A web of tidings

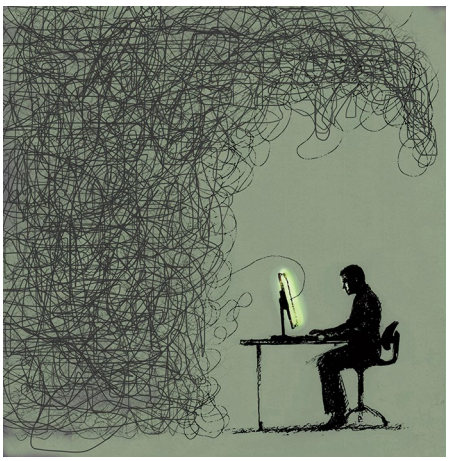
Preprints provide an efficient way for scientific communities to share and discuss results. We encourage authors to post preprints on arXiv, bioRxiv or other recognized community preprint platforms.

Nature journals have welcomed submissions that also exist as preprints for over two decades[1], but we still hear surprisingly often that scientists are unsure about our policy. To highlight more clearly that we recognize the important role of preprint posting in the process of scientific discourse, *Nature Machine Intelligence* offers authors the option to add a link back to their arXiv or bioRxiv preprint from the published paper, visible to all readers. Readers can find two examples of this function in the current issue — the Articles by Brendon Lutnick et al. and William Severa et al.

A well-known story is that the first preprint server arXiv had its origin in high-energy physics. Paul Ginsparg, a physicist at Los Alamos National Laboratory in New Mexico, decided to launch an electronic bulletin for sharing unpublished papers with colleagues and friends[2]. It was 1991 and the main mode of exchanging papers was mailing paper copies to one another. Even though it hardly seems long ago for some of us, this was a time of big, noisy desk computers, floppy disks and a world without a wide web.

But with the next information revolution around the corner and the rapid uptake of the world wide web during the rest of the decade, arXiv was well positioned to serve a physics community eager to benefit from the rapid flow of information. It expanded quickly, taking other fields on board: computer science, mathematics, economics and several more. At arXiv's 20-year anniversary in 2011 Paul Ginsparg noted[2], presumably with a mixture of pride and concern, that no community that had adopted arXiv had renounced it. The server's growth has been unstoppable and currently a staggering 1.5 million preprints exist on arXiv. In 2018, 140 thousand new submissions were posted, up by 14% from 2017.

Different scientific communities have different ways of disseminating and discussing their findings prior to peer-reviewed publication: preprint posting has become popular in many fields, while others stick to conferences or other platforms. The reasons scientists give for posting preprints centre on being able to share results without delay with experts and getting feedback from the community, putting a time-stamp on the work to establish priority, and increasing visibility.



Credit: Gary Waters/Alamy Stock Photo

Already in 2011, the increase in arXiv preprint posting in the field of computer science was noted. The growth continued and in 2018, the field accounted for 26% of new submissions, second after physics. There is no doubt that the rapid interactions allowed by preprint sharing contributed to the quick advances in the field of machine learning in the past decade. But the daily output in this field and related ones has made it a challenge for anyone trying to keep up with latest developments. In 2016, arXiv began planning for a significant overhaul[3], first to improve the infrastructure to deal with the high volumes of preprints, but also to offer a more user-friendly interface. External tools to browse and filter content now also exist and a popular one is 'arxiv sanity preserver'[4], which attempts to tame the flood of preprints, thereby 'preserving the sanity' of anyone trying to keep up with new papers on arXiv. This tool filters by subject terms and popularity, provides a useful interface for skimming through content and offers several ways to sort papers. It is running live on the web for the subjects in computer science and statistics closely associated with artificial intelligence and machine learning — from a quick count on www.arxiv-sanity.com, currently close to a 100 preprints are posted per day in these topics.

Some other fields, such as the life sciences, have been slower to embrace open preprint sharing. However, bioRxiv came into existence in 2013, and quickly became highly popular. Unlike arXiv, it offers a commenting function and provides metrics on article usage and attention. ChemRxiv launched in 2017 to serve the chemistry community.

With the year-on-year growth in scientific output, an ongoing challenge is how to filter papers, provide quality control, integrate papers with data, code and other tools, and make papers reach intended and new audiences. Such questions, tangled up with the future of science publishing, have been discussed for at least two decades. While there are no simple answers, the availability and wide acceptance of central repositories for preprints ensures open scientific discourse. ❐

Published online: 11 February 2019
https://doi.org/10.1038/s42256-019-0027-2

References

1. *Nature* **434**, 257 (2005).
2. Ginsparg, P. *Nature* **476**, 145–147 (2011).
3. Van Noorden, R. *Nature* **534**, 602–602 (2016).
4. *GitHub* https://github.com/karpathy/arxiv-sanity-preserver (2018).

Educational strategies to foster diversity and inclusion in machine intelligence

To attract and retain talent from all backgrounds, new educational models and mentorship programmes are needed in machine intelligence, says Shannon Wongvibulsin.

Shannon Wongvibulsin

Machine intelligence is making an increasing impact on society in areas such as transportation, sustainability, agriculture, healthcare and finance[1]. Nevertheless, the diversity of society is not currently represented in the machine intelligence community. Specifically, education and workforce data from the United States indicate that women and minority groups, including African Americans, Hispanics, Native Americans and people with disabilities, are underrepresented[2]. Although these groups constitute over half of the population, of the undergraduate computer science degrees awarded at non-profit institutions, only 20% are to women, 8.3% to African Americans, 11% to Hispanics and 0.4% to Native Americans. Within industry, only 26% of entry-level technical positions are women, 8.3% African Americans and 6.3% Hispanics[3,4].

Without the diversity and inclusion of the full range of the population, we risk the development of biased algorithms and a lack of the necessary talent pool required to fill the growing workforce needed to address the expanding impact of this field on research and developments in other disciplines and in society in general[5,6]. Numerous organizations have recognized the need to foster diversity and inclusion in machine intelligence and have created programmes such as CSforAll (https://www.csforall.org), Code.org (https://code.org), Girls Who Code (https://girlswhocode.com), Black Girls Code (http://www.blackgirlscode.com) and AI4ALL (http://ai-4-all.org).

While these programmes have begun to make an impact with the general objective of early exposure to the field to foster interest and provide foundational skills and knowledge, it is clear that the issues surrounding diversity and inclusion remain difficult to address. This Comment shares key proposals concerning an accessible education and mentorship structure to promote a sustainable infrastructure for active participation, as well as retention and long-term success within the machine intelligence community for individuals of all backgrounds.

Building a welcoming culture

For individuals belonging to groups currently underrepresented in the field, machine intelligence can seem to be an exclusive discipline, open only to those with strong technical backgrounds. Offering introductory courses with no prerequisites can help minimize this impression. Such courses can provide the fundamentals for long-term success in machine intelligence by teaching programming skills, critical thinking, mathematical concepts and the foundations of machine learning algorithms. Furthermore, these courses can serve as the bridge to excite students with no prior technical background to pursue further education in this area.

For example, at Johns Hopkins University, the Hopkins Engineering Applications & Research Tutorials (HEART) programme provides undergraduates with the opportunity to learn about cutting-edge engineering research and its societal impact[7]. These courses are designed and taught by advanced graduate students and postdoctoral fellows, and have no prerequisites to ensure that the classes are accessible to entering undergraduates. Additionally, the class sizes are kept small (typically around ten students) to facilitate an interactive learning environment with ample student–instructor interaction. I designed and instructed HEART Foundations of Statistical Machine Learning, which introduces both the theoretical foundations of modern statistical machine learning models and the implementation of these algorithms in the R programming language. The class is structured to include lectures, discussions and R labs. The discussion part provides an opportunity for students to think about the real-world applications of machine learning algorithms, discuss their ideas in smaller groups, and then share with the class. The R labs offer students the chance to practice their programming skills, see the algorithms discussed in action, and obtain feedback on and assistance with their coding from the instructor as well as their peers.

HEART Foundations of Statistical Machine Learning was offered for the first time in the fall 2018 semester. In a survey at the conclusion of the course, 9 of the 10 students indicated that after taking the course, they were more interested in statistical machine learning and the remaining one student indicated equal interest in statistical machine learning from before the course. The students found the interactive portions of the class and connections to real-world applications most enjoyable. Designing and teaching HEART Foundations of Statistical Machine Learning reinforced the importance of offering accessible courses to excite students from diverse backgrounds. Furthermore, it highlighted possible approaches to mitigate some of the challenges associated with fostering a sustainable infrastructure for diversity and inclusion, through pedagogical models that transform the student into the teacher, active learning in a 'flipped classroom', and longitudinal outreach programmes.

Student as teacher educational models

Teaching peers can increase confidence and mastery of the material. Furthermore, building constructive relationships with peers can further promote a sense of belonging within the community and help foster inclusivity. Machine intelligence education could benefit from borrowing ideas from pedagogical models such as 'see one, do one, teach one', which is common in medical education[8]. For instance, in medical training, the junior doctor or medical student often learns a new procedure by seeing one performed by another healthcare professional, then doing one under supervision and then teaching one to another trainee. The machine intelligence community could implement a similar strategy by structuring introductory courses to include a seeing component (for example, lecture), a doing component (for example,

coding lab) and a teaching component (for example, peer teaching or outreach to teach high school students). Transforming the student into the teacher offers the potential to reinforce the course material, build strong peer relationships, and amplify the impact of the course through outreach projects.

Active learning in a flipped classroom

Rather than using class time for lecture, the flipped classroom minimizes lecture-based instruction and promotes active learning in the classroom[9]. Active learning has been shown to increase students' performance in science, engineering and mathematics[10]. For instance, HEART Foundations of Statistical Machine Learning facilitates an active learning environment through problem-solving activities, discussion, coding exercises and connections to real-world challenges. This flipped classroom structure further encourages students to develop problem-solving skills to address current societal challenges and demonstrates the potential impact that students of all backgrounds can have as members of the machine intelligence community.

Longitudinal outreach programmes

Outreach programmes with leadership or teaching roles at every stage offer the potential to provide a sustainable infrastructure for promoting diversity and inclusion. Peer mentorship programme structures offer the opportunity for students to learn from their peers in addition to faculty role models or established leaders in the machine intelligence community. As the student progresses in training, the individual student can be both a mentor (of a younger student) and mentee (of an older student). This structure offers the opportunity to obtain advice from individuals who have recently faced similar challenges and have had recent relatable experiences. Furthermore, this support structure facilitates continuity of mentorship across the different stages of training (for example, elementary, middle and high school, college and beyond). Additionally, pedagogical models such as 'see one, do one, teach one' as part of the core curriculum in the science, technology, engineering and mathematics (STEM) educational system can amplify outreach efforts. For example, college students can provide outreach in high schools to encourage high school students to pursue STEM majors and provide mentorship on college applications; high school students' outreach efforts can provide middle and elementary school students with exposure to STEM. This structure allows individuals to be involved as valued members of the machine intelligence community from an early stage as well as increase self-confidence and sense of belonging in the field of machine intelligence to help overcome the retention and inclusivity challenges that women and minority groups often face as underrepresented members of the community.

Conclusion

As machine intelligence increasingly impacts society, diversity and inclusion are issues of growing concern. Expanding HEART-type classes that are accessible and prerequisite-free in college-level education could attract more individuals from a broader range of backgrounds, some of which might consider pursuing a degree and eventual career in machine intelligence. Nevertheless, offering welcoming courses to entering undergraduates is only one small part of the necessary steps to promote active and long-term participation from individuals of all backgrounds. Moving forward, it will also be essential to engage students in primary and secondary education and facilitate a mentorship structure to promote a sustainable infrastructure for diversity and inclusion to ensure the development and implementation of safe, equitable and impactful applications of machine intelligence. ❐

Shannon Wongvibulsin[1,2,3]

[1]Johns Hopkins University School of Medicine, Johns Hopkins University, Baltimore, MD, USA. [2]Department of Biomedical Engineering, Johns Hopkins University School of Medicine, Baltimore, MD, USA. [3]Johns Hopkins University Medical Scientist Training Program, Baltimore, MD, USA.
e-mail: swongvi1@jhmi.edu

Published online: 28 January 2019
https://doi.org/10.1038/s42256-019-0021-8

References

1. Horvitz, E. *Science* **357**, 7 (2017).
2. Whitney, T. & Taylor, V. *Computer* **51**, 24–31 (2018).
3. Integrated Postsecondary Education Data System. *NCES* https://nces.ed.gov/ipeds (2019).
4. *Diversity in Tech* (US Equal Employment Opportunity Commission, 2016).
5. Ferrini-Mundy, J. *Science* **340**, 278 (2013).
6. Miller, F. A., Katz, J. H. & Gans, R. *OD Practictioner* **50**, 6–12 (2018).
7. Hopkins Engineering Applications & Research Tutorials (HEART). *Johns Hopkins University* https://engineering.jhu.edu/heart/ (2019).
8. Tuthill, J. *Lancet* **371**, 1906 (2008).
9. Velegol, S. B., Zappe, S. E. & Mahoney, E. *Adv. Eng. Educ.* **4**, n3 (2015).
10. Freeman, S. et al. *Proc. Natl Acad. Sci. USA* **111**, 8410–8415 (2014).

Acknowledgements

This work was supported by the Johns Hopkins School of Medicine Medical Scientist Training Program (National Institutes of Health: Institutional Predoctoral Training Grant — T32), National Institutes of Health: Ruth L. Kirschstein Individual Predoctoral NRSA for MD/PhD: F30 Training Grant, the Johns Hopkins Individualized Health (inHealth) Initiative, and the Hopkins Engineering Applications & Research Tutorials (HEART) programme.

Competing interests

The author declares no competing interests.

Responsible AI for conservation

Artificial intelligence (AI) promises to be an invaluable tool for nature conservation, but its misuse could have severe real-world consequences for people and wildlife. Conservation scientists discuss how improved metrics and ethical oversight can mitigate these risks.

Oliver R. Wearn, Robin Freeman and David M. P. Jacoby

Machine learning (ML) is revolutionizing efforts to conserve nature. ML algorithms are being applied to predict the extinction risk of thousands of species[1], assess the global footprint of fisheries[2], and identify animals and humans in wildlife sensor data recorded in the field[3]. These efforts have recently been given a huge boost with support from the commercial sector. New initiatives, such as Microsoft's AI for Earth[4] and Google's AI for Social Good, are bringing new resources and new ML tools to bear on some of the biggest challenges in conservation. In parallel to this, the open data revolution means that global-scale, conservation-relevant datasets can be fed directly to ML algorithms from open data repositories, such as Google Earth Engine for satellite data[5] or Movebank for animal tracking data[6]. Added to these will be Wildlife Insights, a Google-supported platform for hosting and analysing wildlife sensor data that launches this year. With new tools and a proliferation of data comes a bounty of new opportunities, but also new responsibilities.

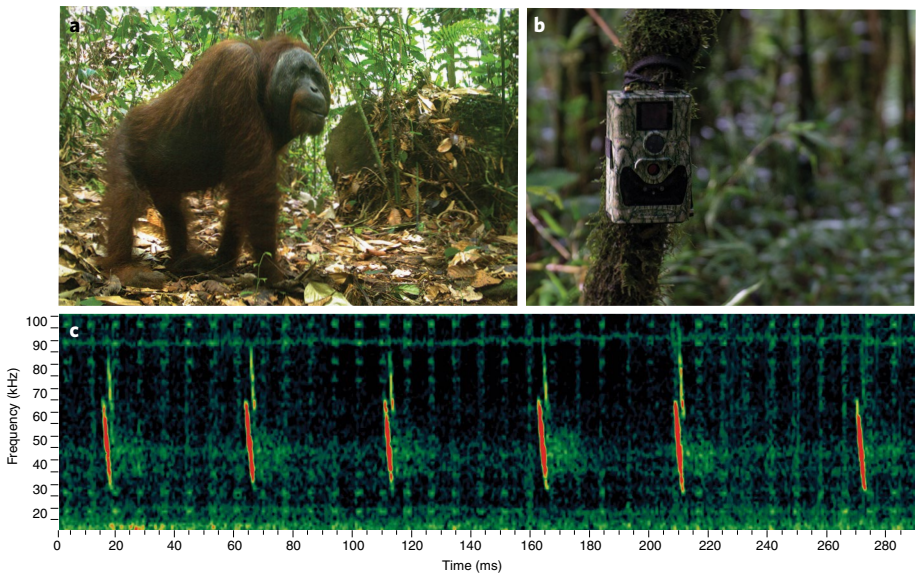


Fig. 1 | Machine learning algorithms on the front line of conservation. ML methods are applied to identify wildlife or people using sensors deployed in the field. **a**,**b**, An image of a critically endangered Bornean orangutan (*Pongo pygmaeus*) (**a**) captured using a camera trap (**b**). **c**, A threatened bat species (*Natalus primus*) detected on a sonogram from in situ acoustic monitoring.

Potential misuse and misinterpretation

The opaque nature of some ML algorithms means that the potential for unintended consequences may be high and this could have real-world consequences for people and wildlife. Understanding, even in an intuitive sense, how neural networks process a given input can currently be very challenging. This has several ramifications that are not yet fully appreciated in the conservation field. First, it can be difficult to identify the implicit assumptions of an algorithm (for example, how much of the contextual background information it is using when identifying species in images), and therefore the potential risks of using it for this purpose. Second, it might be unclear when an algorithm is being asked to make predictions beyond the scope of the training data. Indeed, making sure algorithms 'fail gracefully' is a major research problem[7]. Third, an algorithm might not be easily interrogated as to why it made a particular decision. While these considerations are well-appreciated within the wider AI community, they have been largely absent from recent discussions around the potential benefits of the technology to conservation.

The use of ML to solve conservation problems without consideration of these factors might have severe negative outcomes. A bias against underrepresented classes in a dataset could, for example, mean that a rare species is missed during an environmental impact assessment, leading to the eventual loss of its habitat due to development. Equally, the use of training data with poor coverage of the domain in which predictions are to be made could result in a species being wrongly assessed as extinct on the International Union for Conservation of Nature (IUCN) Red List, meaning that conservation resources are diverted elsewhere (the 'Romeo error'[8]). A misclassification error could also wrongly flag local community members as poachers, raising potentially severe legal and safety concerns. Under any of these circumstances the lack of interpretability and accountability we have for an algorithm's decision would be laid bare[9]. As such, perverse outcomes of applying ML in conservation have the potential to waste scarce resources, increase the costs of conservation to local communities, and erode trust in science-led approaches to environmental problems.

Better metrics needed

That is not to say that these outcomes are unavoidable. One area of research that conservation might benefit from is the development — by ML researchers and conservationists working together — of better metrics for assessing the usefulness of any given algorithm for actually doing conservation. Currently, much of the focus is on standard predictive accuracy metrics. While useful for assessing performance in a controlled 'laboratory' setting (using a single or very limited number of datasets, sometimes with pre-treatment of the data such as cropping), accuracy metrics may

prove inadequate once the algorithms are released to make automated decisions in the wild. Here, extraneous factors may play a much more important role in the output than anticipated. For example, there has been a recent proliferation of studies presenting deep neural network approaches to classifying imagery taken by autonomous cameras (camera traps; Fig. 1) deployed in the field[10,11], with reported accuracies as high as 98%. Perhaps not helped by media reports, these studies can sometimes be seen as a 'silver bullet' to solve some of the major bottlenecks in wildlife monitoring today. However, simple accuracy metrics are unlikely to provide a good indicator of success when an algorithm is transferred to new datasets — for example, for a new point in time, a new study site or on different species. If the model requires retraining, conservationists are unlikely to have the same abundance of training data as the original study. Perhaps more importantly, accuracy metrics may tell us little about how accurately we will in practice be able to monitor the populations of a suite of species.

Towards responsible use of AI

As well as better metrics, we need better ethical oversight of the use of AI in conservation. We have been here before: a promising new tool is developed, followed by a period of mass uptake among conservationists. We then enter a period of critical appraisal, eventually resulting in the well-considered and effective use of the approach. A good example of this is population viability analysis (PVA), a widely used tool to predict the risk of a species going extinct in the future. PVA first emerged in the 1980s and then saw a surge in use during the 1990s, especially after software became available offering a 'canned' approach. Towards the end of that decade, various researchers began to critique the use of PVAs in conservation, expressing the view that it could act as a 'loaded gun' in the wrong hands, rather than an aid to conservation[12]. Best-practice guidelines were eventually promoted[13,14], nearly two decades after the tool first emerged.

The AI community as a whole is already grappling with the concepts of 'fair AI' and goal alignment — central tenets of the beneficial AI movement — and there is much that conservationists could learn from[15,16]. There is also an emerging consensus within the broader AI community on what responsible and ethical guidelines for AI development look like (for example, the Asilomar AI Principles or the Biosphere Code Manifesto). Such guidelines for conservation could be designed to steer algorithm development in the right direction for humanity and wildlife in ways that are collaborative, maximally beneficial, liberating and yet robust to misuse and corruption (for example, by those involved in the illegal wildlife trade). As conservationists, we are already familiar with ethical oversight of our practices, in particular with respect to the care and husbandry of animals in research (for example, animals kept in captivity for the purposes of ex situ conservation, or animals captured and released for the purposes of research). Robust ethical review processes already exist in many research departments and ethics statements on the use of animals are often provided, or required, in published research. A pragmatic approach may therefore be to encourage the inclusion of a 'responsible AI' statement, which outlines the ethical review process, provides responsible guidance on the limits to an algorithm, and gives a description of the training data involved. This would not only promote greater transparency but would also ensure that researchers are able to demonstrate that they have considered both the generalities and the limitations of their method.

