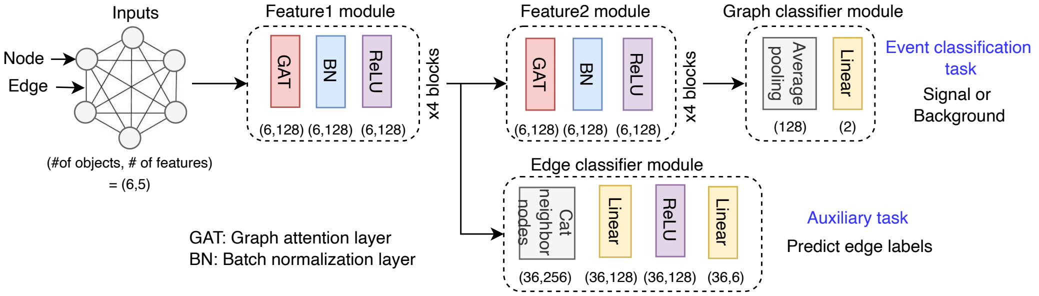


Figure 3: Overview of the proposed DL model. The numbers in the bracket indicate output shapes.

The model consisted of four parts: feature1, feature2, edge classifier, and graph classifier modules. Each feature module consists of a stack of blocks, where each block consists of the graph attention network (GAT) layer [15, 16], batch normalization (BN) layer [17], and ReLU activation function. The feature1 module is a shared representation of the concept of multi-task learning. The graph classifier module consists of an average pooling, which averages node attributes, and a fully connected layer (Linear) to predict the graph-level class, that is, a signal or background event. The edge classifier module is aimed at predicting the edge-level class, which was described in Section 3.1. Edge features are obtained by concatenating the source and target node attributes in the edge classifier module; then, they are processed by Linear, ReLU, and Linear layers.

5 Experiments

Our proposed model for this experiment was implemented using PyTorch [18] and DGL [19] and is available at [20]. All executions used a local cluster of NVIDIA Tesla A100 graphics cards.

A loss function is defined as $L = L^{\text{graph}} + \alpha \cdot L^{\text{edge}}$, where L^{graph} and L^{edge} are loss functions for the event classification task and auxiliary task, respectively. CrossEntropyLoss and BCEWithLogitsLoss functions in PyTorch were used. A parameter of α is introduced to scale the loss value of the auxiliary task. Therefore, a normal event classification without the auxiliary task is performed if $\alpha = 0$. The best epoch for the validation data was used as the final weight parameters after the training was performed for up to 100 epochs to minimize L. The SGD [21] algorithm was used as an optimizer, and the learning rate was decreased from 0.01 to 0.0001 by the cosine annealing algorithm [22]. The batch size was fixed at 2,048. Other hyperparameters, such as the number of nodes in the GAT layer, were optimized by a grid search with the fixed parameter of $\alpha = 0$.

Figure 4 presents obtained AUC values of the event classification task in terms of α. The green points show the values using the auxiliary task labels described in Section 3.1. The figures confirm that the AUCs are improved by adding the auxiliary task, that is, α values are greater than 0. The best α values were 44 (2HDM dataset) and 6 (Z' dataset). The orange points show the values using random labels in the auxiliary task. No improvements were observed by the random labels as expected. The AUC values of the auxiliary task with the best α values were 0.968 (2HDM dataset) and 0.995 (Z' dataset), which were obtained by averaging each class with weights.

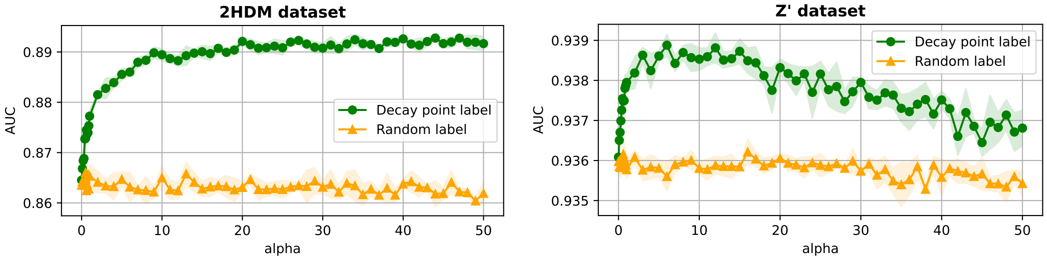


Figure 4: AUC of the event classification task for each dataset. The AUC values are an average of 5 runs with different random seeds for the test data. The error bands show one standard deviation.

BDT and simple MLP models were also executed to compare the performance. Table 1 summarizes observed AUCs for each experimental condition. The "low-level" in the BDT and MLP means that the same input variables with our model, that is, the four-momenta and object-type for each object, were used with a one-dimensional shape. The "w/ high-level" means that the invariant masses of m_{bb}, $m_{\ell\nu bb}$, $m_{\ell\nu qq}$, and m_{bbWW} were calculated from the observed objects, and were added to the input variables to utilize the high-level features. The BDT was executed using XGBoost [23]. The MLP model consisted of a stack of Linear, BN layers, and ReLU activation functions. The max depth in BDT and the number of layers and nodes in MLP were optimized by a grid search. The table confirms that our model with the best α value shows better performances for both 2HDM and Z' datasets. In comparison to the BDT and simple MLP models, our proposed model can extract high-level features effectively based on particle decay information.

Our experiments have the following limitation. The training of our model is much slower than the MLP model, which was approximately 36 batch/s (our model) and 300 batch/s (MLP model). Due to

Table 1: AUC values of the event classification task for each experimental condition. The AUC values are an average of 5 runs with different random seeds for the test data. The errors show one standard deviation. The max depth in BDT is 5 (2HDM) and 4 (Z'). The number of Linear layers in MLP is 5 (2HDM) and 4 (Z'). The number of nodes in each Linear layer is 64 for both datasets.

