Reproducing Machine Learning Research on Binder

Anonymous Author(s)
Affiliation
Address
`email`

1 Introduction

For decades, computational scientists have highlighted the importance of publishing the software pipelines associated with a given research publication. In 1995, Buckheit and Donoho [6] summarized the work of Claerbout and Karrenbach [9] by arguing,

> An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.

The adoption of open-source tools to write machine learning pipelines, often in Python [66], has provided researchers with access to an author's experiments or allowed them to replicate a study by reimplementing an algorithm. Open-source libraries popular in machine learning experiments include Jupyter Notebooks [27], NumPy [65], CodaLab [30], TensorFlow [2], PyTorch [39]. To share these experiments, researchers use platforms such as OpenML [67], Papers with Code [61], Code Ocean, Inc. [12], RunMyCode [57], Colaboratory [21], and GitHub, Inc. [20]. These platforms have all developed rich communities of researchers dedicated to open science, though many of the deployments are closed-source or run by a single company or project.

Binder [17, 44, 48, 16, 40] is an open-source project that lets users share interactive, reproducible science. Binder's goal is to allow researchers to create interactive versions of their code utilizing pre-existing workflows and minimal additional effort. It uses standard configuration files in software engineering to let researchers create interactive versions of code they have hosted on commonly-used platforms like GitHub.

Binder's underlying technology, BinderHub, is entirely open-source and utilizes entirely open-source tools. By leveraging tools such as Kubernetes [10] and Docker [14], it manages the technical complexity around creating containers to capture a repository and its dependencies, generating user sessions, and providing public URLs to share the built images with others. BinderHub combines two open-source projects within the Jupyter ecosystem: repo2docker [45, 15] and JupyterHub [42]. repo2docker builds the Docker image of the git repository specified by the user, installs dependencies, and provides various front-ends to explore the image. JupyterHub then spawns and serves instances of these built images using Kubernetes to scale as needed (Figure 1b). Because each of these pieces is open-source and uses popular tools in cloud orchestration, BinderHub can be deployed on a variety of cloud platforms, or even on your own hardware.

One example of a BinderHub deployment is at mybinder.org [41], a free public service that the BinderHub team maintains. Over 3,000 public repositories have been built using mybinder.org, covering topics such as LIGO's gravational waves [31], textbooks on Kalman Filters [28], and open-source libraries such as PyMC3 [50]. As of September 2018, mybinder.org serves an average of 8,000 users per day and has served as many as 22,000 a given day. For NIPS 2018, we plan to share a Binder deployment that would feature machine learning research repositories from the open-source community.

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

2 Leveraging Common Practices in Scientific Computing

Binder provides scalable, open-source, interactive computing in a language- and platform-agnostic manner. Many researchers don't share Dockerfiles in git repos [37], so it can be difficult to fully describe and replicate the environment used for a machine learning experiment. As a result, many researchers struggle to find the correct configuration of dependencies used by the author, resulting in dependency hell [3]. Using mybinder.org, they can build and share images of their existing repos by following best practices in computational science (such as specifying dependencies in a requirements.txt file). To build a Docker image, Binder simply requires configuration files typical in Python, R [18], and Julia [5] programming that are hosted on online platforms such as GitHub, GitLab, or Bitbucket. Its underlying tool, repo2docker, is inspired by Heroku buildpacks [23] and tailored to software conventions used in scientific computing. These configuration files include Python's `setup.py` conda's `environment.yml`, pip's `requirements.txt`, and Julia's `REQUIRE`. Binder also accepts `start` and `postBuild` scripts that allows the author of a repository to run additional software at runtime or following the building of a Docker image. The binder-examples organization on GitHub provides simple repository examples of how one can use these configuration files to build Docker images [46, 48]. While the majority of repositories shared with mybinder.org are written in Python, R, or Julia, members of the open-source community have also shared repositories written in Go [63, 69], C++ [62, 8, 51, 13] and Haskell [24, 19].