Given the potentially severe social and environmental costs of AI misuse and misinterpretation in conservation, we ask whether we might avoid the pitfalls of the past by building, from the outset, the technical and ethical capacity to harness these new tools responsibly. With this in mind, we have outlined two potential goals for the conservation and AI communities to tackle in the immediate term: the development of metrics to better allow conservationists to assess the usefulness of an algorithm, and the formulation of ethical guidelines for the responsible use of AI in conservation. Importantly, these metrics and guidelines will need to exist in the application domain, not just within the machine intelligence field. Critical to this will be the input of the AI community. Now is the time to bring together conservationists, AI experts and industry, to ensure maximum benefit with minimum harm comes from the application of AI to protect the Earth's most threatened species and habitats. ❐

Oliver R. Wearn *, Robin Freeman and David M. P. Jacoby *

Institute of Zoology, Zoological Society of London, London, UK.

**e-mail: oliver.wearn@ioz.ac.uk; david.jacoby@ioz.ac.uk*

Published online: 11 February 2019
https://doi.org/10.1038/s42256-019-0022-7

References

1. Darrah, S. E., Bland, L. M., Bachman, S. P., Clubbe, C. P. & Trias-Blasi, A. *Divers. Distrib.* **23**, 435–447 (2017).
2. Kroodsma, D. A. et al. *Science* **908**, 904–908 (2018).
3. Mac Aodha, O. et al. *PLoS Comput. Biol.* **14**, e1005995 (2018).
4. Joppa, L. N. *Nature* **552**, 325–328 (2017).
5. Gorelick, N. et al. *Remote Sens. Environ.* **202**, 18–27 (2017).
6. Kranstauber, B. et al. *Environ. Model. Softw.* **26**, 834–835 (2011).
7. Amodei, D. et al. Preprint at https://arxiv.org/abs/1606.06565 (2016).
8. Collar, N. J. *Oryx* **32**, 239–243 (1998).
9. Doshi-Velez, F. & Kim, B. Preprint at https://arxiv.org/abs/1702.08608 (2017).
10. Tabak, M. A. et al. *Methods Ecol. Evol.* https://doi.org/10.1111/2041-210X.13120 (2018).
11. Norouzzadeh, M. S. et al. *Proc. Natl Acad. Sci. USA* **115**, E5716–E5725 (2018).
12. Burgman, M. & Possingham, H. P. in *Genetics, Demography and Viability of Fragmented Populations* (eds Young, A. G. & Clarke, G. M.) 97–112 (Cambridge Univ. Press, Cambridge, 2000).
13. Reed, J. M. et al. *Conserv. Biol.* **16**, 7–19 (2002).
14. Ralls, K., Beissinger, S. R. & Cochrane, J. F. in *Population Viability Analysis* (eds Beissinger, S. R. & McCullough, D. R.) 521–550 (Univ. Chicago Press, Chicago, 2002).
15. Crawford, K. & Calo, R. T. *Nature* **538**, 311–313 (2016).
16. Zou, J. & Schiebinger, L. *Nature* **559**, 324–326 (2018).

Competing interests

The authors declare no competing interests.

PERSPECTIVE
https://doi.org/10.1038/s42256-019-0020-9
nature machine intelligence

Hopes and fears for intelligent machines in fiction and reality

Stephen Cave * and Kanta Dihal *

This paper categorizes some of the fundamental hopes and fears expressed in imaginings of artificial intelligence (AI), based on a survey of 300 fictional and non-fictional works. The categories are structured into four dichotomies, each comprising a hope and a parallel fear, mediated by the notion of control. These are: the hope for much longer lives ('immortality') and the fear of losing one's identity ('inhumanity'); the hope for a life free of work ('ease'), and the fear of becoming redundant ('obsolescence'); the hope that AI can fulfil one's desires ('gratification'), alongside the fear that humans will become redundant to each other ('alienation'); and the hope that AI offers power over others ('dominance'), with the fear that it will turn against us ('uprising'). This Perspective further argues that these perceptions of AI's possibilities, which may be quite detached from the reality of the technology, can influence how it is developed, deployed and regulated.

In the anglophone West, the prospect of intelligent machines is often portrayed in tones of great optimism or equally great pessimism. Regardless of how accurate they are, these portrayals matter, as they create a backdrop of assumptions and expectations against which AI is interpreted and assessed.

There are at least three ways in which these narratives could shape the technology and its impacts. First, they could influence the goals of AI developers. Recently, Dillon and Schaffer-Goddard (manuscript in preparation) have explored this systematically with regard to AI researchers' leisure reading, noting that narratives can "inform and develop research already underway and open up new directions of exploration." Second, narratives could influence public acceptance and uptake of AI systems: for example, a UK parliamentary report[1] notes that those they consulted "wanted a more positive take on AI and its benefits to be conveyed to the public, and feared that developments in AI might be threatened with the kind of public hostility directed towards genetically modified (GM) crops". Third, narratives could influence how AI systems are regulated, as they shape the views of both policymakers and their constituents[2–4].

Given these lines of influence, it is important that narratives about intelligent machines should broadly reflect the actual state and possibilities of the technology. However, the aforementioned parliamentary report emphasized that currently "many of the hopes and the fears presently associated with AI are out of kilter with reality." To understand why this is so, we must first clearly identify and describe those hopes and fears, and second understand why they are prevalent and perpetuated.

This Perspective focusses on the former, with some moves towards the latter. We offer a categorization of what we consider to be the most prevalent hopes and fears for AI, and the dynamics between them. Based on a survey of fictional and non-fictional narratives, we argue that these responses can be structured into four dichotomies, each comprising a hope and a parallel fear. We hope further studies will build on this to examine how and why these narratives are "out of kilter with reality", and the nature of their influence.

Methodology

We set out to categorize strongly prevalent hopes and fears for AI, as expressed in a corpus of popular works, both fiction and speculative non-fiction. We directly examined over 300 works from the twentieth and twenty-first centuries (see Supplementary Information). We also tested whether these categories applied to historical imaginings of intelligent machines as they are described in secondary sources, such as Truitt's *Medieval Robots* and Kang's *Sublime Dreams of Living Machines*[5–9].

Our corpus is not definitive: the range of works engaging with the possibility of intelligent machines is vast and continually growing. But we believe it is large enough to extract key themes. Our primary sources are anglophone Western narratives, plus those narratives that were not originally written in English but are widely available in translation (such as Čapek's *R.U.R.*[10]). We have relied on a variety of indicators in compiling the corpus: for example, for film, we included the IMDB top-35 best-grossing robot films (https://www.imdb.com/list/ls025545074/); for fiction, we have considered older works that are still in print or otherwise deemed classics (such as those that appear in the SF Masterworks series), or more recent works that have won major prizes (such as Leckie's *Ancillary Justice*); for non-fiction, we have looked to relevant works that have attained bestseller status (such as Bostrom's *Superintelligence*, or Kurzweil's *The Age of Spiritual Machines*). We have also included works used as reference points by the media or reports on AI, and those mentioned by the public in a recent survey[11]. Inevitably there will be an element of subjectivity in our selection, and we welcome further suggestions to consider.

In discussing narratives around 'AI', we are conscious that this term was coined only in 1955[12]. Relevant stories both before then and since use a range of other terms. We have therefore not limited ourselves to portrayals that explicitly describe 'AI', but include those that feature machines to which intelligence is ascribed (sometimes, it turns out in the story, falsely). In understanding the term 'intelligence', we follow Margaret Boden's suggestion that it describes "the sorts of things that minds can do"; in particular, she adds, the application of those psychological skills that are used by animals for goal attainment[13]. Under the categories 'artificial' and 'machine' we have similarly cast the net widely, including anything that is built, not born. Common cognate terms describing entities that fall into our corpus are 'robot', 'android', 'automaton' and 'cyborg' (though not all uses of these terms describe relevant entities).

Leverhulme Centre for the Future of Intelligence, University of Cambridge, Cambridge, UK. *e-mail: sjc53@cam.ac.uk; ksd38@cam.ac.uk

In distilling our categorizations for the most prevalent hopes and fears for AI, we have considered various previous attempts. A number of authors have written about fears of intelligent machines: Minsoo Kang uses Ann Radcliffe's distinction between terror and horror[14] to categorize negative responses to automata[8]; Kevin LaGrandeur draws from his sources the theme of rebellion[7]; and Daniel Dinello suggests the dystopian themes found in science fiction are a critique of the utopian visions of the technologists themselves[15]. In their analysis of references to AI in the *New York Times*, Fast and Horvitz list both hopes and fears. They include as hopes: improvements to work, education, transportation, healthcare, decision-making and entertainment, as well as a beneficial singularity event and beneficial merging of humans and AI; and as fears: the loss of control of powerful AI, negative impact on work, military applications, a lack of ethics in AI, a lack of progress in AI, a harmful singularity event, and harmful merging of humans and AI[2]. This is a useful survey, although it does not attempt to discern any underlying system to these responses.

We have attempted to distil the positive and negative projections of AI found in our corpus in a way that highlights what we consider to be the most basic themes. There are of course many alternative ways of categorizing these narratives. But we hope this one is at a high enough level to capture the majority of narratives in just a few categories, while still offering some new insight into their underlying structure.

The four dichotomies

We argue that the affective responses to AI explored in the corpus can be placed within a framework of four dichotomies—that is, four hopes and four parallel fears (pictured). We refer to the four hopes as immortality, ease, gratification and dominance. Each is associated with a range of narratives in which intelligent machines have a transformatively positive impact on the lives of some or all humans. Immortality refers to how AI might be used to radically extend life: we give it primacy as staying alive is the precondition for the pursuit of almost any other goal or wish. Once people have that time, ease refers to how AI might grant them the ability to spend it as they wish by freeing them from work. Gratification refers to how AI can help people use that free time, assisting in whatever constitutes pleasurable activity. Finally, AI technologies can be used for what we call dominance, or power over others, as the means to protect this paradisiacal existence.

In claiming that each of these hopes forms one part of a dichotomy, we argue that the utopian visions that they reflect contain inherent instabilities. The conditions required to fulfil each hope also make a dystopian future possible. Thus, the hope for immortality contains the threat of inhumanity—that is, in the pursuit of an ever longer lifespan, a person risks losing their humanity or identity. Ease threatens to become obsolescence, as the desire to be free from work becomes the fear of being put out of work, replaced by a machine. Gratification carries the risk of alienation when in their desire for (artificially) perfect interactions, humans become alienated from each other. And the pursuit of dominance evokes fears of an uprising, as a people's own AI-enabled power is turned—or turns—on them. The factor of control, we argue, balances the hopes and fears: the extent to which the relevant humans believe they are in control of the AI determines whether they consider the future prospect utopian or dystopian.

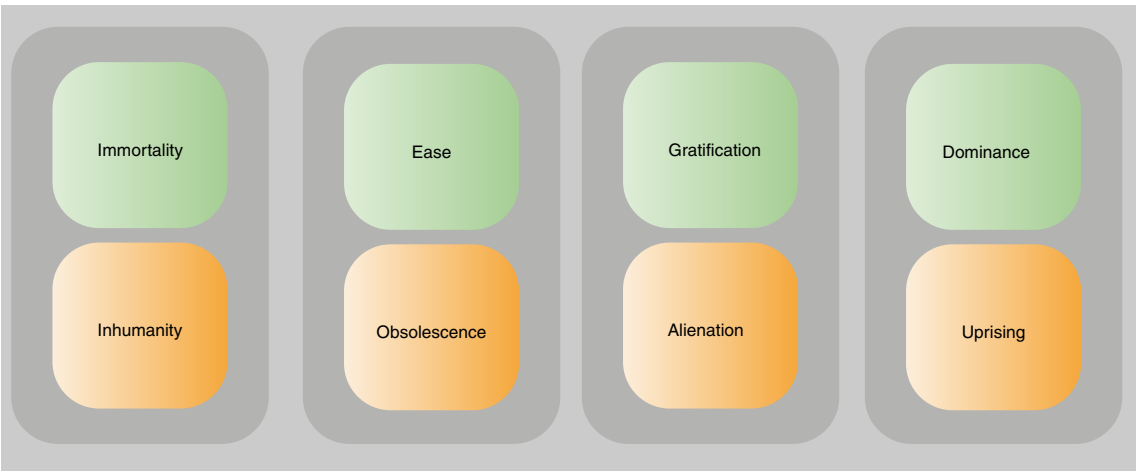
This structure contributes to explaining why responses to AI in the anglophone West are so extreme. The hopeful narratives show the extent to which AI is perceived to be a master tool that can solve problems that have preoccupied humanity throughout history. It represents the apotheosis of the technological dream that humans can use machines to create a paradise on earth[16]. But at the same time, as we explore further below, the idea of creating tools with minds of their own contains (in our imaginings) inherent instabilities. Losing control over such agential machines, or the world they create, is the primary source of the exaggerated fears.

We will now briefly describe these dichotomies and their attendant hopes and fears, illustrating them with examples from our primary sources. Although the examples are used here to illustrate individual hopes or fears, many stories actually address a plurality: the film *Big Hero 6*, for example, expresses all four hopes as well as some of the fears[17].

Immortality versus inhumanity

The pursuit of health and longevity is a basic human drive for the simple reason that it is a precondition for almost anything else one might want. Consequently, humans have always used technology to try to extend lives[18–20]—it is therefore no surprise that one of the great hopes for AI is that it will do just this[3,16,21]. In newspapers, industry reports and elsewhere, AI is portrayed as bringing about a revolution in healthcare, offering better diagnoses, personalized medicine, fewer medical errors and so on[22,23]. Taken to its conclusion, this pursuit of healthier, longer lives becomes the pursuit of indefinitely long lives.

Cave notes two main ways in which people have imagined immortality via AI: transformation and transcendence[24]. In the process of transformation, increasingly sophisticated prophylactics, medicines and prostheses are used to make the body immune to ageing and disease. At its extreme, this is a process of cyborgization, in which the body's unreliable organic parts are replaced by more durable machine parts. By contrast, the process of transcendence



The four dichotomies of hopes and associated fears for AI.

involves total departure from the human body and 'uploading' one's mind onto a machine (which must, by definition, be AI).

These possibilities have been extensively explored in the non-fiction works of the influential technologist Ray Kurzweil, director of engineering at Google, who has written about achieving indefinite lifespans through both transformation and transcendence. In *Fantastic Voyage* he describes what he sees as the increasingly radical medical and other technological interventions that could enable our bodies to keep going indefinitely[25]. In *The Age of Spiritual Machines*[26], he focuses on humans "merging with the world of machine intelligence", resulting in a world in which 'the number of software-based humans vastly exceeds those still using native neuron-cell-based computation.'

But this hope for immortality has its flipside, as Bill Joy put it in his influential 2000 essay 'Why the future doesn't need us'[27]: "on this path our humanity may well be lost." The central concern is whether it is possible for an individual to preserve their identity through the radical metamorphosis that is required to turn an ordinary mortal into something immortal. In one form, this loss of humanity can mean something like loss of human values and emotions. In its more literal form, this fear is that the person hoping for immortality does not really survive at all.

For example, in two episodes, 'Be Right Back' and 'White Christmas', the television series *Black Mirror* critiques different notions of AI-powered survival[28,29]. In the former, a physical replica of a deceased loved one, loaded with his digital data, proves to be a disappointing substitute for the original. In the latter, the character Greta has a digital clone made of herself. The clone is portrayed as having the full personality of the original, but otherwise a much-reduced virtual life. She is informed that her job is to be a digital assistant to the original (who is portrayed as a wholly separate entity—for example, in one scene, asleep while the clone is awake). The narrative makes clear that what the techno-optimists call 'mind-uploading' can also be seen merely as an act of creating copies. Such copies might—in a reduced digital form—outlive us, but they will not literally be us, and so do not offer true survival.

Ease versus obsolescence

Being relieved from the burdens of work is the most ancient hope for intelligent machines. It can be found in the *Iliad*, written around 800 BC, in the form of the golden handmaidens that assist the god Hephaestus[30]. The robot that does our bidding without the complex social and psychological complexities of human servants has been a recurring theme since, both in science fiction and sober predictions of the future. In his essay mentioned above, Joy gives this hope primacy: "the dream of robotics is, first, that intelligent machines can do our work for us, allowing us lives of leisure, restoring us to Eden." Indeed, according to a recent survey exploring public awareness of these dichotomies, this promise that AI will bring a life of luxury and ease is the best known of these hopes for AI[11].

The artificial servant Robby the Robot was the most famous robot of the twentieth century until he was replaced in prominence by the Terminator[31]. In *Forbidden Planet*, Robby is the perfect servant, created by the scientist Morbius as "simply a tool", programmed with "absolute selfless obedience"[32]. Robby is constrained by what Morbius calls 'his basic inhibitions against harming rational beings': he cannot harm a human being, and will overheat and crash when commanded to do so[32]. He is intelligent, but has no will of his own: he will obey humans at all times, with no judgement, being equally eager to protect his master from intruders as to mass-produce bourbon for the spaceship's cook.

But at the same time as people dream of being free from work, they can be terrified of being put out of work. There seems to be a limit to how much leisure time people can tolerate before the fear of becoming entirely obsolete sets in. Work provides people not only with an income, but also with a role in society, status and standing, pride and purpose.

The fear of obsolescence can be divided into two underlying processes. On the one hand, there is the fear of involuntary obsolescence. Jack Williamson's 1947 science fiction novelette 'With Folded Hands' describes a world in which robots protect humans so well, taking away so many jobs that that they consider dangerous or strenuous, that there is nothing left for the human protagonists to do but sit 'with folded hands'[33]. On the other hand, there is the fear of voluntary obsolescence and the long-term effects that that may have on humanity. In *WALL·E*, the human characters seem to be content with their AI-controlled lives, which they spend in immobilizing obesity, in a floating chair, watching screens[34]. For the viewers, this limited life is a dystopian prospect, and they are instead encouraged to identify with the intelligent robots WALL·E and EVE.

Notoriously, a dystopian vision of a future lacking meaningful work was one of the motivations for the 'Unabomber' Ted Kaczynski's violent terrorist campaign against technologists[35].

Gratification versus alienation

Once AI has fulfilled the hopes for longer life and ease, the next goal is to fill all that time with that which brings us pleasure. Just as AI promises to be the perfect servant without the complications of human social hierarchy, so it promises to automate—and thus uncomplicate—the fulfilment of every desire. It could be the perfect companion, for example: always there, always ready to listen, never demanding anything in return. Imaginings of AI are full of such friends: Isaac Asimov's first robot story, 'Robbie' (1939) describes the friendly relationship between a girl and her robot nanny Robbie (not to be confused with Robby from *Forbidden Planet*, although the latter's subservient friendliness was influenced by the former)[36].

In embodied form, AI could also be the perfect lover; fiction tends to present this as a male heterosexual dream. The TV series *Westworld* shows some of the forms such a perfect lover could take: from always-ready prostitutes such as Maeve, to women who have to be courted or subjugated such as Dolores[37]. But even disembodied AI has been portrayed as fulfilling the role of lover: in the film *Her*, Theodore Twombly develops a romantic relationship with his virtual assistant Samantha, who is represented only through her voice[38].

Yet the flipside to the idea of human–AI relationships is that, while some may embrace AI becoming an intimate part of our lives, others may reject the idea of something they perceive to be so unnatural, even monstrous, invading our homes. In robotics, the term 'uncanny valley' describes the revulsion people feel when faced with a replica that is almost human, but not quite[39]. It seems to conjure in us the deep and ancient fear of the doppelgänger, or changeling.

While that fear is based on AI not being human enough, there are also fears around AI being better than humans. If we all have our desires fulfilled by AIs, then we will have become redundant to each other. We might therefore not only become obsolete in the workplace, but even in our own homes and in our own relationships. E. M. Forster anticipated this fear in 'The Machine Stops', in which human interactions are mediated by a machine to such an extent that people never meet each other in person[40]. When the machine stops, they stumble out of their dwellings, disoriented, scared and helpless, and are revolted to come face to face with other humans.

Dominance versus uprising

Finally, the fourth dichotomy concerns power. Once people have long lives and ample free time, and all their desires are fulfilled, they might want to protect this utopia. Indeed, humans have a habit, not just of fighting to protect their favoured way of life, but also of forcing it on others. One major hope for AI is that it can help in retaining or attaining this position of dominance.

Stories of what we now call 'autonomous weapons' are ancient, going back to the bronze giant Talos in the *Argonautica*[41]. In recent times, serious efforts have been underway to make these myths a reality, with significant funding for AI research coming from the military. These themes are also explored in fiction: for example, Iain M. Banks's Culture novels (1987–2012) depict constant clashes between AI-enabled utopians and other life forms resistant to the Culture's imperialism[42]. In other imaginings, AI gives power to the oppressed. In Robert A. Heinlein's *The Moon is a Harsh Mistress* (1966), the inhabitants of the former lunar penal colony make a bid for self-governance aided by a supercomputer that becomes self-aware[43].

The downside of creating autonomous weapons is that such entities might autonomously decide to turn their weapons on their creators. This happens in the very first robot story: Karel Čapek's 1920 play *R.U.R. (Rossum's Universal Robots)*, and has been a persistent theme since[10].

The fears of an uprising are twofold: first, the fear of losing control of AI as a tool—the sorcerer's apprentice scenario. A 2016 White House report[44] highlights the prevalence of this narrative: "In a dystopian vision, super-intelligent machines would exceed the ability of humanity to understand or control. If computers could exert control over many critical systems, the result could be havoc, with humans no longer in control of their destiny at best and extinct at worst. This scenario has long been the subject of science fiction stories, and recent pronouncements from some influential industry leaders have highlighted these fears."