2HDM	Our model		BDT		MLP	
	$\alpha = 0$	best α value	low-level	w/ high-level	low-level	w/ high-level
2HDM	0.865 ± 0.002	**0.893** **± 0.001**	0.794 ± 0.001	0.838 ± <0.001	0.862 ± 0.001	0.865 ± 0.001
Z'	0.936 ± <0.001	**0.939** **± <0.001**	0.915 ± <0.001	0.922 ± <0.001	0.932 ± <0.001	0.933 ± <0.001

this computational constraint, we were unable to increase the signal types and the number of training events as we had intended. Additionally, we were interested in conducting a more thorough analysis of the edge-level outputs, such as the attention weights in the GAT layer, to increase explainability. Therefore, improving the training speed and explainability is a future subject.

6 Conclusion

In this paper, the decay-aware neural network based on the multi-task learning technique was discussed. The label of auxiliary task is defined based on the particle decay information to introduce an inductive bias with our domain knowledge. We successfully trained the event classification task and the auxiliary task simultaneously using the GNN architecture. Our experiments confirmed that the event classification performance improved significantly by adding the auxiliary task, and better performances were achieved compared with BDT and MLP models due to the inductive bias.

Broader Impact

This work can potentially contribute to the explainability of machine learning, which would be interested in a wide range of scientific communities. In the field of collider physics, researchers have a deep understanding of data based on particle physics theories and past experimental results. We can probe behaviors of the machine learning by introducing the domain knowledge as demonstrated in this paper. We believe that this approach is effective to understand what machine learning has learned and improve its explainability. This study is aimed at improving a problem in pure fundamental science, and we do not expect our study to result in a negative social impact.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number JP22K14050 and Institute for AI and Beyond of the University of Tokyo.

References

[1] Lyndon Evans and Philip Bryant. LHC machine. *Journal of Instrumentation*, Vol. 3, No. 08, pp. S08001–S08001, aug 2008.

[2] Piero Altoe Kim Albertsson and Dustin Anderson *et al*. Machine learning in high energy physics community white paper. *Journal of Physics: Conference Series*, Vol. 1085, p. 022008, sep 2018.

[3] Alexander Radovic, Mike Williams, David Rousseau, Michael Kagan, Daniele Bonacorsi, Alexander Himmel, Adam Aurisano, Kazuhiro Terao, and Taritree Wongjirad. Machine learning at the energy and intensity frontiers of particle physics. *Nature*, Vol. 560, pp. 41–48, 2018.

[4] Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for Exotic Particles in High-Energy Physics with Deep Learning. *Nature Commun.*, Vol. 5, p. 4308, 2014.

[5] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks, 2018.

[6] Murat Abdughani, Jie Ren, Lei Wu, and Jin Min Yang. Probing stop pair production at the LHC with graph neural networks. *Journal of High Energy Physics*, Vol. 2019, No. 8, aug 2019.

[7] Jie Ren, Lei Wu, and Jin Min Yang. Unveiling CP property of top-higgs coupling with graph neural networks at the LHC. *Physics Letters B*, Vol. 802, p. 135198, mar 2020.

[8] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H. S. Shao, T. Stelzer, P. Torrielli, and M. Zaro. The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. *JHEP*, Vol. 07, p. 079, 2014.

[9] Torbjörn Sjöstrand, Stefan Ask, Jesper R. Christiansen, Richard Corke, Nishita Desai, Philip Ilten, Stephen Mrenna, Stefan Prestel, Christine O. Rasmussen, and Peter Z. Skands. An introduction to PYTHIA 8.2. *Comput. Phys. Commun.*, Vol. 191, pp. 159–177, 2015.

[10] Michele Selvaggi. DELPHES 3: A modular framework for fast-simulation of generic collider experiments. *Journal of Physics: Conference Series*, Vol. 523, p. 012033, jun 2014.

[11] G.C. Branco, P.M. Ferreira, L. Lavoura, M.N. Rebelo, Marc Sher, and Joao P. Silva. Theory and phenomenology of two-higgs-doublet models. *Physics Reports*, Vol. 516, No. 1, pp. 1–102, 2012. Theory and phenomenology of two-Higgs-doublet models.

[12] Celine Degrande. Automatic evaluation of UV and R2 terms for beyond the Standard Model lagrangians: A proof-of-principle. *Computer Physics Communications*, Vol. 197, , 06 2014.

[13] Guido Altarelli, B. Mele, and M. Ruiz-Altaba. Searching for New Heavy Vector Bosons in $p\bar{p}$ Colliders. *Z. Phys. C*, Vol. 45, p. 109, 1989. [Erratum: Z.Phys.C 47, 676 (1990)].

[14] B. Fuks and R. Ruiz. A comprehensive framework for studying W′ and Z′ bosons at hadron colliders with automated jet veto resummation. *Journal of High Energy Physics*, Vol. 2017, pp. 1–38, 2017.

[15] Shaked Brody, Uri Alon, and Eran Yahav. How Attentive are Graph Attention Networks? . In *International Conference on Learning Representations*, 2022.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc., 2017.

[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 448–456. JMLR.org, 2015.

[18] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

[19] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks. *arXiv preprint arXiv:1909.01315*, 2019.

[20] hepdecay. Available: `https://github.com/ktomoe/hepdecay`, 2022. (Accessed: Oct. 24, 2024).

[21] Sebastian Ruder. An overview of gradient descent optimization algorithms, 2016.

[22] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts, 2016.

[23] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794, New York, NY, USA, 2016. ACM.

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] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See 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] See Section 5.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 3 and 5.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Figure 4 and Table 1.
   (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] See Section 5.
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? [N/A]
   (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]