Because BinderHub is open-source, a Binder service can be deployed on any system that supports Kubernetes. We provide an online tutorial, Zero to BinderHub [47], to teach anyone how to deploy their own BinderHub on their own server. We also provide Helm Charts [11] to manage the configuration of Binder's Kubernetes cluster [10]. The tutorials can be completed in a manner of hours on major cloud providers, and have been deployed by several institutions. The Binder team has a curated list of public BinderHub deployments [49] which includes the Leibniz Institute for the Social Sciences [29] and Pangeo Contributors [38].

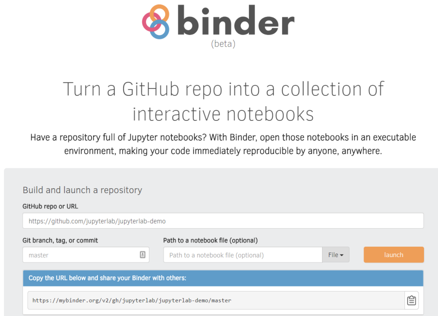
3 Interacting with Research Software

In our demonstration, we will showcase a deployment of BinderHub and feature research repositories from the machine learning community. Users of mybinder.org can easily build images by entering the URL of a GitHub, GitLab, or other git repository (Figure 1a). We hope to give attendees the opportunity to build images of their desired repositories and interact with them on Binder. Once a repository is built on Binder, no additional installation is necessary to run the repository's code on our server. Anyone can access a built image with simply the mybinder.org URL to the image. While we plan to bring a laptop and monitor for our demonstration, anyone can access mybinder.org for free at any time. We will also share public metrics on our mybinder.org deployment such as our Grafana dashboard [1] and our public cost calculator [26] provide additional transparency on how Binder works.

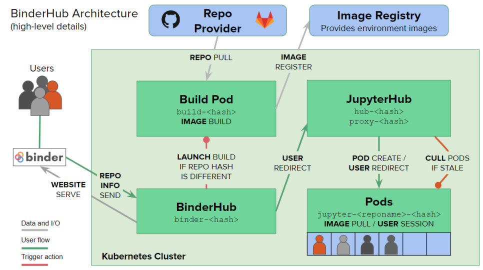
We also will feature machine learning publications from GitHub with reproducible examples by building them on Binder to share with the NIPS community. These research repositories can be explored with JupyterLab [43] and Jupyter Notebook so that attendees can run code in the built image and query models. We demonstrate how one can query the pre-trained model from Mascharka et al. [35] in Figure 2a. Because Binder provides an interactive environment, attendees can also modify the code presented in the repository to alter the experiments of the authors. In Figure 2b, we modify an experiment from Ross et al. [56]. Our public deployment, mybinder.org, currently features built images by the authors Mascharka et al. [35], Ross et al. [56], and Lundberg and Lee [33]. We also have re-implementations of Vaswani et al. [68] by Rush [58] and Rajpurkar et al. [53] by Zech [70]. We would like to demonstrate how Binder can be used to evaluate, reproduce, and extend research [16] based on a research paper's repository. We hope that by sharing publications on Binder and providing attendees the opportunity to interact with the repositories, researchers will deploy their own Binder to share their research.

4 Emphasizing Reproducible Science

We believe that tools such as Binder can be used to help solve problems in creating reproducible science. Baker [4] surveyed 1,500 scientists and found that over 70% had reported a failed attempt to

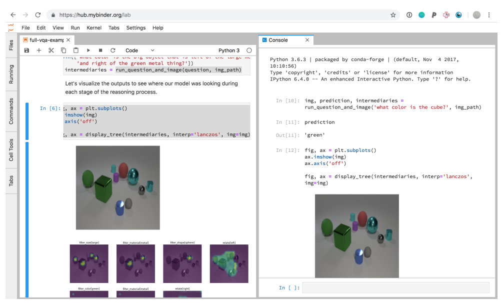


(a) The BinderHub user interface, which allows users to input a link to a public git repository.