Second, there is the fear that AI systems will turn from mere tools into agents in their own right—what Isaac Asimov called 'the Frankenstein complex'[45]. One of the best-known examples—and beloved of the tabloids—is Arnold Schwarzenegger's T-800 in *The Terminator*[46]. The T-800 is a humanoid robot created by Skynet, an AI that attempts to eliminate humanity as soon as it becomes self-aware: on that day "three billion human lives ended"[47]. The Terminator films symbolize the fears underlying the human hope for dominance by means of AI. Skynet was intended to be an autonomous defence system: it was therefore deliberately given the power and means to destroy other human beings, and at the same time the capacity to develop a will of its own.

As this categorization of the hopes and fears expressed in narratives about AI shows, the idea of creating machines with minds of their own contains inherent instabilities. Of course, there are positive portrayals of intelligent machines, such as the droids R2-D2 and C-3PO in the original Star Wars trilogy. But even here latent dystopian possibilities are visible: the most recent film in that franchise, *Solo: A Star Wars Story*, shows droids standing up for themselves against their human exploiters[48]. The aforementioned survey by Cave et al. shows that public recognition of narratives fits these pairs of hopes and fears: recognition of a positive narrative such as ease equates to recognition of the negative flipside, such as obsolescence[11]. Isaac Asimov was one of the first to recognize this tension, and both critiqued and exploited it in his many robot stories, such as *The Naked Sun*[49]: "One of the reasons the first pioneers left Earth to colonise the rest of the Galaxy was so that they might establish societies in which robots would be allowed to free men of poverty and toil. Even then, there remained a latent suspicion not far below, ready to pop up at any excuse."

Conclusion

In this paper, we have offered a way of approaching the deep-rooted hopes and fears aroused by the prospect of intelligent machines. By structuring them as a series of dichotomies, we hope to have captured both the ambivalence and the strength of feeling (positive and negative) that they invoke. To some researchers in the field, these narratives might seem far removed from the actual power and purpose of the algorithms they are developing. But they nonetheless provide an important context in which their research will be interpreted. As we noted at the start, these narratives around AI stand in a complex causal relationship with the technology itself, both at times inspiring it, and at times trying to reflect it. Yet it is a relationship that also frequently breaks down, in ways that can affect how AI systems will be deployed, adopted and regulated. We hope that understanding the structure and appeal of these framings is a step towards fostering a more balanced discussion of AI's potential.

Received: 26 September 2018; Accepted: 11 January 2019;
Published online: 11 February 2019

References

1. Select Committee on Artificial Intelligence *AI in the UK: Ready, Willing and Able?* (House of Lords, 2018).
2. Fast, E. & Horvitz, E. Long-term trends in the public perception of artificial intelligence. Preprint at https://arxiv.org/abs/1609.04904 (2016).
3. Johnson, D. G. & Verdicchio, M. Reframing AI discourse. *Minds Mach.* **27**, 575–590 (2017).
4. Baum, S. Superintelligence skepticism as a political tool. *Information* **9**, 209 (2018).
5. Mayor, A. *Gods and Robots: The Ancient Quest for Artificial Life*. (Princeton Univ. Press, Princeton, 2018).
6. Truitt, E. R. *Medieval Robots: Mechanism, Magic, Nature, and Art*. (Univ. Pennsylvania Press, Philadelphia, 2015).
7. LaGrandeur, K. *Androids and Intelligent Networks in Early Modern Literature and Culture: Artificial Slaves*. (Routledge, New York, 2013).
8. Kang, M. *Sublime Dreams of Living Machines: The Automaton in the European Imagination*. (Harvard Univ. Press, Cambridge, 2011).
9. Wood, G. *Edison's Eve: A Magical History of the Quest for Mechanical Life*. (Anchor Books, New York, 2002).
10. Čapek, K. *R.U.R.* (Aventinum, Prague, 1920).
11. Cave, S., Coughlan, K. & Dihal, K. 'Scary robots': examining public responses to AI. in *Proc. AIES* http://www.aies-conference.com/wp-content/papers/main/AIES-19_paper_200.pdf (2019).
12. McCarthy, J., Minsky, M. L., Rochester, N. & Shannon, C. E. A proposal for the Dartmouth summer research project on artificial intelligence. *AI Mag.* **27**, 12–14 (Winter, 2006).
13. Boden, M. A. *AI: Its Nature and Future* (Oxford Univ. Press, Oxford, 2016).
14. Radcliffe, A. On the supernatural in poetry. *New Mon. Mag.* **7**, 145–152 (1826).
15. Dinello, D. *Technophobia! Science Fiction Visions of Posthuman Technology* (Univ. Texas Press, Austin, 2005).
16. Noble, D. F. *The Religion of Technology: The Divinity of Man and the Spirit of Invention* (Penguin, New York, 1999).
17. Hall, D. & Williams, C. *Big Hero 6* (Disney, 2014).
18. Gruman, G. J. *A History of Ideas About the Prolongation of Life* (Springer, New York, 2003).
19. Haycock, D. B. *Mortal Coil: A Short History of Living Longer* (Yale Univ. Press, New Haven, 2008).
20. Cave, S. *Immortality: The Quest to Live Forever and How it Drives Civilization* (Crown, New York, 2012).
21. Geraci, R. M. *Apocalyptic AI: Visions of Heaven in Robotics, Artificial Intelligence, and Virtual Reality* (Oxford Univ. Press, Oxford, 2010).
22. Wilson, T. *No Longer Science Fiction, AI and Robotics are Transforming Healthcare* (PwC, 2017).
23. Cockerell, J. Scientists use artificial intelligence to predict how cancers evolve and spread. *The Independent* (2018).
24. Cave, S. in *AI Narratives: A History of Imaginative Thinking about Intelligent Machines* (eds Cave, S., Dihal, K. & Dillon, S.) (Oxford Univ. Press, Oxford, 2020).
25. Kurzweil, R. & Grossman, T. *Fantastic Voyage: Live Long Enough to Live Forever* (Rodale, New York, 2004).
26. Kurzweil, R. *The Age of Spiritual Machines: When Computers Exceed Human Intelligence* (Penguin, New York, 2000).
27. Joy, B. Why the future doesn't need us. *Wired* https://www.wired.com/2000/04/joy-2/ (2000).
28. Harris, O. 'Be Right Back'. *Black Mirror* (Channel 4, 2013).
29. Tibbetts, C. 'White Christmas'. *Black Mirror* (Channel 4, 2014).
30. Cave, S. & Dihal, K. Ancient dreams of intelligent machines: 3,000 years of robots. *Nature* **559**, 473–475 (2018).
31. Telotte, J. P. *Robot Ecology and the Science Fiction Film* (Routledge, New York, 2018).
32. Wilcox, F. M. *Forbidden Planet* (Metro-Goldwyn-Mayer, 1956).

33. Williamson, J. With folded hands. *Astounding Sci. Fict.* **39**, 6–45 (1947).
34. Stanton, A. *WALL·E* (Disney, Pixar, 2008).
35. Kaczynski, T. Industrial society and its future. *The Washington Post* https://www.washingtonpost.com/wp-srv/national/longterm/.../manifesto.text.htm (22 September 1995).
36. Asimov, I. in *The Complete Robot* 164–187 (HarperCollins, London, 1982).
37. Nolan, J. & Joy, L. *Westworld* (HBO, 2016).
38. Jonze, S. *Her* (Sony, 2013).
39. Mori, M. the uncanny valley [from the field]. *IEEE Robot. Autom. Mag.* **19**, 98–100 (2012).
40. Forster, E. M. *The Machine Stops* (The Oxford and Cambridge Review, London, 1909).
41. Apollonius of Rhodes Argonautica Book IV (ed. Hunter, R.) (Cambridge Univ. Press, Cambridge, 2015).
42. Banks, I. M. *Consider Phlebas*. (Macmillan, London, 1987).
43. Heinlein, R. A. *The Moon is a Harsh Mistress*. (Hodder & Stoughton, London, 1966).
44. Preparing for the Future of Artificial Intelligence (Executive Office of the President National Science and Technology Council, 2016).
45. Asimov, I. *The Caves of Steel* (HarperCollins, New York, 1954).
46. Cameron, J. *The Terminator* (Orion, 1984).
47. Cameron, J. *Terminator 2: Judgment Day* (TriStar, 1991).
48. Howard, R. Solo: A Star Wars Story (Lucasfilm, 2018).
49. Asimov, I. *The Naked Sun* (Doubleday, New York, 1957).

Acknowledgements
The authors would like to thank S. Dillon, B. Singler and E. R. Truitt for their helpful comments. This work was funded by a Leverhulme Trust Research Centre Grant awarded to the Leverhulme Centre for the Future of Intelligence.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary information is available for this paper at https://doi.org/10.1038/s42256-019-0020-9.

Reprints and permissions information is available at www.nature.com/reprints.

Correspondence should be addressed to S.C. or K.D.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© Springer Nature Limited 2019

nature machine intelligence

PERSPECTIVE

https://doi.org/10.1038/s42256-019-0024-5

The evolution of citation graphs in artificial intelligence research

Morgan R. Frank[1], Dashun Wang[2,3], Manuel Cebrian[1] and Iyad Rahwan[1,4,5*]

As artificial intelligence (AI) applications see wider deployment, it becomes increasingly important to study the social and societal implications of AI adoption. Therefore, we ask: are AI research and the fields that study social and societal trends keeping pace with each other? Here, we use the Microsoft Academic Graph to study the bibliometric evolution of AI research and its related fields from 1950 to today. Although early AI researchers exhibited strong referencing behaviour towards philosophy, geography and art, modern AI research references mathematics and computer science most strongly. Conversely, other fields, including the social sciences, do not reference AI research in proportion to its growing paper production. Our evidence suggests that the growing preference of AI researchers to publish in topic-specific conferences over academic journals and the increasing presence of industry research pose a challenge to external researchers, as such research is particularly absent from references made by social scientists.

Today's artificial intelligence (AI) has implications for the future of work[1], the stock market[2,3], medicine[4,5], transportation[6,7], the future of warfare[8] and the governance of society[9–11]. On one hand, AI adoption has the positive potential to reduce human error and human bias[12]. As examples, AI systems have balanced judges towards more equitable bail decisions[13], AI systems can assess the safety of neighbourhoods from images[14] and AI systems can improve hiring decisions for board directors while reducing gender bias[15]. On the other hand, recent examples suggest that AI technologies can be deployed without understanding the social biases they possess or the social questions they raise. Consider the recent reports of racial bias in facial recognition software[16,17], the ethical dilemmas of autonomous vehicles[6] and income inequality from computer-driven automation[18–20].

These examples highlight the diversity of today's AI technology and the breadth of its application; an observation leading some to characterize AI as a general-purpose technology[1,21]. As AI becomes increasingly widespread, researchers and policymakers must balance the positive and negative implications of AI adoption. Therefore, we ask: how tightly connected are the social sciences and cutting-edge machine intelligence research?

Here, we employ the Microsoft Academic Graph (MAG) to explore the research connections between AI research and other academic fields through citation patterns. The MAG data offer coverage for both conference proceedings, where AI papers are often published, and academic journals, where other fields prefer to publish. Although early AI research was inspired by the several other fields, including some social sciences, modern AI research is increasingly focused on engineering applications—perhaps due to the increasingly central role of the technology industry. Furthermore, the most central research institutions within the AI research community are increasingly based in industry rather than academia.

Modern AI research

The effort to create human-like intelligence has dramatically advanced in recent decades thanks to improvements in algorithms and computers. However, engineering the entirety of human intelligence has proved difficult. Instead, progress has come from engineering specific human capabilities. While we often use the term AI today in reference to machine learning, the meaning of AI has fluctuated in the past 60 years to variably emphasize vision, language, speech and pattern recognition.

To study the nature of AI research, we use the MAG to identify relevant computer science (CS) subfields from the citations of academic publications from 1950 to 2018. The MAG uses natural language processing (NLP), including keyword analysis, to identify the academic field of each publication according to a hierarchy of academic fields. These data have been particularly useful for studying bibliometric trends in CS[22–25]. Our analysis relies strongly on the MAG's field of study classifications and, thus, our analysis is potentially limited in its accounting of more specific research areas within CS and within AI-related fields. These data enable us to study the paper production and referencing behaviour of different academic fields. For example, CS has risen to the fourth most productive academic field according to annual paper production (see Supplementary Fig. 1) with AI being the most prominent subfield of CS in recent decades[26] (see also Fig. 1d).

To identify the CS subfields that are most relevant to AI research, we construct a citation network using all CS papers published within each decade from 1950 to 2018. We consider CS subfields to represent AI research if they are strongly associated with AI, which is itself a CS subfield, throughout a significant proportion of the time period under analysis. Examples include computer vision, machine learning and pattern recognition. Interestingly, NLP, which is colloquially thought of as a specific problem area in AI[27], is strongly associated with AI research before the mid 1980s, after which NLP becomes more strongly associated with information retrieval and data mining for text-based data (Fig. 1a–c,e). In the remainder, we use papers published in AI, computer vision, machine learning, pattern recognition and NLP to approximate AI research from the 1950s to today.

[1]Media Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA. [2]Kellogg School of Management, Northwestern University, Evanston, IL, USA. [3]Northwestern Institute on Complex Systems, Northwestern University, Evanston, IL, USA. [4]Institute for Data, Systems, and Society, Massachusetts Institute of Technology, Cambridge, MA, USA. [5]Center for Humans and Machines, Max Planck Institute for Human Development, Berlin, Germany. *e-mail: irahwan@mit.edu

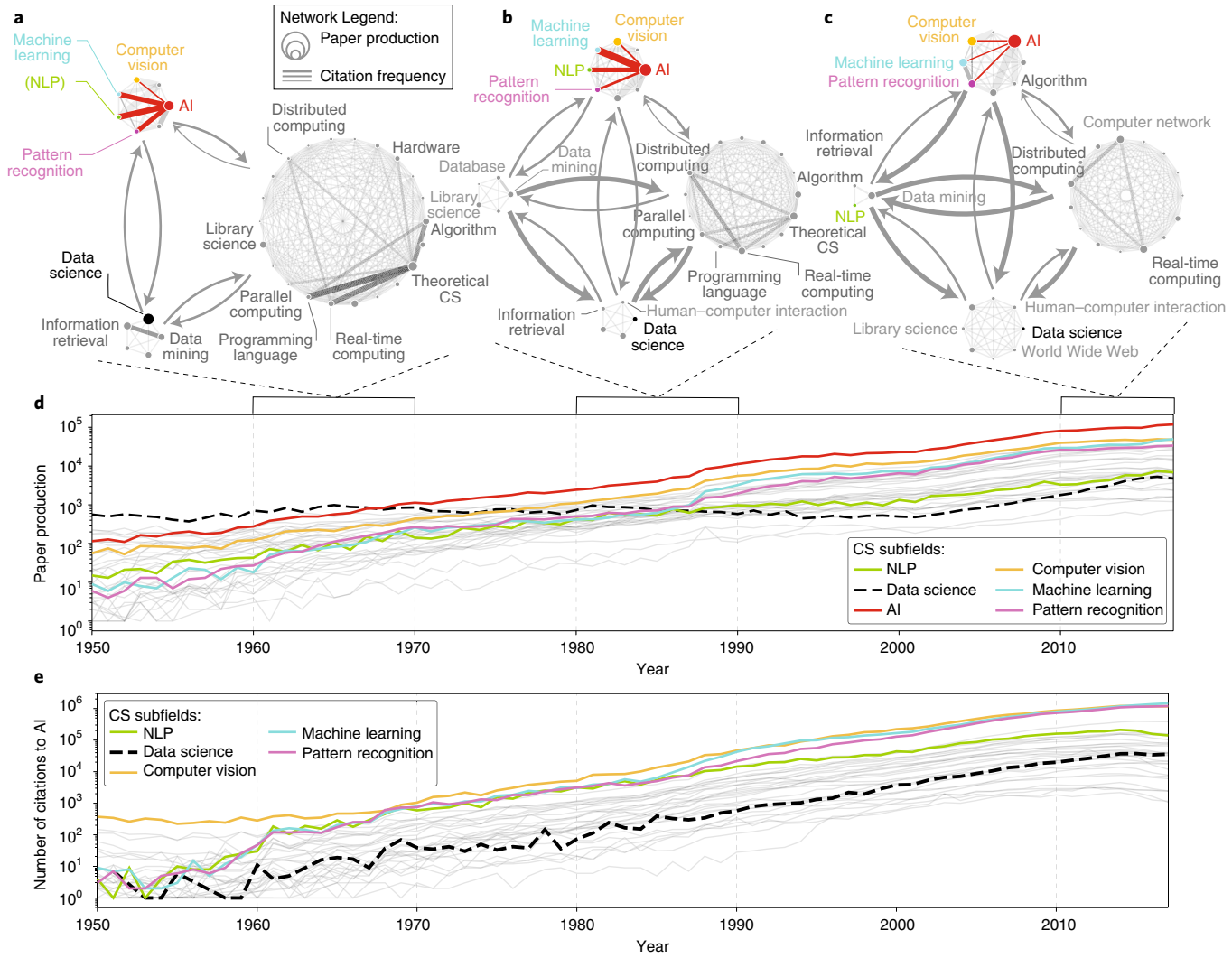


Fig. 1 | Citation patterns among CS subfields identify areas of AI-related research. a–c, We examine the rate of citations between CS subfields based on journal and conference publications from three different decades: the 1960s (**a**), the 1980s (**b**) and the 2010s through 2017 (**c**). For each network, the nodes (circles) correspond to CS subfields according to the MAG data, and the node size corresponds to the number of papers published in each subfield (note, the same paper may belong to multiple subfields). The width of the links connecting the nodes corresponds to the number of references made between papers published in those subfields. After constructing the complete network, we apply topological clustering[45] and report the number of citations made between these clusters using weighted arrows. Networks with labels for each subfield are provided in Supplementary Section 2. **d**, Annual paper production by CS subfield. Subfields related to AI are coloured, as well as data science (black) because of its notable decline in relative paper production. **e**, The annual number of references from papers in each CS subfield to papers in the AI subfield, and vice versa (that is, (subfield → AI) + (subfield ← AI)).

The paper production of CS subfields has varied over the past half-century. For example, data science has gradually diminished in relative paper production and theoretical CS has been replaced by increased focus on real-time and distributed computing. However, AI-related research areas have experienced steadily growing paper production since 1950 and account for the largest share of paper production in CS today (Fig. 1d).

Shaping the study of intelligent machines

Just as early myths and parables emphasized the social and ethical questions around human-created intelligence[28–30], today's intelligent machines provide their own interesting social questions. For example, how responsible are the creators, the manufacturers and the users for the outcomes of an AI system? How should regulators handle distributed agency[11,31]? How will AI technologies reduce instances of human bias? As AI systems become more widespread[1,21], it becomes increasingly important to consider these social, ethical and societal dynamics to completely understand the impact of AI systems[9–11,32,33]. However, the developers of new AI systems are often separate from the scientists who study social questions. Therefore, we might hope to see increasing research interest between these fields of study and AI.

To investigate, we study the association between various academic fields and AI research through the referencing relationship of papers published in each academic field. External fields reference AI research for a number of reasons. Some fields, such as engineering or medicine, reference AI research because they use AI methods for optimization or data analysis. Other fields, such as philosophy, reference AI research because they explore its consequences for society (for example, moral and/or ethical consequences). Similarly, AI researchers reference other fields, such as mathematics or psychology, because AI research incorporates methods and models from these areas. AI researchers may also cite other fields because they use them as application domains to benchmark AI techniques.

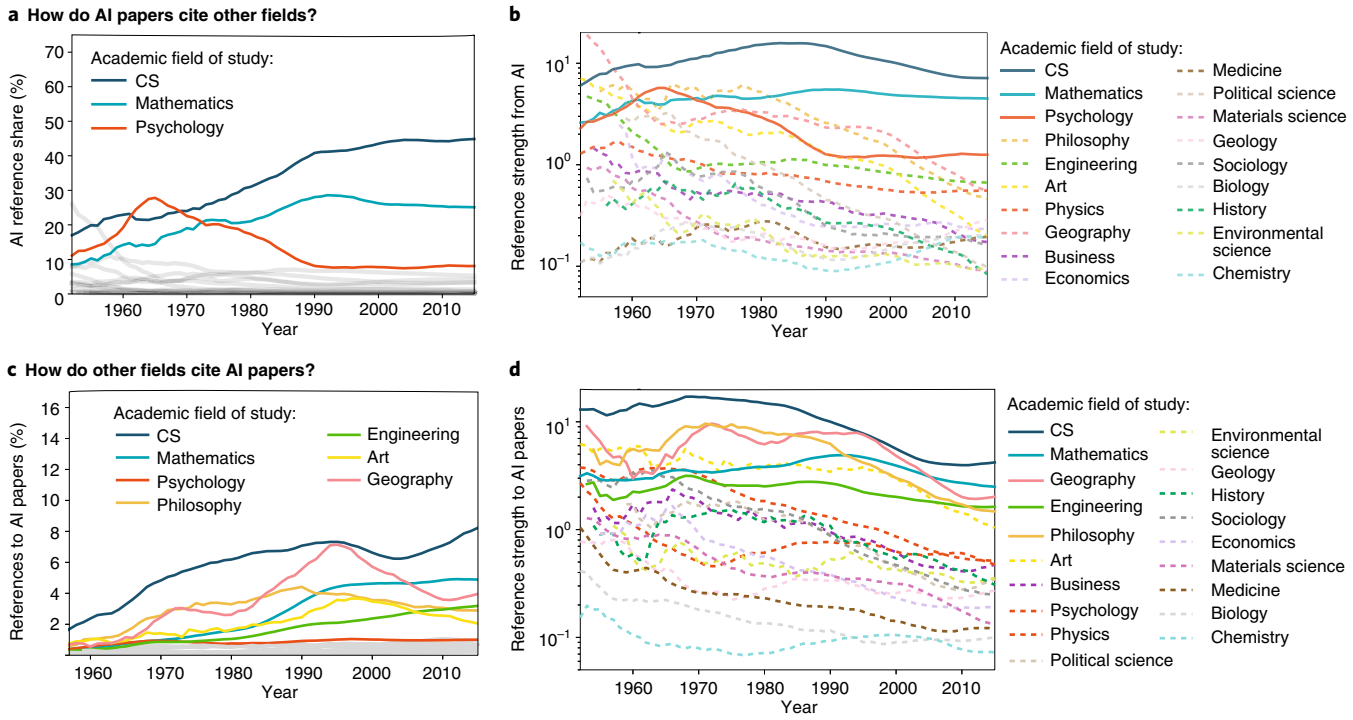


Fig. 2 | The referencing strength between AI and other sciences is declining. a, The share of references made by AI papers in each year to papers published in other academic fields. **b**, The reference strength (see equation (2)) from AI papers to papers published in other academic fields. **c**, The share of references made by each academic field to AI papers in each year. **d**, The reference strength from each other academic field to AI papers in each year. All lines are smoothed using a five-year moving average. In **b**,**d**, dashed lines indicate academic fields exhibiting lower reference strength than would be expected under random referencing behaviour in 2017.

