https://doi.org/10.1038/s42256-024-00835-5

The rewards of reusable machine learning code

Check for updates

Research papers can make a long-lasting impact when the code and software tools supporting the findings are made readily available and can be reused and built on. Our reusability reports explore and highlight examples of good code sharing practices.

The availability of increasingly advanced deep learning tools in the past two decades has transformed scientific research in most, if not all, disciplines. 'AI for Science' is a popular theme at key machine learning conferences such as NeurIPS and ICML and in academic as well as industry research labs. Example application areas include protein design, materials discovery, precision medicine, quantum computing and analysis of complex dynamical systems such as in engineering or climate modelling.

Research efforts that involve well-designed machine learning tools can be of long-lasting value to the research community and beyond, provided that the methods, datasets and code are clearly described and shared. In recent years, we have observed a clear rise in standards regarding availability of code and data in submitted manuscripts, which is good news for open science and reproducibility.

Our editorial policies mandate that authors share code used to produce results that are central to the main claims. Code should be made available to referees during the peer review process, and then released publicly upon publication. We ask referees to review the code and, if possible, to try and run it and reproduce the findings in a paper. To facilitate this process, authors have the option to upload their code in the form of executable compute capsules via the Code Ocean platform. This enables referees to access code without needing to install various libraries or software packages. Accessing Code Ocean compute capsules that accompany a manuscript has been simplified, as weblinks are now integrated in our online manuscript system. Similar weblinks are also provided to upload data on the Figshare repository.

The availability of code and data, and the reproducibility of the main research findings, are important goals during the peer review process; however, the gold standard is ensuring that code can be re-implemented, extended and reused by other researchers on different datasets and applications. Making code fully reusable can be a tall order as research groups may not have the resources to develop and maintain well-structured code repositories and accompanying documentation. At a minimum, we expect authors to provide a clear README file to describe the code and its intended uses, an overview of dependencies, and some example data. If applicable, a description of the pre-training process and a pre-trained model should be provided. Furthermore, authors should provide a license to explain terms of use and redistribution and mint a digital object identifier (DOI) to ensure that a permanent version exists that is associated with the published paper.

To highlight the value of high-quality code developments, we introduced an article format in 2020 known as 'reusability reports'[1], which are dedicated to testing robustness, extendability and reusability of previously published code. So far, 12 reusability reports have been published and we are encouraged by the consistently positive feedback from authors and referees. We regularly send out invitations to write a reusability report linked to selected accepted papers that have promising code development. But we are also happy to receive proposals for such articles, which can be linked to papers published in *Nature Machine Intelligence* or elsewhere. However, for critical comments on articles that highlight issues with reproducibility or other technical problems, authors should refer to the Matters Arising format. Reusability reports undergo peer review and count as primary research articles.

Since last year, reusability reports are based on our regular research article type to enable more space and ensure a similar editorial and peer review process. In contrast to regular Articles, we do not assess novelty in reusability reports but instead examine whether reusability is tested in technically correct and interesting ways, and whether clear value is added with respect to the original article. To highlight some examples, in a reusability report in this issue, Tao Xu et al. test a recent bilinear attention model for predicting drug–target interactions, with adaptability across domains Xu et al. study and highlight this generalization capability of the model and also apply the method on a task not explored in the original publication — the prediction of cell line–drug responses. In another example[2], Yingying Cao et al. tested and re-used code from PENCIL, a supervised method to identify cell populations with specific phenotypes from single-cell RNA data. They found that the method can be combined with an approach called gene set variation analysis to predict responses to immune checkpoint blockade therapy in several skin cancer datasets[2]. As a last example, Yuhe Zhang et al.[3] revisited a deep learning method that uses a parameterized physical forward model to reconstruct holographic images and extended it to non-perfect optical systems (for example, subjected to noise or blurring) by incorporating system-specific response functions in the forward propagator.

The development of good quality code and software, which can be re-implemented and extended to new data, even beyond the original scope, can catalyse further research and inspire new directions. As our colleagues at *Nature Computational Science* wrote a few years ago[4], efforts in code and software development deserve more praise and recognition, to start with by ensuring that code is clearly shared, discoverable, and citable. As we expand our series of reusability reports, we hope to contribute to a virtuous circle of high standards in code development and sharing, and more recognition and reward for such efforts.

Published online: 24 April 2024

References

1. *Nat. Mach. Intell.* **2**, 729 (2020).
2. Cao, Y., Chang, T. G., Sahni, S. & Ruppin, E. *Nat. Mach. Intell.* **6**, 307–314 (2024).
3. Zhang, Y., Ritschel, T. & Villanueva-Perez, P. *Nat. Mach. Intell.* **6**, 284–290 (2024).
4. *Nat. Comput. Sci.* **1**, 89 (2021).