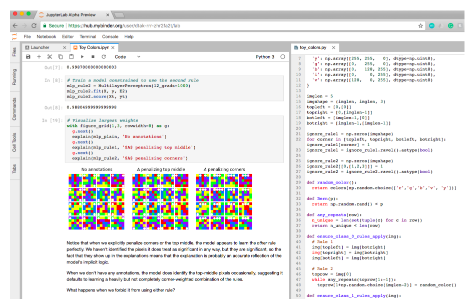


(b) The BinderHub architecture for interactive sessions.

Figure 1: BinderHub's UI and architecture. The user enters a URL to a public git repository, which Binder will use to build a Docker image. Binder provides a URL to the image so that they may run an interactive session that runs the repository's code. The Kubernetes deployment (light green) manages the pods (dark green) that make up BinderHub. Interactive user pods (blue squares) are spawned and managed by JupyterHub.



(a) Exploring predictions from Mascharka et al. [35]



(b) Extending experiments in Ross et al. [56]

Figure 2: Because Binder includes all software with dependencies pre-installed, we can use Binder to examine the experimental pipeline of a paper with tools such as JupyterLab. In Figure 2a, we run the authors' notebook on the left and query the pre-trained model provided by the authors to test its predictions using the console on the right. In Figure 2b, we have modified the Python file of a toy-color experiment on the right to include the color yellow, which is shown in the notebook on the left.

reproduce a colleague's work and over half had failed to reproduce their own work. Developments in machine learning ablation studies, which externally compare algorithmic methods for a given task, suggests that researchers are growing concerned with their ability to reproduce work in the field [25, 68, 36, 22, 34, 64, 55]. Researchers also have shown how machine learning systems have difficulty safely generalizing in real-world deployments [72, 60, 7, 54, 71]. Providing interactive, working research pipelines to the public for examination helps researchers inspect the methods applied by authors and independently evaluate performance on the and data models provided. They can also modify the experimental pipeline's code or obtain predictions on new data. While tools to share research software cannot address all concerns within the machine learning community regarding the state of scholarship today [32, 59], we hope easy access to experiments can help researchers in "understanding and explaining phenomena" within machine learning systems [52].

References

[1] Binder grafana board. `https://grafana.mybinder.org/?orgId=1`. URL `https://grafana.mybinder.org/?orgId=1`. Accessed: 2018-5-23.

[2] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, and Others. Tensorflow: a system for large-scale machine learning. In *OSDI*,

volume 16, pages 265–283. usenix.org, 2016. URL https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf.

[3] R. Anderson. The end of DLL hell. *Microsoft Developer Network*, Jan. 2000. URL https://web.archive.org/web/20010605023737/http://msdn.microsoft.com/library/techart/dlldanger1.htm.

[4] M. Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, May 2016. URL http://dx.doi.org/10.1038/533452a.

[5] J. Bezanson, A. Edelman, S. Karpinski, and V. Shah. Julia: A fresh approach to numerical computing. *SIAM Rev.*, 59(1):65–98, Jan. 2017. URL https://doi.org/10.1137/141000671.

[6] J. B. Buckheit and D. L. Donoho. WaveLab and reproducible research. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*, pages 55–81. Springer New York, New York, NY, 1995. URL https://doi.org/10.1007/978-1-4612-2544-7_5.

[7] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In S. A. Friedler and C. Wilson, editors, *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 77–91, New York, NY, USA, 2018. PMLR. URL http://proceedings.mlr.press/v81/buolamwini18a.html.

[8] CERN PH-SFT. rootbinder, 2015. URL https://github.com/cernphsft/rootbinder.

[9] J. F. Claerbout and M. Karrenbach. Seismology on CD-ROM. https://web.archive.org/web/19981202153004/http://sepwww.stanford.edu:80/sep/jon/blurb.html, 1994. URL https://web.archive.org/web/19981202153004/http://sepwww.stanford.edu:80/sep/jon/blurb.html. Accessed: 2018-9-28.