In Fig. 2a,c, we examine the share of references made from AI papers to other fields, and from papers published in other fields to AI. The reference share from academic field A to field B according to

$$\text{share}_{\text{year}}(A,B) = \frac{\#\text{ refs from } A \text{ papers to } B \text{ papers in year}}{\#\text{ refs made by } A \text{ papers in year}} \quad (1)$$

controls for the total paper production of the referencing field over time, and has been used in other bibliometric studies[34]. However, temporal changes in reference share may be explained by paper production in the referenced field; therefore, we consider another measure that also controls for the total paper production in the referenced field as well (Fig. 2b,d). We calculate the reference strength from field A to field B according to

$$\begin{aligned}\text{strength}_{\text{year}}(A,B) &= \frac{\left(\frac{\#\text{ refs from } A \text{ papers to } B \text{ papers in year}}{\#\text{ refs made by } A \text{ papers in year}}\right)}{\left(\frac{\text{no. of } B \text{ papers published from 1950 to year}}{\text{no. of papers published from 1950 to year}}\right)} \quad (2)\\ &= \frac{(\text{reference share from } A \text{ to } B \text{ in year})}{(B\text{'s share of all papers from 1950 to year})}\end{aligned}$$

A reference strength of $\text{strength}_{\text{year}}(A, B) > 1$ indicates that the rate of referencing from field A to field B is greater than would be expected by random referencing behaviour given the number of published papers in field B. Both reference share and reference strength capture the aggregate referencing behaviour between fields of study, but these calculations may obfuscate other dynamics from sub-communities within larger academic fields.

Before 1980, AI research made relatively frequent reference to psychology in addition to CS and mathematics (Fig. 2a). Controlling for the paper production of the referenced fields, we find that early AI's reference strengths towards philosophy, geography and art were comparable to the field's strength of association with mathematics (Fig. 2b) suggesting that early AI research was shaped by a diverse set of fields. However, AI research transitioned to strongly relying on mathematics and CS soon after 1987, which suggests an increasing focus on computational research.

How important is AI research to other academic fields? Unsurprisingly, CS, which includes all of the AI-related subfields in our analysis, steadily increased its share of references made to AI papers throughout the entire period of analysis (Fig. 2c). Surprisingly, mathematics experienced a notable increase in reference share to AI only after 1980. Meanwhile, several fields that are not often cited in today's AI research played an important role in the field's development, but may not have reciprocated this interest. For example, psychology was relatively important to early AI research, but psychology did not reciprocate as strong of an interest at any point from 1990 onwards (that is, strength(psychology, AI) < 1 in recent years). Instead, philosophy, art, engineering and geography have increased their share of references to AI papers up to 1995. On aggregate, when we control for AI paper production over time, we observe decreasing reference strength towards AI from all external academic fields. This suggests that other fields have difficulty keeping track of increasing AI paper production in recent decades (see Supplementary Fig. 3). This result may in part be explained by the increased complexity of AI-related research that is not relevant to the study of other scientific disciplines.

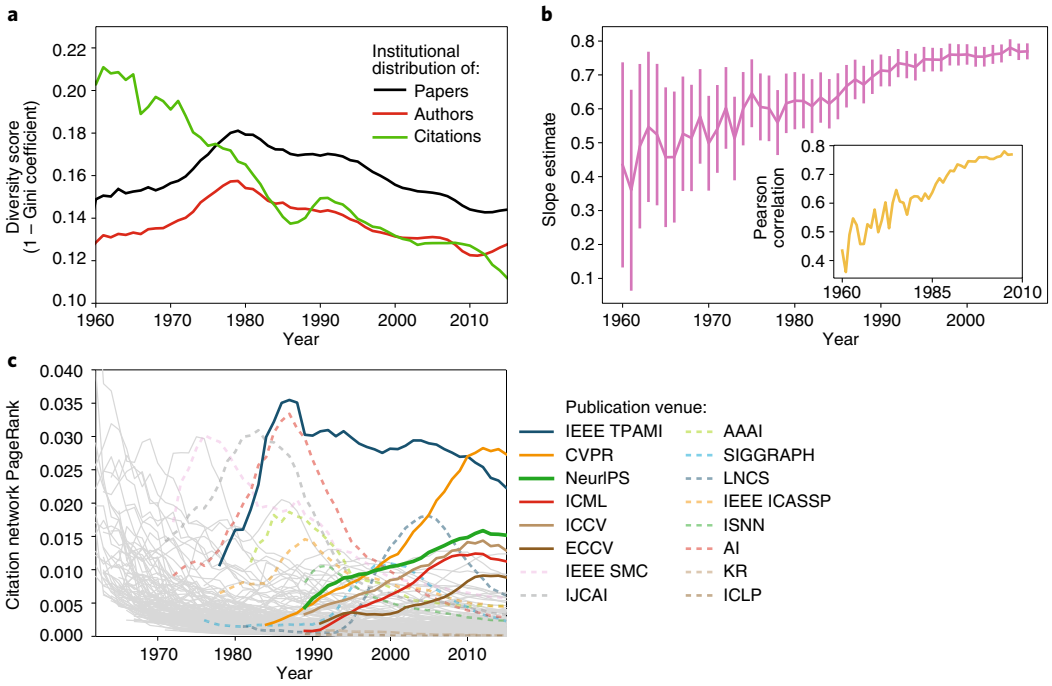


Fig. 3 | AI research is increasingly dominated by only a few research institutions and AI-specific conferences. a, The diversity of the annual distribution of all AI papers (black), AI authors (red) and all citations to AI papers (green) across research institutions according to the Gini coefficient. Example distributions of AI paper production and AI citation share are provided in Supplementary Section 3. **b**, To see whether preferential attachment explains citation dynamics, we include only AI papers with at least one citation and estimate the linear relationship between each research institution's cumulative citation count from 1950 to the institution's citation count in each year (see equation (3)). The model's slope estimation steadily rises throughout the period of analysis to around 0.70 as the model increasingly captures variance in the citation accumulation of institutions according to Pearson correlation (inset). The error bars are 95% confidence intervals for our estimate of the linear model's slope in each year (*m* in equation (3)). **c**, The PageRank of each publication venue for AI papers using the number of references from AI papers published in each venue to papers published in each other venue. The lines of notable publication venues are highlighted with colour. Dashed lines indicate venues whose PageRank has declined during the period of analysis. In all plots, lines are smoothed using a five-year moving average. More recent citation results may change as recent publications continue to accumulate citations. LNCS, Lecture Notes in Computer Science; ICLP, International Conference on Logic Programming; ISNN, International Symposium on Neural Networks; ICCV, International Conference on Computer Vision; ECCV, European Conference on Computer Vision; ICML, International Conference on Machine Learning; CVPR, Conference on Computer Vision and Pattern Recognition; IJCAI, International Joint Conference on AI; KR, Principles of Knowledge Representation and Reasoning.

The consolidation of AI research

How do leading research institutions shape AI research? On one hand, the prestige of an academic university can boost the scientific impact of CS publications[35]. On the other hand, although scientific research is often undertaken at universities, major AI advances have emerged from industry research centres as well. For example, the AI start-up DeepMind received recent attention for their AlphaGo project[36] and Google has been acknowledged as a leader in the development of autonomous vehicles[37–39]. With increased industrial and regulatory involvement, recent work suggests that areas of AI, including deep learning[21], are undergoing a consolidation of research and deployment worldwide. While CS on the whole has become increasingly diverse[40], what can be said about AI research?

If the AI research community is experiencing a consolidation of influence, then what types of citation dynamics might indicate such a phenomenon? We investigate by examining the distribution of AI paper production and the distribution of citations made to AI papers by research institution (see Supplementary Section 3 for visualization of the distributions by decade). Since 1980, the diversity of AI paper production, authorship and citations to AI papers across institutions have decreased by 30% according to the Gini coefficient applied to annual distributions (Fig. 3a). Repeating this analysis for other academic fields, we find that this decreasing diversity is not simply a reflection of aggregate academic trends since most other fields of study actually exhibit increasing diversity over time according to these metrics (see Supplementary Section 5).

This decrease in scientific diversity suggests that notable research 'hubs' may be forming (similar to the industry use of deep learning[21]). This type of hierarchical structure can occur when referencing between institutions is well modelled by preferential attachment[41]. If preferential referencing explains the citation dynamics within AI research, then the proportion of citations gained by a research institution in each year will be proportional to the institution's total accumulation of citations. Figure 3b reports estimates of the slope *m* for the model

$$\log_{10}(\#\text{ of citations}) = m \times \log_{10}(\text{cumulative }\#\text{ of citations}) + b \quad (3)$$

as well as 99% confidence intervals for those slope estimates using linear regression. Both the annual slope estimates and the performance of this model (see inset) rise steadily throughout the period of analysis. Combined, this evidence suggests that preferential referencing may be occurring among AI research institutions.

How have AI publication practices changed over time to enable preferential referencing? To investigate, we calculate the PageRank[42] of each AI publication venue—including both academic journals and conferences—from the references of the AI papers published by each venue in each year (Fig. 3c). Publications venues with larger PageRank are more central to AI research. In the

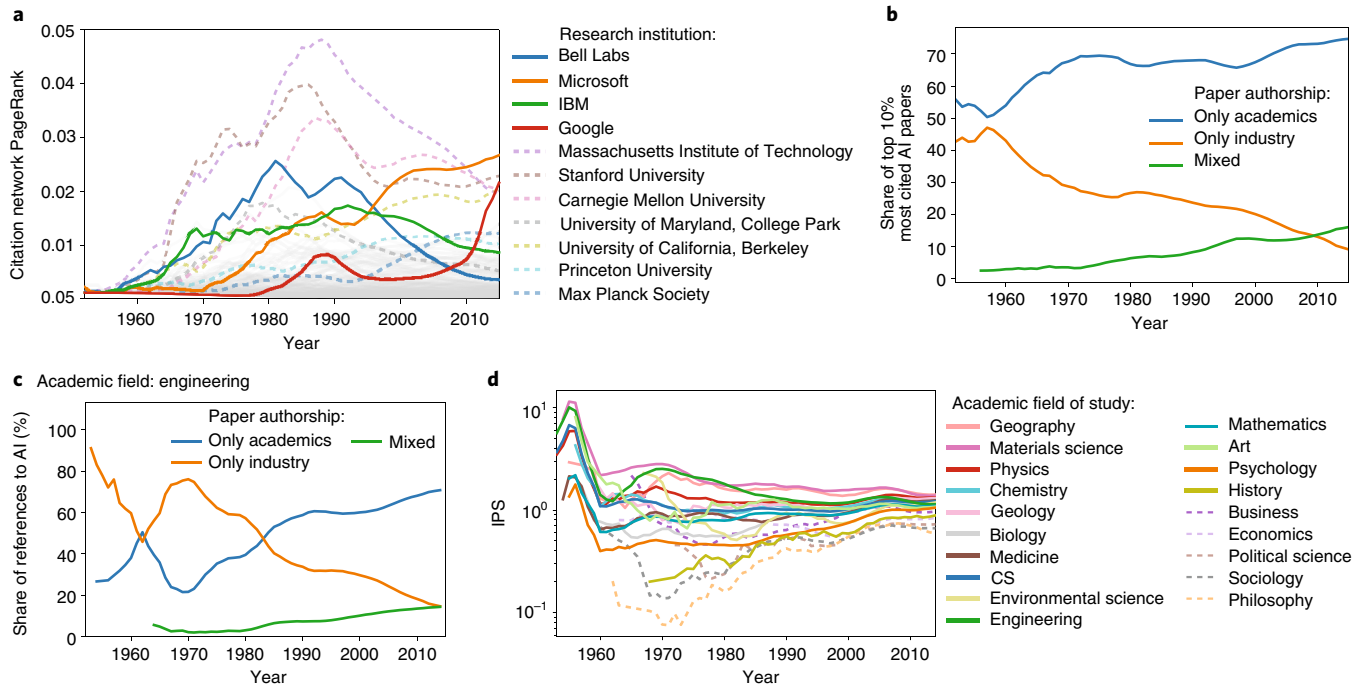


Fig. 4 | Industry is increasingly central to AI research, but industry-authored AI papers are referenced less often by other academic fields. a, The PageRank of each research institution using the number of references from AI papers published by each institution to papers published by each other research institution. The lines of notable research institutions are coloured for visualization. Dashed lines indicate academic institutions while solid lines indicate industry. **b**, The share of the top 10% most cited AI papers published in each year with academic-only, industry-only and mixed authorship. **c**, Similarly to **b**, we examine the referencing behaviour of engineering towards AI papers according to the authorship of the AI papers. Analogous plots for each other academic field are provided in Supplementary Section 4. **d**, Generalizing on **c**, the IPS calculated from each academic field's referencing behaviour towards AI papers (see equation (4)). The solid (dashed) lines indicate fields that reference AI papers with industry-only authorship more (less) than would be expected according to random referencing behaviour. In all plots, lines are smoothed using a five-year moving average.

late 1980s, several specific conferences, including the Conference on Computer Vision and Pattern Recognition, the Conference on Neural Information Processing Systems and the International Conference on Machine Learning, rise in prominence, while more general AI conferences, including the National Conference on Artificial Intelligence and the International Joint Conference on Artificial Intelligence, decline in prominence for AI researchers. Meanwhile, very few academic journals maintain high citation PageRank with the exception of the IEEE Transactions on Pattern Analysis and Machine Intelligence, which remains one of the most central publication venues for AI research.

If preferential referencing is producing research hubs, then which research institutions enjoy a privileged role in the AI research community? To investigate, we calculate the citation PageRank of each institution from the references of the AI papers published by each institution in each year (Fig. 4a). Before 1990, the most prominent research institutions were academic, including the Massachusetts Institute of Technology, Stanford University and Carnegie Mellon University, and included only a few industry-based research institutions, such as Bell Labs and IBM. However, the late 1980s again marks a transition point that reshaped the field. While universities dominate scientific progress across all academic fields[43], industry-based organizations, including Google and Microsoft, are increasingly central to modern AI research, and the PageRank scores of academic institutions are on the decline. Chinese research institutions at today's forefront of AI research are notably absent from Fig. 4a because their rise in prominence is recent in the 65-year time span of our analysis. However, the increasing prominence of Chinese research institutions, as well as other non-US-based institutions, is apparent when focusing on recent years (see Supplementary Section 8).

While academia has remained the largest source of AI papers throughout the entire period of analysis, the increased presence of industry can be seen from the authorship of AI papers over time (Fig. 4b). Out of the 10% of AI papers with the most citations after 10 years, the relative number of papers with industry-only authorship is on the decline. Meanwhile, collaborations between academia and industry are becoming more abundant.

How are other fields of study responding to the increased presence of industry in AI research? As an example, references from engineering showed preference for AI papers with industry-only authorship until the late 1980s, which is contrary to the aggregate trend (Fig. 4c; and see Supplementary Section 3 for similar plots for all academic fields). Similar to reference strength, temporal changes in a field's preference for AI papers with industry authorship (that is, at least one author has an industry affiliation) may result from the abundance of industry-based AI paper production over time. Therefore, we examine each field's industry preference score, which is given for field A by

$$\mathrm{IPS}_{\mathrm{year}}(A) = \frac{(\text{ref. share of } A \text{ to industry AI papers})}{(\text{industry share of AI papers from 1950 to year})} \quad (4)$$

Here, an AI paper has industry authorship if at least one co-author has an affiliation with an industry-based institution. Fields with $\mathrm{IPS}(A) > 1$ exhibit stronger preference for industry AI papers than would be expected under random referencing behaviour towards AI papers. Academic fields that may be interested in the application of AI technology, such as materials science, engineering, chemistry and physics, tend to have greater preference for industry AI papers. However, many of the social sciences and fields that

study social and societal dynamics, such as sociology, economics, philosophy and political science, tend to have lower preference for industry AI papers.

Discussion

Humanity's long-standing quest[28] for AI is rapidly advancing in areas such as vision, speech and pattern recognition. However, as we deploy AI systems, their complete impact includes their social, ethical and societal implications in addition to capabilities and productivity gains. Understanding these implications requires an ongoing dialogue between the researchers who develop new AI technology and the researchers who study social and societal dynamics. Therefore, it is concerning to find a gap between AI research and the research conducted in other fields (Fig. 2).

AI paper production has increased quickly and steadily throughout the past half-century (Fig. 1), which suggests that the remarkable and seemingly sudden progress in AI is rooted in decades of research. Although AI research found as much early inspiration in psychology as CS and mathematics, it has since transitioned towards computational research. Conversely, several other academic fields are dedicating relatively more references to AI research. For example, engineering and mathematics research cite AI papers with increasing relative abundance throughout the period of analysis—making more frequent references to AI papers than would be expected under random referencing behaviour (Fig. 2c,d). However, the decreasing reference strength towards AI papers that we observe on aggregate suggests that most researchers are unable to keep up with the explosion of AI paper production (Fig. 2d). These findings may help explain why recent AI technologies have only recently revealed important (and largely unintentional) social consequences, such as racial bias in facial recognition software[16,17], the ethical dilemmas that have arisen from autonomous vehicles[6] and income inequality in the age of AI[18–20]. If current trends persist, then it may become increasingly difficult for researchers in any academic fields to keep track of cutting-edge AI technology.

The bibliometric gap between AI and other sciences grew with the advent of AI-specific conferences and the increased prominence of industry within AI research. In general, CS conferences can bolster the importance of publications[44] and enable major players to disproportionately influence the entire area of research[40]. Although CS is becoming more diverse on the whole[40], the scientific impact of AI research institutions is becoming less diverse (Fig. 3a). In particular, Microsoft and Google have taken away the central role from universities according to citation PageRank (Fig. 4a), perhaps through preferential referencing of publications within AI (Fig. 3b).

This transition towards industry is challenging for studying the social and societal dynamics of AI technologies. Social science research is less likely to reference AI publications with authors who have industry-based affiliations. Combined with AI's decreasing reference strength towards social sciences, these observations suggest that this gap between research areas will continue to grow. The fields that study social bias, ethical concerns and regulatory challenges may be ignorant of new AI technology—especially when deployed in industry. While our interpretation of these results is speculative, we believe that our observations may highlight an important dynamic within the AI research community that merits further investigation.

Conclusion

The gap between social science and AI research means that researchers and policymakers may be ignorant of the social, ethical and societal implications of new AI systems. While this gap is concerning from a regulatory viewpoint, it also represents an opportunity for researchers. The academic fields that typically inform policymakers on social issues have the opportunity to fill this gap. While our study is a step towards this goal, further work may explicitly quantify the social and societal benefits and consequences of today's AI technology as well as identifying the mechanisms that limit communication between research domains.

Received: 6 September 2018; Accepted: 15 January 2019;
Published online: 11 February 2019

References

1. Brynjolfsson, E. & Mitchell, T. What can machine learning do? Workforce implications. *Science* **358**, 1530–1534 (2017).
2. Kirilenko, A., Kyle, A. S., Samadi, M. & Tuzun, T. The flash crash: high-frequency trading in an electronic market. *J. Finance* **72**, 967–998 (2017).
3. Brogaard, J. et al. *High Frequency Trading and its Impact on Market Quality* Working Paper No. 66 (Northwestern University Kellogg School of Management, 2010).
4. Verghese, A., Shah, N. H. & Harrington, R. A. What this computer needs is a physician: humanism and artificial intelligence. *J. Am. Med. Assoc.* **319**, 19–20 (2018).
5. Hosny, A., Parmar, C., Quackenbush, J., Schwartz, L. H. & Aerts, H. J. Artificial intelligence in radiology. *Nat. Rev. Cancer* **18**, 500–510 (2018).
6. Bonnefon, J.-F., Shariff, A. & Rahwan, I. The social dilemma of autonomous vehicles. *Science* **352**, 1573–1576 (2016).
7. *The Road to Zero: A Vision of Achieving Zero Roadway Deaths by 2050* (National Safety Council and the RAND Corporation, 2018).
8. Russell, S., Hauert, S., Altman, R. & Veloso, M. Ethics of artificial intelligence. *Nature* **521**, 415–416 (2015).
9. Rahwan, I. Society-in-the-loop: programming the algorithmic social contract. *Ethics Inf. Technol.* **20**, 5–14 (2018).
10. Crandall, J. W. et al. Cooperating with machines. *Nat. Commun.* **9**, 233 (2018).
11. Taddeo, M. & Floridi, L. How AI can be a force for good. *Science* **361**, 751–752 (2018).
12. Miller, A. P. Want less-biased decisions? Use algorithms. *Harvard Business Review* https://hbr.org/2018/07/want-less-biased-decisions-use-algorithms (2018).
13. Kleinberg, J., Lakkaraju, H., Leskovec, J., Ludwig, J. & Mullainathan, S. Human decisions and machine predictions. *Q. J. Econ.* **133**, 237–293 (2017).
14. Naik, N., Kominers, S. D., Raskar, R., Glaeser, E. L. & Hidalgo, C. A. Computer vision uncovers predictors of physical urban change. *Proc. Natl Acad. Sci. USA* **114**, 7571–7576 (2017).
15. Erel, S. L. H. T. C., Isil & Weisbach, M. S. Could machine learning help companies select better board directors? *Harvard Business Review* https://hbr.org/2018/04/research-could-machine-learning-help-companies-select-better-board-directors (2018).
16. Buolamwini, J. & Gebru, T. Gender shades: intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency* 77–91 (MLR, 2018).
17. Buolamwini, J. How I'm fighting bias in algorithms. *TED Talks* https://www.ted.com/talks/joy_buolamwini_how_i_m_fighting_bias_in_algorithms (2016).
18. Frank, M. R., Sun, L., Cebrian, M., Youn, H. & Rahwan, I. Small cities face greater impact from automation. *J. R. Soc. Interface* **15**, 20170946 (2018).
19. Frey, C. B. & Osborne, M. A. The future of employment: how susceptible are jobs to computerisation? *Technol. Forecast. Soc. Change* **114**, 254–280 (2017).
20. Acemoglu, D. & Restrepo, P. Robots and jobs: evidence from US labor markets (National Bureau of Economic Research, 2017).
21. Klinger, J., Mateos-Garcia, J. C. & Stathoulopoulos, K. Deep learning, deep change? Mapping the development of the artificial intelligence general purpose technology. Preprint at https://arxiv.org/abs/1808.06355 (2018).
22. Sinha, A. et al. An overview of Microsoft Academic Service (MAS) and applications. In *Proc. 24th International Conference on World Wide Web* 243–246 (ACM, 2015).
23. Effendy, S. & Yap, R. H. Analysing trends in computer science research: a preliminary study using the microsoft academic graph. In *Proceedings of the 26th International Conference on World Wide Web Companion*, 1245–1250 (International World Wide Web Conferences Steering Committee, 2017).
24. Hug, S. E. & Brändle, M. P. The coverage of Microsoft academic: analyzing the publication output of a university. *Scientometrics* **113**, 1551–1571 (2017).
25. Burd, R. et al. GRAM: global research activity map. In *Proc. 2018 International Conference on Advanced Visual Interfaces* 31 (ACM, 2018).
26. Fiala, D. & Tutoky, G. Computer science papers in web of science: a bibliometric analysis. *Publications* **5**, 23 (2017).
27. Russell, S. J. & Norvig, P. *Artificial Intelligence: A Modern Approach* (Pearson Education Limited, London, 2016).
28. McCorduck, P. *Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence* (CRC, Natik, 2009).
29. Kieval, H. J. Pursuing the Golem of Prague: Jewish culture and the invention of a tradition. *Mod. Jud.* **17**, 1–20 (1997).

30. Pollin, B. R. Philosophical and literary sources of Frankenstein. *Comp. Lit.* **17**, 97–108 (1965).
31. Floridi, L. Distributed morality in an information society. *Sci. Eng. Ethics* **19**, 727–743 (2013).
32. Plant, S. *Zeros and ones* (Doubleday Books, 1997).
33. David, A. H. Why are there still so many jobs? The history and future of workplace automation. *J. Econ. Perspect.* **29**, 3–30 (2015).
34. Sinatra, R., Deville, P., Szell, M., Wang, D. & Barabási, A.-L. A century of physics. *Nat. Phys.* **11**, 791 (2015).
35. Morgan, A. C., Economou, D., Way, S. F. & Clauset, A. Prestige drives epistemic inequality in the diffusion of scientific ideas. *EPJ Data Sci.* **7**, 40 (2018).
36. Silver, D. et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484 (2016).
37. Bergholz, R., Timm, K. & Weisser, H. Autonomous vehicle arrangement and method for controlling an autonomous vehicle. US patent 6,151,539 (2000).
38. Pilutti, T. E., Rupp, M. Y., Trombley, R. A., Waldis, A. & Yopp, W. T. Autonomous vehicle identification. US patent 9,552,735 (2017).
39. Herbach, J. S. & Fairfield, N. Detecting that an autonomous vehicle is in a stuck condition. US patent 8,996,224 (2015).
40. Pham, M. C., Klamma, R. & Jarke, M. Development of computer science disciplines: a social network analysis approach. *Soc. Netw. Anal. Min.* **1**, 321–340 (2011).
41. Barabási, A.-L. & Albert, R. Emergence of scaling in random networks. *Science* **286**, 509–512 (1999).
42. Page, L., Brin, S., Motwani, R. & Winograd, T. *The PageRank Citation Ranking: Bringing Order to the Web* (Stanford InfoLab, 1999).
43. Larivière, V., Macaluso, B., Mongeon, P., Siler, K. & Sugimoto, C. R. Vanishing industries and the rising monopoly of universities in published research. *PLoS ONE* **13**, 1–10 (2018).
44. Freyne, J., Coyle, L., Smyth, B. & Cunningham, P. Relative status of journal and conference publications in computer science. *Commun. ACM* **53**, 124–132 (2010).
45. Newman, M. E. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* **74**, 036104 (2006).

Acknowledgements

The authors would like to thank E. Moro and Z. Epstein for their comments.

Author contributions

M.R.F. and D.W. processed data and produced figures. All authors wrote the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s42256-019-0024-5.

Reprints and permissions information is available at www.nature.com/reprints.

Correspondence should be addressed to I.R.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© Springer Nature Limited 2019

nature machine intelligence

ARTICLES

https://doi.org/10.1038/s42256-018-0015-y

Training deep neural networks for binary communication with the Whetstone method

William Severa*, Craig M. Vineyard, Ryan Dellana, Stephen J. Verzi and James B. Aimone*

The computational cost of deep neural networks presents challenges to broadly deploying these algorithms. Low-power and embedded neuromorphic processors offer potentially dramatic performance-per-watt improvements over traditional processors. However, programming these brain-inspired platforms generally requires platform-specific expertise. It is therefore difficult to achieve state-of-the-art performance on these platforms, limiting their applicability. Here we present Whetstone, a method to bridge this gap by converting deep neural networks to have discrete, binary communication. During the training process, the activation function at each layer is progressively sharpened towards a threshold activation, with limited loss in performance. Whetstone sharpened networks do not require a rate code or other spike-based coding scheme, thus producing networks comparable in timing and size to conventional artificial neural networks. We demonstrate Whetstone on a number of architectures and tasks such as image classification, autoencoders and semantic segmentation. Whetstone is currently implemented within the Keras wrapper for TensorFlow and is widely extendable.

Artificial neural network (ANN) algorithms, specifically deep convolutional networks (DCNs) and other deep learning methods, have become the state-of-the-art techniques for a number of machine learning applications[1–3]. While deep learning models can be expensive both in time and energy to operate and even more expensive to train, their exceptional accuracy on fundamental analytics tasks such as image classification and audio processing has made their use essential in many domains.

Some applications can rely on remote servers to perform deep learning calculations; however, for many applications such as onboard processing in autonomous platforms like self-driving cars, drones and smart phones, the resource requirements of running large ANNs may still prove to be prohibitive[4,5]. Large ANNs with many parameters require a significant storage capacity that is not always available, and data movement energy costs are greater than that of performing the computation, making large ANNs intractable[6]. Additionally, onboard processing capabilities are often limited to meet energy budget requirements, further complicating the challenge. Other factors such as privacy and data sharing also provide a motivation for performing computation locally rather than on a remote server.

The development of specialized hardware to enable more efficient ANN calculations seeks to facilitate moving ANNs into resource-constrained environments, particularly for trained algorithms that simply require the deployment of an inference-ready network. A common approach today is to optimize key computational kernels of ANNs in application-specific integrated circuits (ASICs)[7–10]. However, while these ASICs can provide substantial acceleration, their power costs are still too high for some embedded applications and often lack flexibility for implementing alternative ANN architectures.

Brain-inspired neuromorphic hardware presents an alternative to conventional ASIC accelerators, and has been shown to be capable of running ANNs with potentially orders-of-magnitude lower power consumption (that is, performance-per-watt). The landscape of neuromorphic hardware is rapidly evolving[11–16]; however, increasingly these approaches leverage spiking to achieve substantial energy savings. Neuromorphic spiking, which emulates all-or-none action potentials in biological neurons, limits communication in hardware only to discrete events. For spiking neuromorphic hardware to be useful, however, it is necessary to convert an ANN, for which communication between artificial neurons can be high-precision, to a spiking neural network (SNN). Supplementary Note 1 provides further details of spiking and ANN acceleration.

The conversion of ANNs to SNNs—whatever their form—is non-trivial, as ANNs depend on gradient-based backpropagation training algorithms, which require high-precision communication, and the resultant networks effectively assume the persistence of that precision. While there are methods for converting existing ANNs to SNNs, these transformations often require using representations that diminish the benefits of spiking. Here, we describe a new approach to training SNNs, where the ANN training is to not only learn the task, but to produce a SNN in the process. Specifically, if the training procedure can include the eventual objective of low-precision communication between nodes, the training process of a SNN can be nearly as effective as a comparable ANN. This method, which we term Whetstone (Fig. 1) inspired by the tool to sharpen a dull knife, is intentionally agnostic to both the type of ANN being trained and the targeted neuromorphic hardware. Rather, the intent is to provide a straightforward interface for machine learning researchers to leverage the powerful capabilities of low-power neuromorphic hardware on a wide range of deep learning applications (see section 'Implementation and software package details').

Results

Whetstone method converts general ANNs to SNNs. The Whetstone algorithm operates by incorporating the conversion into binary activations directly into the training process. Because most techniques to train ANNs rely on stochastic gradient descent methods, it is necessary that the activations of neurons be differentiable during the training process. However, as networks become trained, the training process is able to incorporate additional constraints, such as targeting discrete communication between nodes. With this shift of the optimization target in

Center for Computing Research, Sandia National Laboratories, Albuquerque, NM, USA. *e-mail: wmsever@sandia.gov; jbaimon@sandia.gov

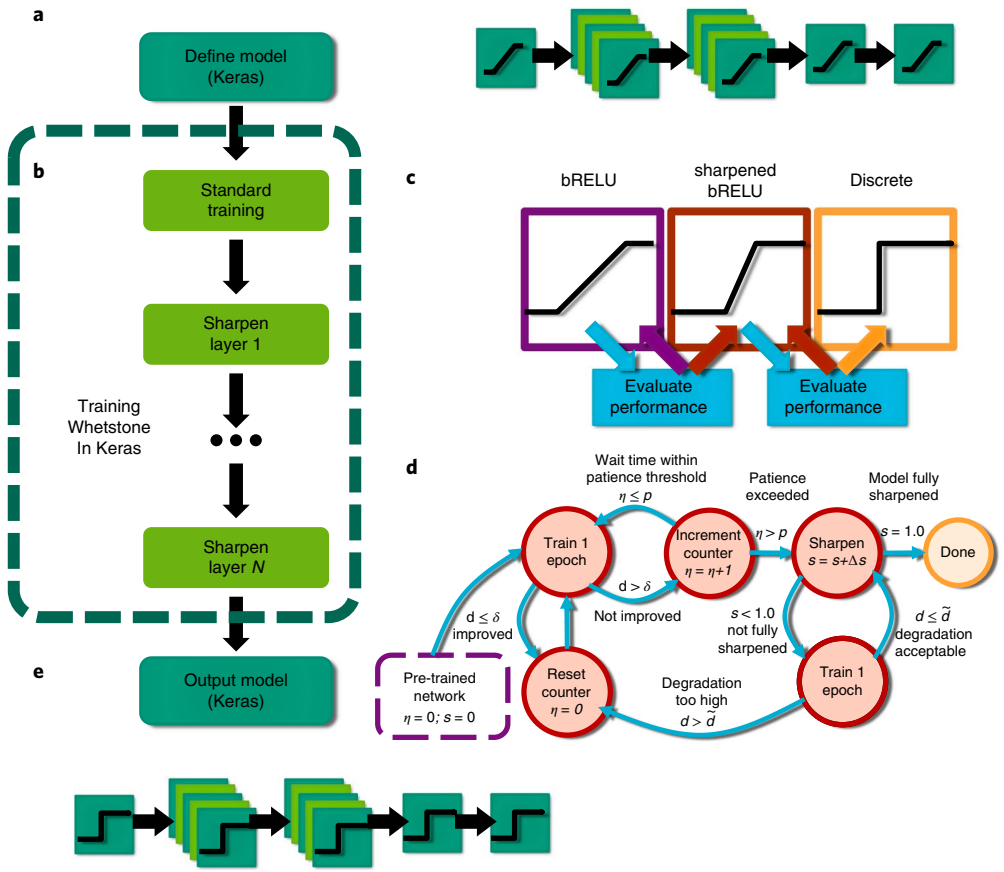


Fig. 1 | Overview of the Whetstone process. Whetstone is a process for training binary, threshold-activation SNNs using existing deep learning methods. **a**, The first step of Whetstone is to define a conventional network architecture within Keras. **b**, Whetstone first performs normal network training until performance begins to plateau. Whetstone then sharpens each layer one at a time, beginning with the input layer. **c**, Sharpening is implemented within each layer by adjusting neurons' bRELU activation functions while continuing training to have a progressively steeper slope. This sharpening continues within each layer until the activation is a binary threshold. **d**, The sharpening process is automated using an adaptive sharpening schedule. At the end of each epoch, the percentage increase in loss d (degradation) determines whether to sharpen during the next epoch, or pause so the performance can stabilize. Any increase in loss d above the critical threshold $\tilde{d}$ causes a transition to the waiting cycle. A transition from waiting back to sharpening occurs when there have been p consecutive epochs without significant improvement per the η counter and δ threshold. This process continues until the model reaches a sharpness s of 1.0 (100%). **e**, Output networks are conventional neural networks that require one time-step per layer and spiking-compatible discrete activation functions.

mind, Whetstone gradually pushes the network towards discrete 'spike' activations by shifting the gradient of bounded rectified linear unit (bRELU) activation functions incrementally towards a discrete perceptron-like step function, then fine-tuning the network to account for any loss as a result of that conversion (see section 'Converging to spiking activations'). By gradually 'sharpening' neurons' activations layer by layer, the network can slowly approach an SNN that has minimal loss from the full-precision case (see section 'Sharpening schedule').

The outputs of Whetstone are shown in Fig. 2 for the training of an example network. The goal of Whetstone training is to produce an ANN with discrete activations (either 1 or 0) for all communication between neurons. However, because networks are not typically trained with this goal incorporated into their optimization, the immediate conversion of activations into a binary 1 or 0 results in a substantial drop in accuracy. However, as Whetstone gradually converts networks to SNNs through the incremental sharpening of each layer, one by one (Fig. 2b), the performance of the sharpened Whetstone networks only experiences minor impairment compared to the standard trained network. Furthermore, once the early layers are discretized through Whetstone, the loss introduced by forcing networks to have discrete communication is minimized.

Description of baseline Whetstone accuracy. We examined the performance of Whetstone on image classification within four different datasets: MNIST, Fashion MNIST, CIFAR-10 and CIFAR-100, and several different network configurations. We then plotted the performance of networks across a wide hyperparameter space (Fig. 3). Hyperparameter optimization for ANNs is a complex, open problem with several popular solutions[17,18]. These methods are generally compatible with our approach as they do not depend on the specific activation functions, and so in a production environment the hyperparameters of Whetstone networks can be optimized by industry-standard approaches. We hope that, in providing this wide scope of networks and performance levels, we can gain insight into Whetstone's performance across applications and hyperparameters rather than only present the hand-tuned top-performing networks in Table 1. For most experiments, equivalent spiking networks were somewhat more brittle, leading to modest overall performance losses, as shown in Fig. 3. This is not surprising, given that the spike representations mean less precision in the communication between layers, and the relatively small differences suggest that small specializations to networks for spiking may mitigate much of this loss.

Within the configurations tested, there was not a common trend to suggest that a coarse consideration would improve sharpened

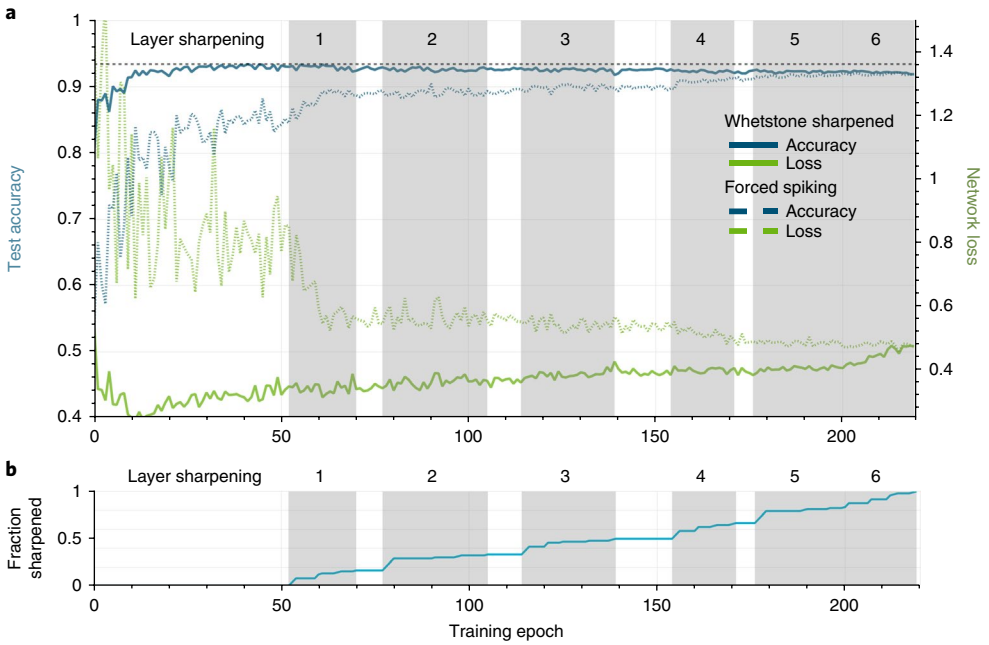


Fig. 2 | Training a single network through the Whetstone process. a, A six-layer ANN was trained on the MNIST dataset, reaching a steady-state accuracy (dashed black line) at around 50 epochs. Within the shaded regions, the Whetstone process is gradually pushing neurons, layer by layer, to have progressively sharper activation functions until they are essentially discrete. The blue dotted line shows the accuracy of the network if the network is forced to fully discrete activations at that time, and the blue solid line shows the accuracy of the Whetstone network. The observed decrease in test accuracy through the sharpening process is due primarily to the precision drop of communication between layers. Green dotted and dashed lines correspond to the network loss for the forced spiking and Whetstone sharpened networks, respectively. **b**, Sharpening of layers of the network shown in **a**. Each shaded region shows the relative sharpening of the layer over a number of epochs. A layer that is fully blue during that sharpening phase is discrete, and the height of blue within each shaded region reflects how sharp that layer is through the sharpening phase.

performance. For instance, deeper networks leveraging convolutional layers performed better for both non-sharpened networks and their sharpened equivalents, as one would expect. For MNIST, the largest network had roughly equivalent sharpened and non-sharpened performance; however, this is not the case for other datasets. Likewise, we observed some runs where larger kernel sizes were helpful for the sharpened networks, although this was not universally the case.

While the modest penalty for sharpening that we observed may be permissible for some applications where the energy savings of a spiking representation would outweigh the accuracy hit, we sought to further improve Whetstone performance by examining a few aspects of network representation and training that may uniquely impact the spike-conversion process. We thus examined strategies for output encoding as we observed that a number of Whetstone runs occasionally suffered from whole classes failing to be classified, and we further examined the effects of optimizers, batch normalization and weight quantization on Whetstone performance.

***N*-hot output encoding and addressing 'dead' nodes.** One challenge of the bRELUs used in Whetstone is that nodes may stop responding to any inputs, effectively rendering them 'dead' nodes. This is particularly an issue at the output layer, where if a node ceases to respond a class may no longer be represented at all. These encoding failures have been noted for conventional networks, both utilizing sigmoids and RELUs, particularly in the context of transfer learning or other applications where a subset of classes cease to be trainable. However, because our sharpening process can move a node's encoding from a differentiable to a non-differentiable regime, it is likely that the problem is exacerbated here.

The results reported in Fig. 3 used a conservative encoding scheme to avoid any loss due to 'dead' nodes in the output layer, in which each output class is represented by redundant neurons that independently determine if that class is activated (see section 'Output encodings'). Immediately noticeable is that, as predicted from the aforementioned observation of dead nodes, 1-hot encoding is unreliable and insufficient for our purposes. This unreliability, however, is essentially eliminated by even modest 4-way redundancy, and the 4-hot encoding appears to offer an advantage beyond just size, with that architecture showing high overall performance on MNIST (99.24%), in this study surpassing the performance of equivalent networks with larger output layers. See Supplementary Note 2 for more discussion.

To better quantify this concentration of activity, we measured the Gini coefficient of all the neurons within a network as Whetstone sharpened it (Fig. 4b). The Gini coefficient ranges from 0 to 1 and is commonly used as a metric of inequality in economics, and in the context of ANNs—particularly those with bounded activation functions such as the bRELU used here—we can use it to measure the relative efficiency of a coding scheme across the neurons within a network. A high Gini coefficient indicates that a small subset of neurons is used to encode most information, whereas a low value indicates that the full population of neurons is used equivalently across all information.