[10] Cloud Native Computing Foundation. Kubernetes. URL https://kubernetes.io/.

[11] Cloud Native Computing Foundation. Helm - the package manager for kubernetes. https://docs.helm.sh/developing_charts, 2015. URL https://docs.helm.sh/developing_charts. Accessed: 2018-9-30.

[12] Code Ocean, Inc. Code ocean. https://codeocean.com/. URL https://codeocean.com/. Accessed: 2018-9-25.

[13] DIANA/HEP. pyhf, 2018. URL https://github.com/diana-hep/pyhf.

[14] Docker, Inc. Docker. https://www.docker.com/. URL https://www.docker.com/. Accessed: 2018-5-24.

[15] J. Forde, T. Head, C. Holdgraf, Y. Panda, G. Nalvarete, B. Ragan-Kelley, and E. Sundell. Reproducible research environments with Repo2Docker. In *Reproducibility in ML Workshop, ICML'18*, June 2018. URL https://openreview.net/forum?id=B1lYOwuoxm.

[16] Forde J,Holdgraf C, Panda Y, Culich A, Bussonnier M, Ragan-Kelley B, Pacer M, Willing C, Head T, Perez F, Granger B, Project Jupyter Contributors. Post-training evaluation with binder. In *Conference on Fairness, Accountability, and Transparency. Online. Available at https://fat conference. org/static/tutorials/forde_binder18. pdf. Accessed March*, volume 22, page 2018, 2018. URL https://github.com/jupyterhub/binder/issues.

[17] J. Freeman and A. Osheroff. Toward publishing reproducible computation with binder. https://elifesciences.org/labs/a7d53a88/toward-publishing-reproducible-computation-with-binder, May 2016. URL https://elifesciences.org/labs/a7d53a88/toward-publishing-reproducible-computation-with-binder. Accessed: 2017-12-11.

[18] R. Gentleman and D. T. Lang. Statistical analyses and reproducible research. *J. Comput. Graph. Stat.*, 16(1):1–23, 2007. URL http://www.jstor.org/stable/27594227.

[19] A. Gibiansky. IHaskell, 2013. URL https://github.com/gibiansky/IHaskell.

[20] GitHub, Inc. GitHub. https://github.com. URL https://github.com. Accessed: 2018-9-25.

[21] Google. Google colaboratory. https://colab.research.google.com/. URL https://colab.research.google.com/. Accessed: 2018-9-25.

[22] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger. Deep reinforcement learning that matters. Sept. 2017. URL http://arxiv.org/abs/1709.06560.

[23] Heroku. Heroku buildpacks. https://www.heroku.com/elements/buildpacks. URL https://www.heroku.com/elements/buildpacks. Accessed: 2018-9-25.

[24] P. Hudak, S. Peyton Jones, P. Wadler, B. Boutel, J. Fairbairn, J. Fasel, M. M. Guzmán, K. Hammond, J. Hughes, T. Johnsson, D. Kieburtz, R. Nikhil, W. Partain, and J. Peterson. Report on the programming language haskell: A non-strict, purely functional language version 1.2. *SIGPLAN Not.*, 27(5):1–164, May 1992. URL http://doi.acm.org/10.1145/130697.130699.

[25] K. Jamieson and B. Recht. The news on auto-tuning. http://benjamin-recht.github.io/2016/06/20/hypertuning/, 2016. URL http://benjamin-recht.github.io/2016/06/20/hypertuning/. Accessed: 2018-9-25.

[26] JupyterHub. binder-billing, 2018. URL https://github.com/jupyterhub/binder-billing.