As seen in Fig. 4b, the sharpening of activations reliably increased the Gini coefficient of our networks by roughly 10%. This is consistent with the above finding that network sharpening leads to dead nodes that are not used in the network. Interestingly, the sharpening process has the greatest effect on the distribution of nodes during the sharpening of the early layers, as shown in the rapid rise in Gini at epoch 300 (Fig. 4b). As shown in Fig. 4c, in the first layer (layer 0) there is a much greater skew of values to either always be active or inactive after sharpening, whereas the distribution of intermediate layers' average activations do not change considerably through the

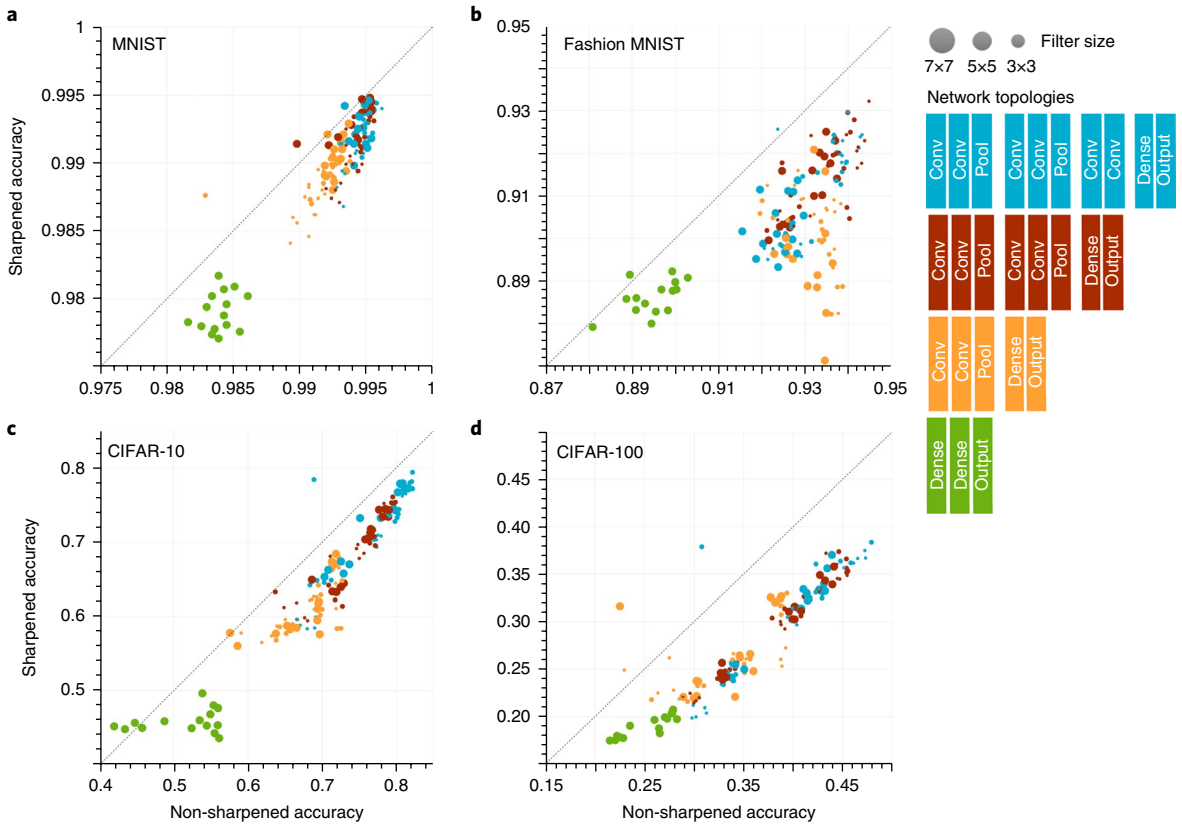


Fig. 3 | How Whetstone training influences the performance of different network topologies and tasks. a–d, Whetstone was tested on several network sizes and topologies against the MNIST (**a**), Fashion MNIST (**b**), CIFAR-10 (**c**) and CIFAR-100 (**d**) datasets. The distance below (or above) the diagonal indicates the penalty (or improvement) that the binarized network exhibited. Not surprisingly, larger networks (yellow, red, blue) had higher classification accuracy than the smallest network (green) for both sharpened and non-sharpened networks. For most cases, sharpened accuracy was moderately lower than the non-sharpened accuracy, indicating a small penalty in classification accuracy due to the reduced precision.

Table 1 | Best reported result of neuromorphic-compatible deep learning algorithms on classification

Algorithm/ author	Method	MNIST accuracy	CIFAR-10 accuracy
Whetstone	Binary communication (VGG-like)	0.9953	0.8467
Whetstone	Binary communication (10-net ensemble)	0.9953	0.8801
Eliasmith et al.[20]	Transfer of trained ANN to spiking leaky integrate-and-fire	0.9912	0.8354
Energy-efficient deep neuromorphic networks[21,22]	TrueNorth compatible convolutional networks	0.9942	0.8932
Rueckauer et al.[23]	Spiking equivalents of common convolutional neural network architecture constructs	0.9944	0.9085
BinaryNet[33]	Binary weights and activations	0.9904	0.8985

Whetstone process. Supplementary Fig. 1 shows how the preferred responses on convolutional filters change as a result of sharpening.

Whetstone is robust to optimizer choice, normalization and weight quantization. A benefit of implementing Whetstone within Keras is that it provides access to numerous training optimizers and normalization schemes that have proven useful for different ANN applications. Because Whetstone is changing the activation function during training, we sought to characterize the interaction of these optimizers with our sharpening process. As shown in Supplementary Fig. 2, some optimizers—most notably Adam—suffered from large performance variance, whereas other optimizers such as adadelta and adamax were reliable under a variety of hyperparameters such as learning rate. With regard to normalization, we found that batch normalization improved the stability of accuracy by about 40% (Supplementary Fig. 2b,c). Implications on using batch normalization on neuromorphic hardware are provided in Supplementary Note 2.

Another consideration for Whetstone networks is how they may be affected by the limited weight precision often seen in neuromorphic hardware. While the representation of weights differs considerably across spiking neuromorphic platforms, these architectures typically do not use the full-precision floating-point representations available on conventional graphics processing units (GPUs) and central processing units (CPUs). We thus tested whether the reduced communication precision targeted by Whetstone is particularly vulnerable to reducing the weight precision as well (Supplementary Table 1). As described further in Supplementary Note 2, the reduced

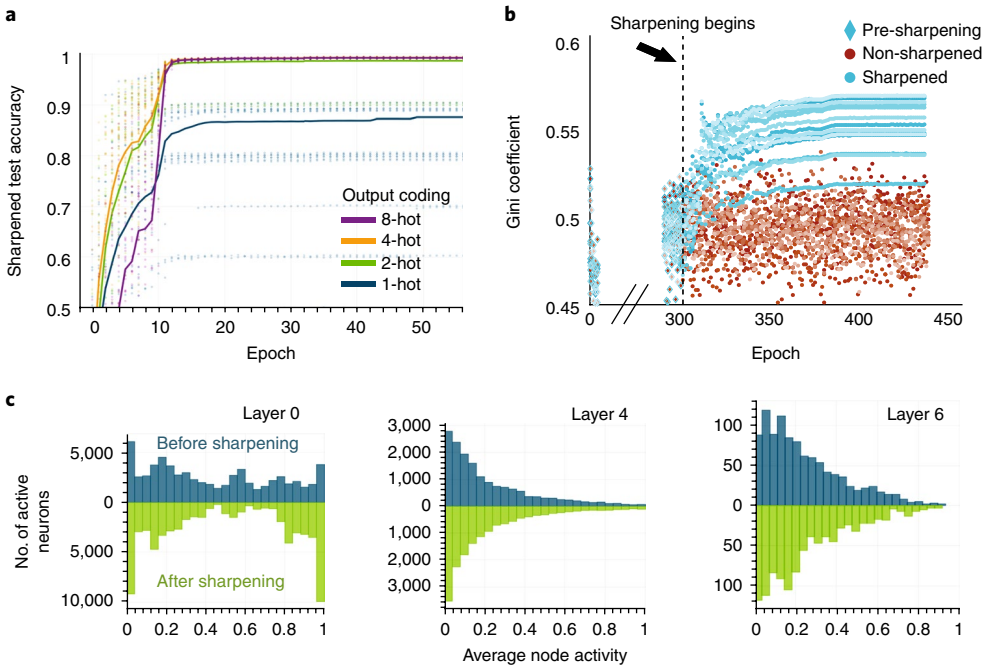


Fig. 4 | Whetstone training requires *N*-hot output encodings. a, As shown by the blue line, 1-hot encoding of outputs often fails due to dead nodes, while 2-hot encodings (green) are more stable, but do occasionally suffer from impaired performance due to dead nodes. The 4-hot (yellow) and 8-hot (red) encodings are more stable. **b**, Sharpening of neuron activations increases the Gini coefficient of trained networks. During conventional training (red), there is a moderate increase in the Gini value, suggesting a moderate inequality of neuron use in encoding training data; however, once Whetstone (blue) begins to sharpen the activation functions, starting with the input layers, the activity of neurons becomes more concentrated within a smaller population of neurons, as indicated by the reliable increase in Gini coefficients. The progression of 20 networks over the course of sharpening is shown, each with a differently shaded dot. **c**, Distribution of average node activity before (above *x* axis) and after (below *x* axis) Whetstone sharpening. Layer 0 (left) initially shows a rather broad activation of neurons, but after Whetstone most neurons are either active most of the time or never activated. Deeper layers are more sparsely active, yet the distribution does not change as much during sharpening.

fixed-point precision for weights commonly seen in neuromorphic platforms affected sharpened and non-sharpened networks similarly; however, for lower weight precision incorporating the precision into the training process is probably necessary.

Whetstone extends to several network types and tasks. Finally, we looked to examine the suitability of Whetstone on ANNs designed for non-classification tasks. As the Whetstone process is intended to be generic, we expected the process to apply to other network structures, although we anticipated that optimal performance will require some application-specific customization.

First, we examined the performance of a 12-layer convolutional network designed to identify people in images selected from the COCO dataset[19]. As shown in Fig. 5a, the Whetstone sharpened network was able to adequately identify people in the images, with an intersection-over-union of 0.482.

Second, we examined the impact of Whetstone on a convolutional autoencoder designed to reconstruct MNIST images. As shown in Fig. 5b, the reconstructed images are qualitatively similar to the sharpened network's inputs. In this example, our sharpened autoencoder (convolution, three dense layers, transpose convolution) had a mean ($N=30$) binary cross-entropy of 0.2299, standard deviation of 0.042 (compared to 0.0763 and 0.009, respectively, for pre-sharpening).

Next, we applied Whetstone to a ResNet architecture, which leverages non-local skip connections in performing classification (Fig. 5c). While some ResNet architectures contain hundreds of intermediate modules, we tested Whetstone on a 21-layer network trained on CIFAR-10. Before sharpening, this ResNet structure achieved 87.26% accuracy, and after sharpening we obtained 83.11% accuracy. While this 4.15% degradation is not suitable for most applications, this result was obtained without any customization of Whetstone for connections between non-sequential layers. In particular, as shown in Fig. 5c, the sharpening of the last stages of the ResNet leads to much of the measured loss. It is possible that the rather narrow network structure of ResNet may expose it to some of the challenges with 'dead' bRELUs as shown in Fig. 4.

Finally, we tested the ability of Whetstone to sharpen the activations of a network designed to perform deep reinforcement learning on the CartPole task. CartPole is a simple reinforcement learning task and is easily solvable by standard deep reinforcement learning methods. However, deep reinforcement learning architectures are quite different from classic supervised learning ANNs, and as such should not immediately be assumed to be compatible with any technique for sharpening activations.

As with the previous examples in this section, although we did not optimize Whetstone specifically for reinforcement learning, we were able to craft an experimental sharpener compatible with the reinforcement learning episodes. This enabled our dense SNNs to 'solve' the task; we trained a linear output network and a population code network with scores, averaged over 100 testing episodes, of 197.92 and 200, respectively (200 is a perfect score). However, these networks are extremely brittle, and convergence to an effective network is challenging, with only a small percentage of trained networks solving the task. Figure 5d shows the episode reward for each episode for representative networks, and it is easy to see that the networks are highly unstable, although some of the variability is a result of purposeful exploration. One issue is that Whetstone is (as we have seen in the classification task) best suited for wide and deep networks, whereas heavily parameterized, large networks are not

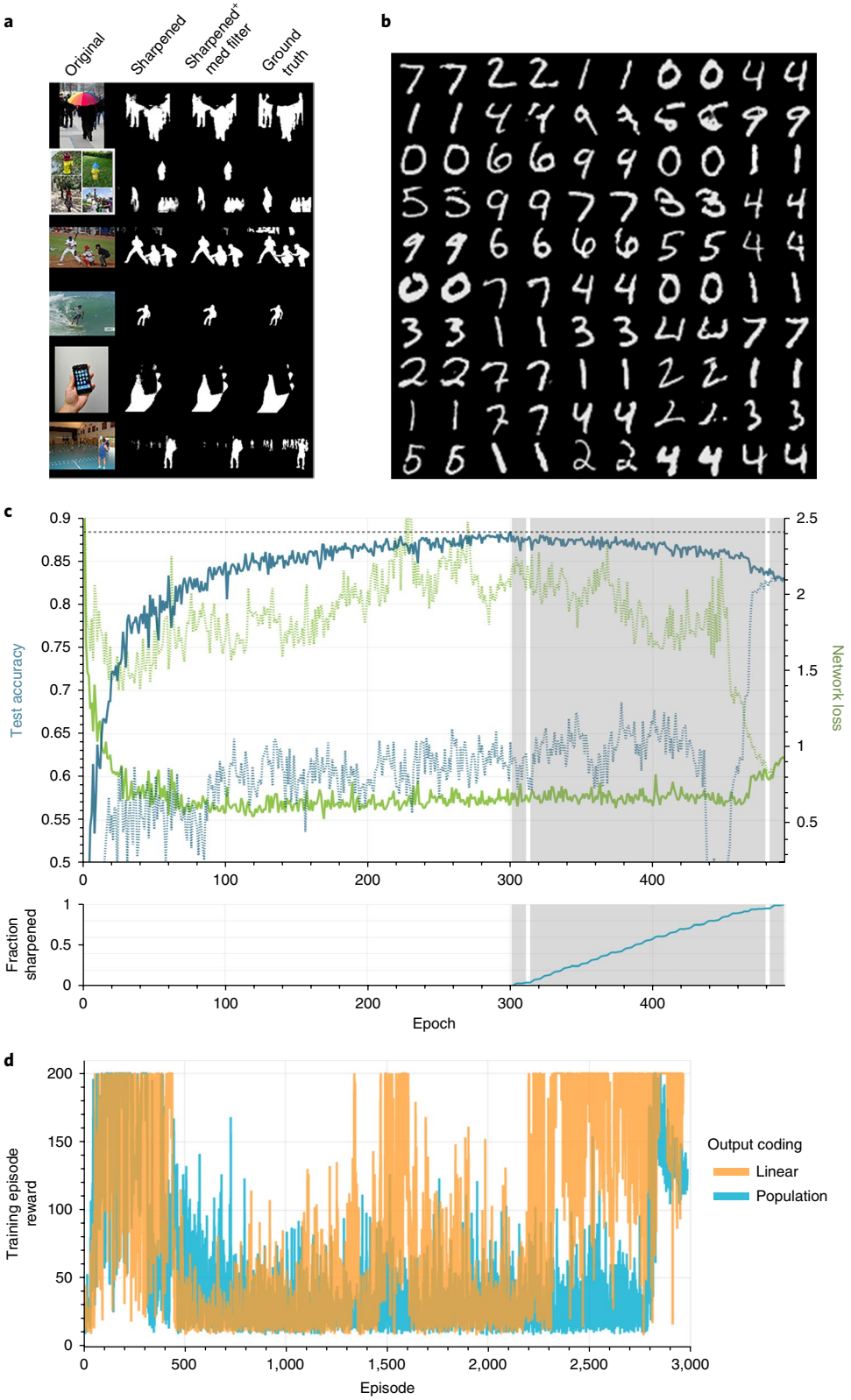


Fig. 5 | Whetstone has the ability to sharpen diverse networks. a, ANN designed for people segmentation. Left column, original image; second column, post-sharpened segmentation outputs; third column, output of sharpened network with median filtering; right column, ground truth. **b**, Outputs from the sharpened autoencoder are capable of reconstruction. For each digit pair, the left image is the original image, and the right image is reconstructed by the sharpened network. **c**, ResNet architecture trained with Whetstone sharpening is capable of maintaining most of its pre-sharpening performance level. Lines correspond to those in Fig. 2a. **d**, Reward for each training episode for a linear (orange, final average test score 200/200) and population code (blue, final average test score 197.92/200) output on the CartPole reinforcement learning task.

appropriate for the CartPole task. Further exploration of this area of research will hopefully identify ways to better tailor Whetstone to the challenges uniquely posed by reinforcement learning.

Discussion

The Whetstone method described here is intended to offer an off-the-shelf capability for machine learning practitioners to convert their ANN approaches to a spiking implementation suitable for neuromorphic hardware. There are a number of other techniques to generate SNNs, but these methods differ both in targeted representations (Whetstone targets a single time step, whereas some other methods use rate-codes or ensembles) and in training process (Whetstone trains ANNs using conventional methods to have discrete communications, as opposed to a post-training conversion or gradient-free learning)[20–27]. Details of these other techniques are available in Supplementary Note 1. Table 1 shows results for both MNIST and CIFAR-10 classification using the Whetstone method presented here as well as several of the other related techniques. Although these related techniques use different network topologies, data augmentation approaches and methodologies, they share a common goal of classification performance on the presented benchmark datasets. As we show both in Fig. 3 and Table 1, with only minimal alterations, such as *N*-hot encoding, Whetstone can achieve strong performance at only a modest penalty compared to equivalent non-spiking networks. However, some specialization, such as more extensive output encodings (Fig. 4) and appropriate choice of the optimizer (Supplementary Fig. 2a), can minimize the performance cost of using spikes to communicate between neurons.

Importantly, while the method described above is well-suited for using standard ANNs to generate SNNs compatible with neuromorphic hardware, this approach does not yet fully take advantage of other aspects of spike-based representations that potentially offer substantial savings in power efficiency. For instance, spiking neuromorphic platforms often leverage event-driven communication, wherein the only information communicated are the spike events. Therefore, there is an energy benefit to tailor an algorithm to have sparse activities. While Whetstone reduces the precision of communication to discrete 1 or 0, we currently make no attempt to sparsify the representations. We envision that techniques leveraging sparse coding approaches could be particularly advantageous when coupled with Whetstone, for this reason.

Another aspect of SNNs is that they can encode information in the time domain. In biological systems, when a spike occurs often confers more information than if a spike occurs. This temporal coding is not common to conventional ANN techniques (although it is present in some form in networks such as liquid-state machines), and because temporal coding introduces other computational trade-offs such as potentially increased latency, the value of a temporal code is limited in the applications examined here. However, for applications such as video processing in which relevant information exists across frames, the ability for spiking neurons to integrate over time may prove useful. We see that using Whetstone to train SNNs to represent information discretely is a potential first step in a true temporal spiking code, as it preserves the temporal dynamics of neurons for use in encoding dynamic information, as opposed to relying on time to encode information that could otherwise be encoded within one time step. Further work is required to fully transfer SNN function into the time domain.

Not only does the Whetstone method offer a means to make use of emerging low power neuromorphic hardware, but it may be beneficial for other accelerators as well. As GPUs and other architectures increasingly pursue the ability to perform sparse computations efficiently, the resulting binary communication from the Whetstone method is well suited for such approaches. For example, if an architecture can replace multiplications with signed addition the binarization of the communication by the Whetstone method converts an ANN to a suitable representation. Or, likewise, a sparse multiplication that can skip multiplications by zero can make use of Whetstone networks regardless of whether the architecture is for SNNs or not.

Methods

Converging to spiking activations. In contrast to many methods that convert fully trained ANNs to SNNs post hoc, the Whetstone algorithm is designed to account for a target of an SNN directly into an otherwise conventional training process. In the standard training of ANNs, for any given layer, a specific and static activation function is pre-determined. Common activation functions include tanh, sigmoid and RELUs. In current practice, RELUs have become the standard due to their quick, reliable training and high network performance. The key insight in Whetstone is that we treat this activation function as dynamic through the training process. In place of a static activation function, we update the activation while training progresses. Specifically, we use a sequence of bounded, continuous functions $h_i : \mathbb{R} \rightarrow [0, 1]$ such that h_i approaches the Heaviside function denoted as h. The Heaviside function is a specific parameterization of the threshold activations present on neuromorphic platforms, and each intermediate activation function is amenable to standard stochastic gradient descent methods. We note that because neither the convergence of the weights nor of the activation functions is uniform, we have poor theoretical guarantees in most cases. However, experimentation has shown that reliable and accurate convergence is possible in a wide variety of networks. Additionally, in practice, we see that it is often beneficial to leave the definition of h_i for training time determination, although the core concept remains unchanged.

The convergent activation method is applicable to a variety of originating activation functions. This implementation of Whetstone focuses on the bRELU. bRELUs have been shown to be as effective or nearly as effective as RELUs, and the bounded range allows them to be easily converted to a spiking threshold function[28]. We parameterize our units as

$$h_{\alpha,\beta} = \begin{pmatrix} 1, & \text{if } x_i \geq \beta \\ (x_i - \alpha)/(\beta - \alpha) & \text{if } \alpha \leq x_i < \beta \\ 0, & \text{if } x_i < \alpha \end{pmatrix} \quad (1)$$

and assert that $\alpha < \beta$ and $|\beta - 0.5| = |\alpha - 0.5|$. With $\alpha = 0$ and $\beta = 1$, $h_{\alpha,\beta}$ is a standard bRELU. However, as we let α tend towards 0.5, $h_{\alpha,\beta}$ approaches the Heaviside function. After an initial period of conventional training, the spiking bRELUs are sharpened by reducing the difference between α and β. The rate and method of convergence can be determined either before training or dynamically during training.

Figure 2 shows the training of a standard deep convolutional network on MNIST. As can be seen, by waiting several epochs to begin sharpening, the network can approach its eventual test accuracy. The progressive sharpening quickly allows binarized communication networks to effectively achieve comparable performance to the non-sharpened case.

Sharpening schedule. The sharpening of networks is performed layer by layer, and the timing of the sharpening is determined by a schedule. Our exploration of training schedules has shown that sharpening the network from the bottom up is more stable than the top-down approach (data not shown). This is probably due to the backwards flow of gradients during training; if top layers are sharpened first, all the nodes in the networks have reduced gradient information.

In addition to the direction of sharpening, we also examined programmed versus adaptive scheduling of sharpening. For programmed sharpening schedules we consider a basic paradigm where, after an initial waiting period, each layer is sharpened bottom-up over a pre-determined number of steps (either epochs or minibatches). This method is easy to implement, but ultimately the addition of sensitive hyperparameters is undesirable.

The adaptive sharpener is inspired by an error-guided closed-loop control system[29] and uses the training loss to dampen the sharpening rate, freeing the user from having to craft a sharpening schedule manually. At the end of each epoch, it looks at the change in training loss, and uses it to decide whether to sharpen for the next epoch, or pause for several epochs. When in a sharpening state, if the loss increases by more than a specified percentage, then sharpening is halted. When in a non-sharpening state, if the loss fails to improve more than a certain percentage after a certain number of epochs, then sharpening resumes. The sharpening rate is specified as the amount per layer per epoch, where amount is a floating point value less than or equal to 1.0. For example, if the sharpening rate is set to 0.25, then it will take four epochs in the sharpening state to completely sharpen one layer. It is important to note that the sharpness of a layer is altered at the end of each batch, providing a more gradual transition than if it were altered at the end of each epoch.

Our experience suggests that frequent, small updates are beneficial. This process is outlined as a state diagram in Fig. 1. In this state diagram, transition rules are only evaluated at the end of each training epoch. 'Wait' states halt sharpening for one epoch of training. Depending on the sharpening mode, the

'Sharp' state will either sharpen all model layers (for uniform) or just the current layer (for bottom-up). The process terminates when all layers of the model have been fully sharpened. While we currently use the loss to guide this process (with a manually specified rate), we hope to develop a more informed method in the future. Conceivably, the distribution of weights/activations, a longer timescale of sharpening effects and an auxiliary network loss function could all be incorporated into a more sophisticated sharpening scheme.

Output encodings. For our classification output encoding, we use an *N*-hot representation of each class. This method has helped mitigate the fragility of the sharpened networks (Fig. 4). Specifically, for an *N*-hot encoding, we design the networks such that the last learning layer has *N* neurons for each class. These neurons are independently initialized and have their own weights. To determine the loss during training, there are two main options. First, we can encode each class with its corresponding vector and use a vector-compatible loss function (for example, mean squared error). Second, we can use the spiking output as a simple population code and calculate the softmax function on these embedded values. We have found this to be the preferred method, and all classification results in this Article use the softmax method. During training, the activations of the neurons corresponding to each class are summed, and these sums are fed into a non-learning softmax calculation. In testing (or on hardware), we simply count the class with the most activations, which is equivalent because the softmax preserves the maximum value. This softmax method allows us to train using cross-entropy loss and still maintain compatibility with neuromorphic hardware targets.

The observation of dead bRELU nodes suggests that perhaps dead nodes could be mitigated directly, thus improving network performance further. To avoid the dead nodes in the output layer, we then attempted to replace the bRELU, which we know is susceptible to death upon sharpening, with a sigmoid layer followed by a softmax function (data not shown). The network performance is quite strong, with only one output neuron per class; however, the final step, which discretizes the output to a spiking form, causes a sharp degradation. This non-graceful degradation of performance on conversion to spiking supports the choice of bRELUs for the initial network training. Nevertheless, this result suggests that if output bRELUs could be kept alive, considerably higher network performance could be achieved.

Task methods. *Reduced weight precision.* For testing the effects of reduced weight precision, we investigated both a densely connected and a convolutional network under both binary and non-binary activation conditions. Under each condition, 10 replicates were trained. Trained parameters were exported for standard matrix operations. In this exported format, we could easily adjust the weight precision by casting to various fixed-point formats.

ResNet task. ResNet is an example of a network architecture that does not have exclusively sequential connectivity. ResNets include 'skip connections', whereby some layers will project to both the next layer as well as one that is several steps downstream. For our tests, we used a 21-activation-layer ResNet. In conventional hardware, the retrieval of input activations is simply a memory retrieval, but in spiking neuromorphic implementations, neuron spikes are only generated at one time, requiring the use of delays between layers if there are intermediate processing stages. This use of delays provides an example of how the spiking Whetstone activations are compatible with more temporally complex processing.

Cartpole reinforcement learning task. The Cartpole task is a classic reinforcement learning challenge, which we believe provides a strong baseline before extending to more complex tasks. We did not seek to design a novel reinforcement learning algorithm, but rather establish the compatibility of Whetstone and existing algorithms and identify some of the challenges that exist in applying spiking networks to standard reinforcement learning tasks.

The most immediate challenge is the attribution of a continuous *Q* value that represents the expected reward associated with a potential decision. *Q* learning requires that the ability to represent many values is maintained within the network even as individual neurons become discretized in their activations. One option is to have linear activation output neurons during training, but replace the output neurons on hardware with a winner-take-all circuit. Another option is to use a population code where the *Q* value is represented by the number of active neurons. For this method, loss is calculated against the sum of the activations. Another challenge is the dynamic range of the input space. For a small task like CartPole, a small number of inputs (4) have a relatively large dynamic range when compared to the number of neurons typically used to solve the task. This is generally not a problem when neurons have sufficient dynamic range and high precision, but the representational space is limited with spiking neurons.

Implementation and software package details. Whetstone is a hybrid method that is intended to provide users with a familiar interface while enabling translation to specialized hardware. Our implementation is thus intended to be compatible with conventional deep learning tools at the software level, while providing a network output suitable for implementation on spiking neuromorphic hardware (or other specialized hardware that can benefit from discrete communication). Whetstone is implemented as a set of custom Keras-compatible modules[30]. We have performed extensive testing using the Tensorflow backend, but, because Whetstone is pure Keras, it should automatically support all underlying backends such as Theano and CNTK.

Because of the challenges associated with spiking algorithms, the implementation of Whetstone was designed with the goal to 'speak the language of the deep learning researcher' so as to minimize the burden on the user. Applied here, this principle means that specifics of the underlying SNN should be abstracted away and that there should be a minimal disruption to the workflow. Compared to a standard Keras model, Whetstone-ready models generally have three modifications:

- Spiking activations: Standard RELU or sigmoid activations need to be replaced with the parameterized spiking versions provided by the Whetstone library.
- Sharpening callback: A sharpening callback must be attached during the training process. This can be simplified using a standard dynamic, adaptive sharpener or by hand-selecting stepping points (see section 'Sharpening schedule').
- Output encoding: For classification problems, it is standard practice for a network to compute a softmax activation on its logits. However, spiking platforms do not innately support this function. Instead, we wrap an output layer (with possible redundant population encoding; see section 'Output encodings') in a non-learning softmax layer that decodes any population code. On hardware, this layer can either be computed on a host machine or the raw number of spikes can be used in a winner-take-all circuit.

The network is then trained as usual, using any methods and packages compatible with Keras (for example, hyperas, opencv). Once training is completed, the final Keras model can be directly transferred to a leaky-integrate-and-fire neuron model that is compatible with spiking neural hardware. The resulting networks can be simulated either on CPUs/GPUs or implemented on neuromorphic hardware using a tool such as N2A or PyNN[31,32].

Data availability

All data used come from publicly available datasets: MNIST[34], Fashion-MNIST[35], CIFAR[36] and COCO[19]. Whetstone is available at https://github.com/SNL-NERL/Whetstone, licensed under the GPL.

Received: 12 July 2018; Accepted: 13 December 2018;
Published online: 28 January 2019

References

1. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 770–778 (IEEE, 2016).
2. Pinheiro, P. O., Collobert, R. & Dollár, P. Learning to segment object candidates. *Proc. 28th International Conference on Neural Information Processing Systems* **2**, 1990–1998 (2015).
3. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
4. Yang, T.-J., Chen, Y.-H. & Sze, V. Designing energy-efficient convolutional neural networks using energy-aware pruning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 6071–6079 (IEEE, 2017).
5. Coppola, G. & Dey, E. Driverless cars are giving engineers a fuel economy headache. *Bloomberg.com* https://www.bloomberg.com/news/articles/2017-10-11/driverless-cars-are-giving-engineers-a-fuel-economy-headache (2017).
6. Horowitz, M. 1.1 Computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* 10–14 (IEEE, 2014).
7. Jouppi, N. P. et al. In-datacenter performance analysis of a tensor processing unit. In *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)* 1–12 (IEEE, 2017).
8. Rao, N. Intel® nervana™ neural network processors (NNP) redefine AI silicon. *Intel* https://ai.intel.com/intel-nervana-neural-network-processors-nnp-redefine-ai-silicon/ (2018).
9. Hemsoth, N. Intel, Nervana shed light on deep learning chip architecture. *The Next Platform* https://www.nextplatform.com/2018/01/11/intel-nervana-shed-light-deep-learning-chip-architecture/ (2018).
10. Markidis, S. et al. Nvidia tensor core programmability, performance & precision. Preprint at https://arxiv.org/abs/1803.04014 (2018).
11. Merolla, P. A. et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **345**, 668–673 (2014).
12. Khan, M. M. et al. Spinnaker: mapping neural networks onto a massively-parallel chip multiprocessor. In *IEEE International Joint Conference on Neural Networks, 2008, IJCNN 2008 (IEEE World Congress on Computational Intelligence)* 2849–2856 (IEEE, 2008).
13. Schuman, C. D. et al. A survey of neuromorphic computing and neural networks in hardware. Preprint at https://arxiv.org/abs/1705.06963 (2017).

14. James, C. D. et al. A historical survey of algorithms and hardware architectures for neural-inspired and neuromorphic computing applications. *Biolog. Inspired Cogn. Architec.* **19**, 49–64 (2017).
15. Knight, J. C., Tully, P. J., Kaplan, B. A., Lansner, A. & Furber, S. B. Large-scale simulations of plastic neural networks on neuromorphic hardware. *Front. Neuroanat.* **10**, 37 (2016).
16. Sze, V., Chen, Y.-H., Yang, T.-J. & Emer, J. S. Efficient processing of deep neural networks: a tutorial and survey. *Proc. IEEE* **105**, 2295–2329 (2017).
17. Bergstra, J., Yamins, D. & Cox, D. D. Hyperopt: a python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in Science Conference* 13–20 (Citeseer, 2013).
18. Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A. & Talwalkar, A. Hyperband: a novel bandit-based approach to hyperparameter optimization. *J. Mach. Learn. Res.* **18**, 6765–6816 (2017).
19. Lin, T.-Y. et al. Microsoft coco: common objects in context. In *European Conference on Computer Vision*, 740–755 (Springer, 2014).
20. Hunsberger, E. & Eliasmith, C. Training spiking deep networks for neuromorphic hardware. Preprint at https://arxiv.org/abs/1611.05141 (2016).
21. Esser, S. K., Appuswamy, R., Merolla, P., Arthur, J. V. & Modha, D. S. Backpropagation for energy-efficient neuromorphic computing. In *Advances in Neural Information Processing Systems 28* (eds Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M. & Garnett, R.) 1117–1125 (Curran Associates, Red Hook, 2015).
22. Esser, S. et al. Convolutional networks for fast, energy-efficient neuromorphic computing. 2016. Preprint at http://arxiv.org/abs/1603.08270 (2016).
23. Rueckauer, B., Lungu, I.-A., Hu, Y., Pfeiffer, M. & Liu, S.-C. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Front. Neurosci.* **11**, 682 (2017).
24. Bohte, S. M., Kok, J. N. & La Poutré, J. A. Spikeprop: backpropagation for networks of spiking neurons. In *European Symposium on Artificial Neural Networks* 419–424 (ELEN, London, 2000).
25. Huh, D. & Sejnowski, T. J. Gradient descent for spiking neural networks. Preprint at https://arxiv.org/abs/1706.04698 (2017).
26. Cao, Y., Chen, Y. & Khosla, D. Spiking deep convolutional neural networks for energy-efficient object recognition. *Int. J. Comput. Vis.* **113**, 54–66 (2015).
27. Hunsberger, E. & Eliasmith, C. Spiking deep networks with LIF neurons. Preprint at https://arxiv.org/abs/1510.08829 (2015).
28. Liew, S. S., Khalil-Hani, M. & Bakhteri, R. Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems. *Neurocomputing* **216**, 718–734 (2016).
29. Nise, N. S. *Control Systems Engineering*, 5th edn (Wiley, New York, NY, 2008).
30. Chollet, F. et al. Keras https://github.com/fchollet/keras (2015).
31. Rothganger, F., Warrender, C. E., Trumbo, D. & Aimone, J. B. N2A: a computational tool for modeling from neurons to algorithms. *Front. Neural Circuits* **8**, 1 (2014).
32. Davison, A. P. et al. Pynn: a common interface for neuronal network simulators. *Front. Neuroinform.* **2**, 11 (2009).
33. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R. & Bengio, Y. Binarized neural networks. In *Proceedings of Advances in Neural Information Processing Systems* 4107–4115 (Curran Associates, Red Hook, 2016).
34. LeCun, Y., Cortes, C. & Burges, C. Mnist handwritten digit database. *AT&T Labs* http://yann.lecun.com/exdb/mnist **2** (2010).
35. Xiao, H., Rasul, K. & Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. Preprint at https://arxiv.org/abs/1708.07747 (2017).
36. Krizhevsky, A. & Hinton, G. *Learning Multiple Layers of Features from Tiny Images*. Technical Report, Univ. Toronto (2009).

Acknowledgements

This work was supported by Sandia National Laboratories' Laboratory Directed Research and Development (LDRD) Program under the Hardware Acceleration of Adaptive Neural Algorithms Grand Challenge project and the DOE Advanced Simulation and Computing program. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, a wholly owned subsidiary of Honeywell International, for the US Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

This Article describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the US Department of Energy or the US Government.

Author contributions

All authors contributed to Whetstone algorithm theory and design. W.S. and R.D. implemented code and performed experiments. W.S., C.M.V., R.D. and J.B.A. analysed results. W.S., C.M.V. and J.B.A. wrote the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s42256-018-0015-y.

Reprints and permissions information is available at www.nature.com/reprints.

Correspondence and requests for materials should be addressed to W.S. or J.B.A.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature Limited 2019

nature machine intelligence

ARTICLES

https://doi.org/10.1038/s42256-019-0019-2

Deep-learning cardiac motion analysis for human survival prediction

Ghalib A. Bello[1,8], Timothy J. W. Dawes[1,2,8], Jinming Duan[1,3], Carlo Biffi[1,3], Antonio de Marvao[1], Luke S. G. E. Howard[4], J. Simon R. Gibbs[2,4], Martin R. Wilkins[5], Stuart A. Cook[1,2,6,7], Daniel Rueckert[3] and Declan P. O'Regan[1*]

Motion analysis is used in computer vision to understand the behaviour of moving objects in sequences of images. Optimizing the interpretation of dynamic biological systems requires accurate and precise motion tracking as well as efficient representations of high-dimensional motion trajectories so that these can be used for prediction tasks. Here we use image sequences of the heart, acquired using cardiac magnetic resonance imaging, to create time-resolved three-dimensional segmentations using a fully convolutional network trained on anatomical shape priors. This dense motion model formed the input to a supervised denoising autoencoder (4Dsurvival), which is a hybrid network consisting of an autoencoder that learns a task-specific latent code representation trained on observed outcome data, yielding a latent representation optimized for survival prediction. To handle right-censored survival outcomes, our network used a Cox partial likelihood loss function. In a study of 302 patients, the predictive accuracy (quantified by Harrell's *C*-index) was significantly higher ($P = 0.0012$) for our model $C = 0.75$ (95% CI: 0.70–0.79) than the human benchmark of $C = 0.59$ (95% CI: 0.53–0.65). This work demonstrates how a complex computer vision task using high-dimensional medical image data can efficiently predict human survival.

Techniques for vision-based motion analysis aim to understand the behaviour of moving objects in image sequences[1]. In this domain, deep-learning architectures have achieved a wide range of competencies for object tracking, action recognition and semantic segmentation[2]. Making predictions about future events from the current state of a moving three-dimensional (3D) scene depends on learning correspondences between patterns of motion and subsequent outcomes. Such relationships are important in biological systems that exhibit complex spatio-temporal behaviour in response to stimuli or as a consequence of disease processes. Here we use recent advances in machine learning for visual processing tasks to develop a generalizable approach for modelling time-to-event outcomes from time-resolved 3D sensory input. We tested this on the challenging task of predicting survival due to heart disease through analysis of cardiac imaging.

The motion dynamics of the beating heart are a complex rhythmic pattern of nonlinear trajectories regulated by molecular, electrical and biophysical processes[3]. Heart failure is a disturbance of this coordinated activity characterized by adaptations in cardiac geometry and motion that lead to impaired organ perfusion[4]. For this prediction task, we studied patients diagnosed with pulmonary hypertension, characterized by right ventricular (RV) dysfunction, as this is a disease with high mortality where the choice of treatment depends on individual risk stratification[5]. Our input data were derived from cardiac magnetic resonance (CMR), which acquires imaging of the heart in any anatomical plane for dynamic assessment of function. While explicit measurements of performance obtained from myocardial motion tracking detect early contractile dysfunction and act as discriminators of different pathologies[6,7], we hypothesized that learned features of complex 3D cardiac motion would provide enhanced prognostic accuracy.

A major challenge for medical image analysis has been to automatically derive quantitative and clinically relevant information in patients with disease phenotypes. Our method employs a fully convolutional network (FCN) to learn a cardiac segmentation task from manually labelled priors. The outputs are smooth 3D renderings of frame-wise cardiac motion that are used as input data to a supervised denoising autoencoder (DAE) prediction network that we refer to as 4Dsurvival. The aim is to learn latent representations robust to noise and salient for survival prediction. We then compared our model to a benchmark of conventional human-derived volumetric indices and clinical risk factors in survival prediction.

Results

Baseline characteristics. Data from all 302 patients with incident pulmonary hypertension were included for analysis. Objective diagnosis was made according to haemodynamic and imaging criteria[5]. Patients were investigated between 2004 and 2017, and were followed-up until 27 November 2017 (median 371 days). All-cause mortality was 28% (85 of 302). Table 1 summarizes characteristics of the study sample at the date of diagnosis. No subjects' data were excluded.

Magnetic resonance image processing. Automatic segmentation of the ventricles from gated CMR images was performed for each slice position at each of 20 temporal phases, producing a total of 69,820 label maps for the cohort (Fig. 1a). Image registration was used to track the motion of corresponding anatomic points. Data for each subject were aligned, producing a dense model of cardiac

[1]MRC London Institute of Medical Sciences, Imperial College London, London, UK. [2]National Heart and Lung Institute, Imperial College London, London, UK. [3]Department of Computing, Imperial College London, London, UK. [4]Imperial College Healthcare NHS Trust, London, UK. [5]Division of Experimental Medicine, Department of Medicine, Imperial College London, London, UK. [6]National Heart Centre Singapore, Singapore, Singapore. [7]Duke-NUS Graduate Medical School, Singapore, Singapore. [8]These authors contributed equally: Ghalib A. Bello, Timothy J. W. Dawes. *e-mail: declan.oregan@imperial.ac.uk

Table 1 | Patient characteristics at the baseline (date of MRI scan)

Characteristic	*n*	% or mean ± s.d.
Age (years)		62.9 ± 14.5
Body surface area (m^2)		1.92 ± 0.25
Male	169	56
Race		
Caucasian	215	71.2
Asian	7	2.3
Black	13	4.3
Other	28	9.3
Unknown	39	12.9
World Health Organization functional class		
I	1	0
II	45	15
III	214	71
IV	42	14
Haemodynamics		
Systolic blood pressure (mmHg)		131.5 ± 25.2
Diastolic blood pressure (mmHg)		75 ± 13
Heart rate (beats min^{-1})		69.8 ± 22.5
Mean right atrial pressure (mmHg)		9.9 ± 5.8
Mean pulmonary artery pressure (mmHg)		44.1 ± 12.6
Pulmonary vascular resistance (Wood units)		8.9 ± 5.0
Cardiac output (l min^{-1})		4.3 ± 1.5
LV volumetry		
LV ejection fraction (%)		61 ± 11.1
LV end-diastolic volume (ml)		110 ± 37.4
LV end-systolic volume (ml)		44 ± 22.9
RV volumetry		
RV ejection fraction (%)		38 ± 13.7
RV end-diastolic volume (ml)		194 ± 62
RV end-systolic volume (ml)		125 ± 59.3
RV strain		
Longitudinal (%)		−16.8 ± 4.7
Radial (%)		+18.0 ± 4.4
Circumferential (%)		−9.6 ± 7.0

motion across the patient population (Fig. 1b) that was then used as an input to the 4Dsurvival network.