[27] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. E. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. B. Hamrick, J. Grout, S. Corlay, and Others. Jupyter notebooks-a publishing format for reproducible computational workflows. In *ELPUB*, pages 87–90. books.google.com, 2016. URL https://books.google.com/books?hl=en&lr=&id=Lgy3DAAAQBAJ&oi=fnd&pg=PA87&dq=jupyter&ots=N1A_5NtAkj&sig=wxwF_hRUStOKTzvFFFXz4u8J-AE.

[28] R. Labbe. Kalman-and-Bayesian-Filters-in-Python. URL https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python.

[29] Leibniz Institute for the Social Sciences. Open research computing. URL https://github.com/gesiscss/orc.

[30] P. Liang and E. Viegas. CodaLab worksheets for reproducible, executable papers, Dec. 2015. URL https://nips.cc/Conferences/2015/Schedule?showEvent=5779.

[31] LIGO Scientific Collaboration. LIGO open science center. https://losc.ligo.org/tutorials/. URL https://losc.ligo.org/tutorials/. Accessed: 2017-12-12.

[32] Z. C. Lipton and J. Steinhardt. Troubling trends in machine learning scholarship. In *ICML 2018: The Debates*, July 2018. URL http://arxiv.org/abs/1807.03341.

[33] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[34] H. Mania, A. Guy, and B. Recht. Simple random search provides a competitive approach to reinforcement learning. Mar. 2018. URL http://arxiv.org/abs/1803.07055.

[35] D. Mascharka, P. Tran, R. Soklaski, and A. Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4942–4950, 2018. URL http://openaccess.thecvf.com/content_cvpr_2018/papers/Mascharka_Transparency_by_Design_CVPR_2018_paper.pdf.

[36] G. Melis, C. Dyer, and P. Blunsom. On the state of the art of evaluation in neural language models. In *ICLR 2018*. arxiv.org, July 2017. URL http://arxiv.org/abs/1707.05589.

[37] Y. Panda. Why repo2docker? why not s2i?, Dec. 2017. URL http://words.yuvi.in/post/why-not-s2i/. Accessed: 2018-6-21.

[38] Pangeo Contributors. Pangeo BinderHub. URL https://github.com/pangeo-data/pangeo-binder.

[39] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. Oct. 2017. URL https://openreview.net/pdf?id=BJJsrmfCZ.

[40] Project Jupyter, M. Bussonnier, J. Forde, J. Freeman, B. Granger, T. Head, C. Holdgraf, K. Kelley, G. Nalvarte, A. Osheroff, M. Pacer, Y. Panda, F. Perez, B. Ragan-Kelley, and C. Willing. Binder 2.0 - reproducible, interactive, sharable environments for science at scale. In *Proceedings of the 17th Python in Science Conference*, Proceedings of the Python in Science Conference, pages 113–120. SciPy, 2018. URL https://conference.scipy.org/proceedings/scipy2018/project_jupyter.html.

[41] Project Jupyter Contributors. Binder (beta). https://mybinder.org/. URL https://mybinder.org/. Accessed: 2017-12-11.

[42] Project Jupyter Contributors. jupyterhub, 2014. URL https://github.com/jupyterhub/jupyterhub.

[43] Project Jupyter Contributors. JupyterLab, July 2015. URL https://github.com/jupyterlab/jupyterlab.

[44] Project Jupyter Contributors. binderhub, 2017. URL https://github.com/jupyterhub/binderhub.

[45] Project Jupyter Contributors. repo2docker, Apr. 2017. URL https://github.com/jupyter/repo2docker/.

[46] Project Jupyter Contributors. Using R with jupyter / RStudio on binder, 2017. URL https://github.com/binder-examples/r.

[47] Project Jupyter Contributors. *Zero to BinderHub*, 2017. URL https://binderhub.readthedocs.io/en/latest/.

[48] Project Jupyter Contributors. Julia binder demo, July 2017. URL https://github.com/choldgraf/demo-julia.

[49] Project Jupyter Contributors. BinderHub deployments. https://binderhub.readthedocs.io/en/latest/known-deployments.html, 2018. URL https://binderhub.readthedocs.io/en/latest/known-deployments.html. Accessed: 2018-9-30.

[50] PyMC Developers. pymc3. URL https://github.com/pymc-devs/pymc3.

[51] QuantStack. xeus-cling, 2017. URL https://github.com/QuantStack/xeus-cling.

[52] A. Rahimi and B. Recht. An addendum to alchemy. http://benjamin-recht.github.io/2017/12/11/alchemy-addendum/, Dec. 2017. URL http://benjamin-recht.github.io/2017/12/11/alchemy-addendum/. Accessed: 2018-9-30.

[53] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng. CheXNet: Radiologist-Level pneumonia detection on chest X-Rays with deep learning. Nov. 2017. URL http://arxiv.org/abs/1711.05225.

[54] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do CIFAR-10 classifiers generalize to CIFAR-10? June 2018. URL http://arxiv.org/abs/1806.00451.

[55] C. Riquelme, G. Tucker, and J. Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. In *ICLR 2018*, Feb. 2018. URL https://openreview.net/forum?id=SyYe6k-CW.

[56] A. S. Ross, M. C. Hughes, and F. Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages Pages 2662–2670., Mar. 2017. URL https://www.ijcai.org/proceedings/2017/371.

[57] RunMyCode. Run my code. http://www.runmycode.org/. URL http://www.runmycode.org/. Accessed: 2018-9-25.

[58] A. Rush. annotated-transformer, 2018. URL http://nlp.seas.harvard.edu/2018/04/03/attention.html.

[59] D. Sculley, J. Snoek, A. Wiltschko, and A. Rahimi. Winner's curse? on pace, progress, and empirical rigor. In *ICLR 2018 Workshop*, Feb. 2018. URL https://openreview.net/forum?id=rJWF0Fywf.

[60] P. Stock and M. Cisse. ConvNets and ImageNet beyond accuracy: Understanding mistakes and uncovering biases. Nov. 2017. URL http://arxiv.org/abs/1711.11443.

[61] R. Stojnic and R. Taylor. Papers with code. https://paperswithcode.com/. URL https://paperswithcode.com/. Accessed: 2018-9-25.

[62] B. Stroustrup. The c++ programming language. 2000. URL http://117.3.71.125:8080/dspace/bitstream/DHKTDN/7135/1/4876.The%20C%2B%2B%20programming%20language,%20third%20edition.pdf.

[63] The Go Authors. The go programming language, 2009. URL https://golang.org/.

[64] G. Tucker, S. Bhupatiraju, S. Gu, R. E. Turner, Z. Ghahramani, and S. Levine. The mirage of Action-Dependent baselines in reinforcement learning. In *ICLR 2018 Workshop*, Feb. 2018. URL https://openreview.net/pdf?id=HyL0IKJwM.

[65] S. van der Walt, S. C. Colbert, and G. Varoquaux. The NumPy array: A structure for efficient numerical computation. *Comput. Sci. Eng.*, 13(2):22–30, Mar. 2011. URL https://aip.scitation.org/doi/abs/10.1109/MCSE.2011.37.

[66] G. van Rossum. Python reference manual. *Department of Computer Science [CS]*, (R 9525), Jan. 1995. URL https://ir.cwi.nl/pub/5008/05008D.pdf.

[67] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, June 2014. URL https://dl.acm.org/citation.cfm?doid=2641190.2641198.

[68] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. U. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[69] Y. Watanabe. lgo, 2017. URL https://github.com/yunabe/lgo.

[70] J. Zech. reproduce-chexnet, 2018. URL https://github.com/jrzech/reproduce-chexnet.

[71] J. R. Zech, M. A. Badgeley, M. Liu, A. B. Costa, J. J. Titano, and E. K. Oermann. Confounding variables can degrade generalization performance of radiological deep learning models. July 2018. URL http://arxiv.org/abs/1807.00431.

[72] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. Nov. 2016. URL http://arxiv.org/abs/1611.03530.