Predictive performance. Bootstrapped internal validation was applied to the 4Dsurvival and benchmark models. The apparent predictive accuracy for 4Dsurvival was $C=0.86$ and the optimism-corrected value was $C=0.75$ (95% confidence interval (CI): 0.70–0.79). The 4Dsurvival model outperformed (1) benchmark models of volumetric CMR parameters ($P=0.0012$): apparent predictive accuracy $C=0.60$ and optimism-adjusted $C=0.59$ (95% CI: 0.53–0.65); (2) myocardial strain parameters ($P=0.016$): apparent predictive accuracy $C=0.64$ and optimism-adjusted $C=0.61$ (95% CI: 0.57–0.66); and (3) a joint analysis of both imaging and clinical risk factors ($P=0.006$): apparent predictive accuracy $C=0.66$ and optimism-adjusted $C=0.64$ (95% CI: 0.57–0.70). Figure 2 shows Kaplan–Meier plots that depict the survival probability estimates over time, stratified by risk groups defined by each model's predictions (see Supplementary Information for details). After bootstrap validation, a final model was created using the training and optimization procedure outlined in the Methods (optimal hyperparameters for this model are summarized in Table 2).

Visualization of learned representations. To assess the ability of the 4Dsurvival network to learn discriminative features from the data, we examined the encoded representations by projection to 2D space using Laplacian eigenmaps[8] (Fig. 3a). In this figure, each subject is represented by a point, the colour of which is based on the subject's survival time (that is, the time elapsed from the baseline (date of magnetic resonance imaging (MRI) scan) to death (for uncensored patients), or to the most recent follow-up date (for censored patients)). Survival time was truncated at 7 years for ease of visualization. As is evident from the plot, our network's compressed representations of 3D motion input data show distinct patterns of clustering according to survival time. Figure 3a also shows visualizations of RV motion for two exemplar subjects at opposite ends of the risk spectrum. We also assessed the extent to which motion in various regions of the RV contributed to overall survival prediction. Fitting univariate linear models to each vertex in the mesh (see Methods for full details), we computed the association between the magnitude of cardiac motion and the 4Dsurvival network's predicted risk score, yielding a set of regression coefficients (one per vertex) that were then mapped onto a template RV mesh, producing a 3D saliency map (Fig. 3b). These show the contribution from spatially distant but functionally synergistic regions of the RV in influencing survival in pulmonary hypertension.

Discussion

Machine-learning algorithms have been used in a variety of motion analysis tasks from classifying complex traits to predicting future events from a given scene[9–11]. We show that compressed representations of a dynamic biological system moving in 3D space offer a powerful approach for time-to-event analysis. In this example, we demonstrate the effectiveness of a deep-learning algorithm, trained to find correspondences between heart motion and patient outcomes, for efficiently predicting human survival.

The traditional paradigm of epidemiological research is to draw insight from large-scale clinical studies through linear regression modelling of conventional explanatory variables, but this approach does not embrace the dynamic physiological complexity of heart disease[12]. Even objective quantification of heart function by conventional analysis of cardiac imaging relies on crude measures of global contraction that are only moderately reproducible and insensitive to the underlying disturbances of cardiovascular physiology[13]. Integrative approaches to risk classification have used unsupervised clustering of broad clinical variables to identify heart failure patients with distinct risk profiles[14,15], while supervised machine-learning algorithms can diagnose, risk stratify and predict adverse events from health record and registry data[16–18]. In the wider health domain, deep learning has achieved successes in forecasting survival from high-dimensional inputs such as cancer genomic profiles and gene expression data[19,20], and in formulating personalized treatment recommendations[21].

With the exception of natural image tasks, such as classification of skin lesions[22], biomedical imaging poses a number of challenges for machine learning as the datasets are often of limited scale, inconsistently annotated and typically high-dimensional[23]. Architectures predominantly based on convolutional neural nets, often using data augmentation strategies, have been successfully applied in computer vision tasks to enhance clinical images, segment organs and classify lesions[24,25]. Segmentation of cardiac images in the time domain is a well-established visual correspondence task that has recently achieved expert-level performance with FCN architectures[26]. Atlas-based analyses of cardiac geometry have demonstrated their value in disease classification and visualization[27–29]. Supervised principal component analysis of semi-automated segmentations has

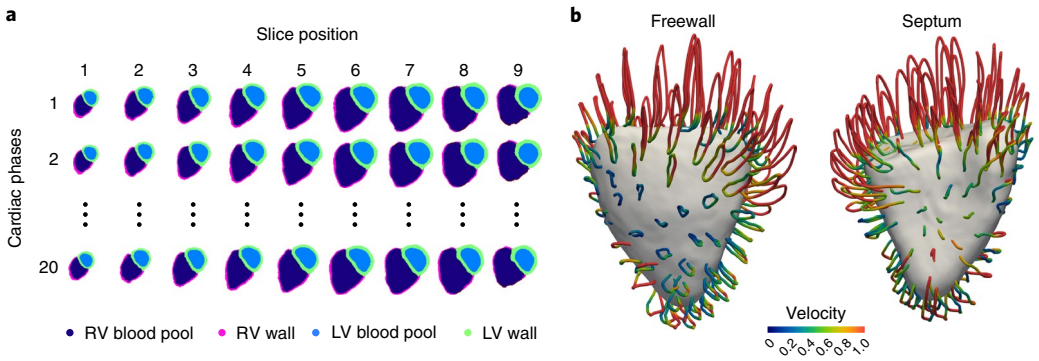


Fig. 1 | Segmentation and motion estimation. a, An example of an automatic cardiac image segmentation of each short-axis cine image from the apex (slice 1) to the base (slice 9) across 20 temporal phases. Data were aligned to a common reference space to build a population model of cardiac motion. **b**, Trajectory of RV contraction and relaxation averaged across the study population plotted as looped pathlines for a subsample of 100 points on the heart (magnification factor of ×4). The colour represents the relative myocardial velocity at each phase of the cardiac cycle. A surface-shaded model of the heart is shown at end-systole. These dense myocardial motion fields for each patient were used as an input to the prediction network.

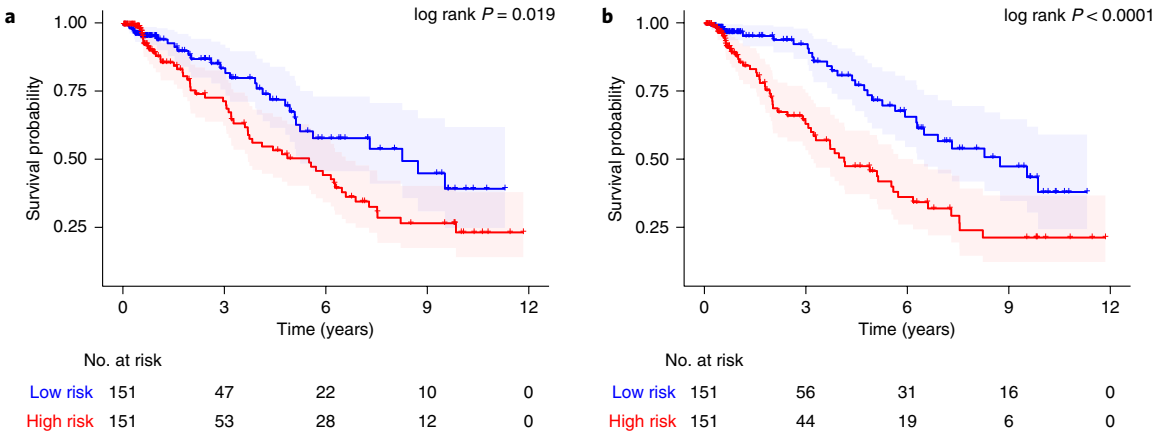


Fig. 2 | Kaplan–Meier Plots. a,**b**, Kaplan–Meier plots for a conventional parameter model using a composite of manually derived volumetric measures (**a**), and a deep-learning prediction model (4Dsurvival) whose input was time-resolved 3D models of cardiac motion (**b**). For both models, patients were divided into low- and high-risk groups by median risk score. Survival function estimates for each group (with 95% CI) are shown. For each plot, the logrank test was performed to compare survival curves between risk groups (conventional parameter model: $\chi^2 = 5.5$, $P = 0.019$; 4Dsurvival: $\chi^2 = 15.6$, $P < 0.0001$).

Table 2 | Hyperparameter search ranges for the deep-learning network (second column) and optimum hyperparameter values in the final model (third column)

Hyperparameter	Search range	Optimized value
Dropout	[0.1, 0.9]	0.71
Number of nodes in hidden layers	[75, 250]	78
Latent code dimensionality (h)	[5, 20]	13
Reconstruction loss penalty (α)	[0.3, 0.7]	0.6
Learning rate	$[10^{-6}, 10^{-4.5}]$	$10^{-4.86}$
L1 regularization penalty	$[10^{-7}, 10^{-4}]$	$10^{-5.65}$

shown prognostic utility compared to conventional parameters[30], but requires human selection of anatomical features and relies on simple predefined motion characteristics. In this work, we harness the power of deep learning for both automated image analysis and inference—learning features predictive of survival from 3D cardiac motion using nonlinear data transformations.

Autoencoding is a dimensionality reduction technique in which an encoder takes an input and maps it to a latent representation (lower-dimensional space) that is in turn mapped back to the space of the original input. The last step represents an attempt to 'reconstruct' the input from the compressed (latent) representation, and this is performed in such a way as to minimize the reconstruction error (that is, the degree of discrepancy between the input and its reconstructed version). Our algorithm is based on a DAE, a type of autoencoder that aims to extract more robust latent representations by corrupting the input with stochastic noise[31]. While conventional autoencoders are used for unsupervised learning tasks we extend recent proposals for supervised autoencoders in which the learned representations are both reconstructive and discriminative[32–38]. We achieved this by adding a prediction branch to the network with a loss function for survival inspired by the Cox proportional hazards model. A hybrid loss function, optimizing the trade-off between survival prediction and accurate input reconstruction, is calibrated during training. The compressed representations of 3D motion predict survival more accurately than a composite measure of conventional manually derived parameters measured on the same images and the improvement in performance is independent of clinical risk factors.

The main limitation of our study is relying on internal validation to evaluate predictive performance, and so the next step towards implementation is to train on larger and more diverse multicentre patient groups using image data and other prognostic variables, before performing external validation of survival prediction in a clini-

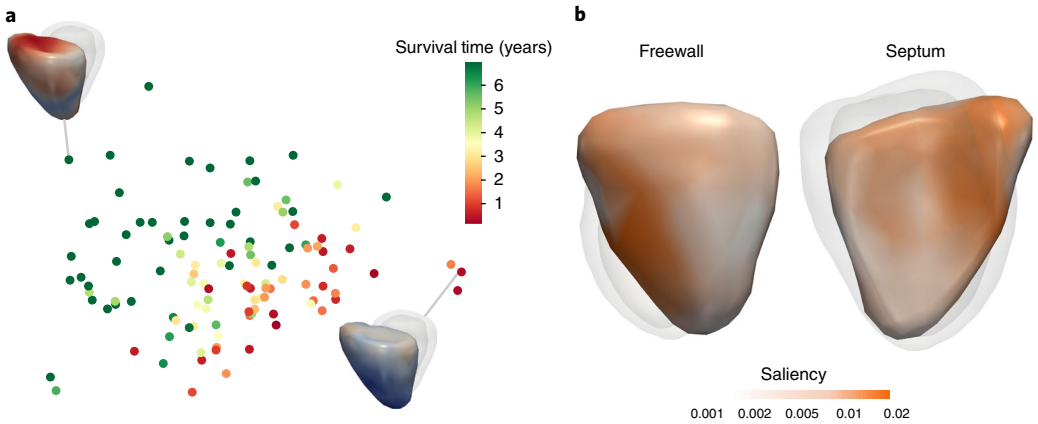


Fig. 3 | Model interpretation. a, A 2D projection of latent representations of cardiac motion in the 4Dsurvival network labelled by survival time. A visualization of RV motion is shown for two patients with contrasting risks. **b**, A saliency map showing regional contributions to survival prediction by RV motion. Absolute regression coefficients are expressed on a log scale.

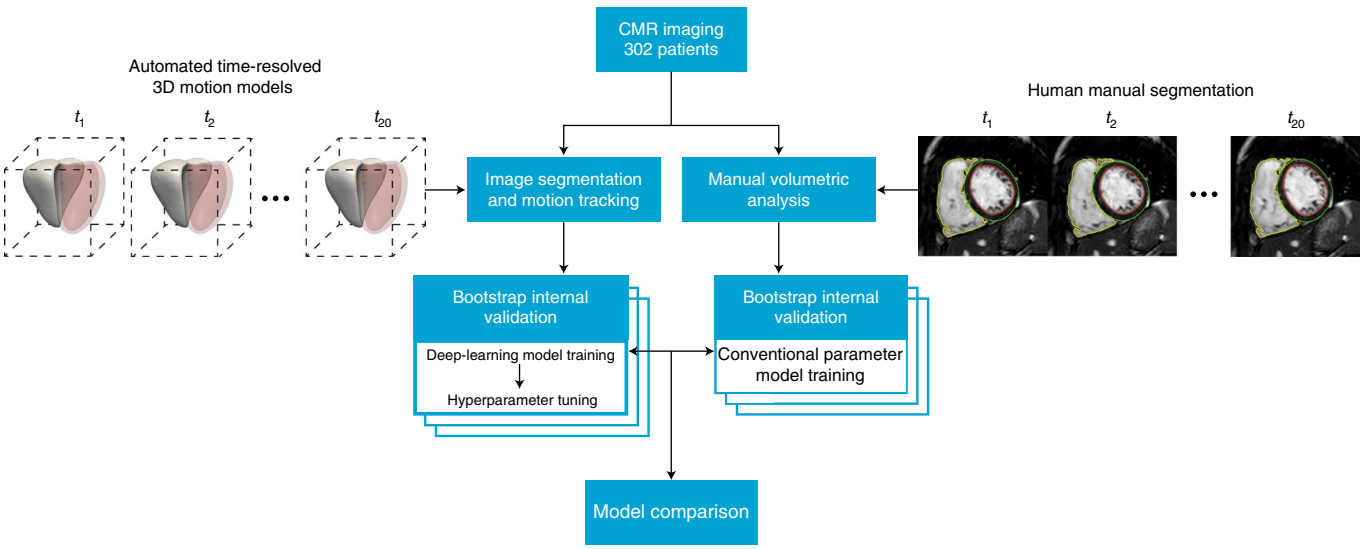


Fig. 4 | Flow chart showing the design of the study. In total, 302 patients with CMR imaging had both manual volumetric analysis and automated image segmentation (right ventricle shown in solid white, left ventricle in red) across 20 temporal phases ($t=1, ..., 20$). The internal validity of the predictive performance of a conventional parameter model and a deep-learning motion model was assessed using bootstrapping.

cal setting against a benchmark of established risk prediction scores[39]. Autoencoders may be more prone to over-fitting than methods such as principal component analysis and are more computationally expensive to train. We mitigated over-fitting using dropout and L1 regularization, and reduced the input space by down-sampling spatially correlated data. We used routinely acquired clinical data and applied normalization to compare motion acquired at different temporal resolutions. Improvement in performance may be achievable at higher temporal resolutions, but would also increase the dimension of the input data. CMR provides accurate assessment of cardiac function but other imaging modalities may offer complementary prognostic markers[40]. Further enhancement in predictive performance may be achievable by modelling multiple observations over time, for instance using long short-term memory and other recurrent neural network architectures[41,42], and handling independent competing risks[43].

Our approach enables fully automated and interpretable predictions of survival from moving clinical images—a task that has not been previously achieved in heart failure or other disease domains. This fast and scalable method is readily deployable and could have a substantial impact on clinical decision-making and the understanding of disease mechanisms. Extending this approach to other conditions where motion is predictive of survival is constrained only by the availability of suitable training cases with known outcomes.

Image acquisition and computational methods

In the following sections we describe patient data collection, the CMR protocol for image acquisition, our FCN network for image segmentation and construction of 3D cardiac motion models.

Study population. In a single-centre observational study, we analysed data collected from patients referred to the National Pulmonary Hypertension Service at the Imperial College Healthcare NHS Trust between May 2004 and October 2017. The study was approved by the Heath Research Authority and all participants gave written informed consent. Criteria for inclusion were a documented diagnosis of Group 4 pulmonary hypertension investigated by right heart catheterization with a mean pulmonary artery pressure ≥25 mmHg and pulmonary capillary wedge pressure <15 mmHg; and signs of chronic

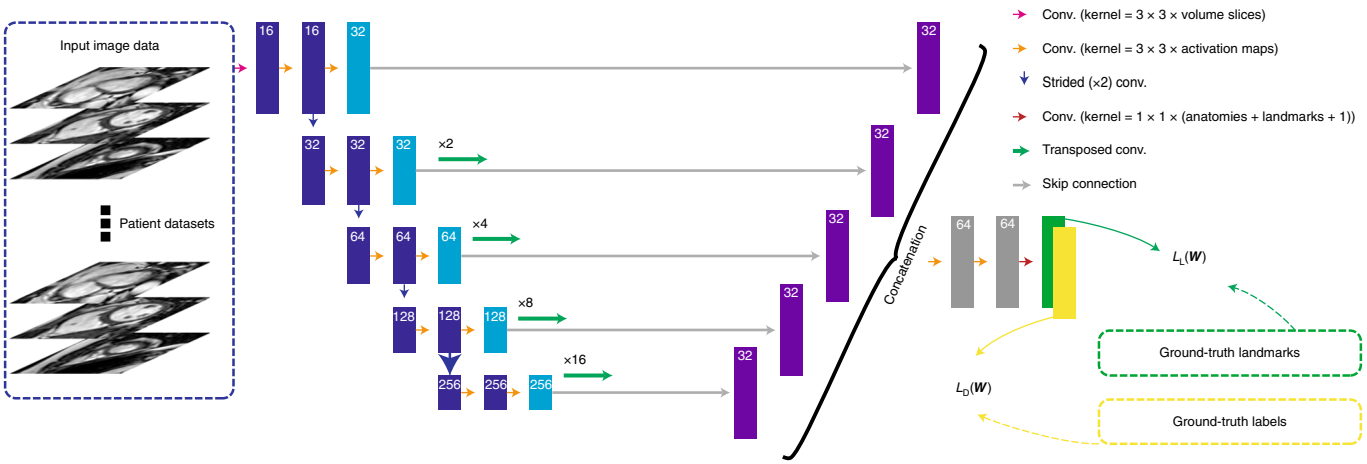


Fig. 5 | The architecture of the segmentation algorithm. A FCN takes each stack of cine images as an input, applies a branch of convolutions, learns image features from fine to coarse levels, concatenates multi-scale features and finally predicts the segmentation and landmark location probability maps simultaneously. These maps, together with the ground-truth landmark locations and label maps, are then used in the loss function (see equation (1)) that is minimized via stochastic gradient descent.

thrombo-embolic disease present on either ventilation–perfusion scintigraphy or computed tomography pulmonary angiography[44]. All patients were treated in accordance with current guidelines including medical and surgical therapy as clinically indicated[5].

MRI image acquisition, processing and computational image analysis. The CMR protocol has been previously described in detail[30]. Briefly, imaging was performed on a 1.5T Achieva (Philips), using a standard clinical protocol based on international guidelines[45]. The specific images analysed in this study were retrospectively gated cine sequences, in the short-axis plane of the heart, with a reconstructed spatial resolution of $1.3\times1.3\times10.0$ mm and a typical temporal resolution of 29 ms. Images were stored on an open-source data management system[46]. Manual volumetric analysis of the images (Fig. 4) was independently performed by accredited physicians using proprietary software (cmr42, Circle Cardiovascular Imaging) according to international guidelines with access to all available images for each subject and no analysis time constraint[47]. The derived parameters included the strongest and most well-established volumetric and functional CMR findings for prognostication reported in disease-specific meta-analyses[48,49].

We developed a convolutional neural net combined with image registration for shape-based biventricular segmentation of the CMR images. The pipeline method has three main components: segmentation, landmark localization and shape registration. First, a 2.5D multi-task FCN is trained to effectively and simultaneously learn segmentation maps and landmark locations from manually labelled volumetric CMR images. Second, multiple high-resolution 3D atlas shapes are propagated onto the network segmentation to form a smooth segmentation model. This step effectively induces a hard anatomical shape constraint and is fully automatic due to the use of predicted landmarks from the network.

We treat the problem of predicting segmentations and landmark locations as a multi-task classification problem. First, let us formulate the learning problem as follows: we denote the input training dataset by $S=\{(U_i, R_i, L_i), i=1, \ldots, N_t\}$, where N_t is the sample size of the training data, $U_i=\{u_j^i, j=1, \ldots, |U_i|\}$ is the raw input CMR volume, $R_i=\{r_j^i, j=1, \ldots, |R_i|\}$ and $r_j^i\in\{1, \ldots, N_r\}$ are the ground-truth region labels for volume U_i ($N_r=5$ representing 4 regions and background) and $L_i=\{l_j^i, j=1, \ldots, |L_i|\}$ and $l_j^i\in\{1, \ldots, N_l\}$ are the labels representing ground-truth landmark locations for U_i ($N_l=7$ representing 6 landmark locations and background). Note that $|U_i|=|R_i|=|L_i|$ stands for the total number of voxels in a CMR volume. Let $\mathbf{W}$ denote the set of all network layer parameters. In a supervised setting, we minimize the following objective function via standard (backpropagation) stochastic gradient descent:

$$L(\mathbf{W})=L_D(\mathbf{W})+\alpha L_L(\mathbf{W})+\beta\ \|\mathbf{W}\|_F^2 \quad (1)$$

where α and β are weight coefficients balancing the corresponding terms. $L_D(\mathbf{W})$ is the region-associated loss that enables the network to predict segmentation maps. $L_L(\mathbf{W})$ is the landmark-associated loss for predicting landmark locations. $\|\mathbf{W}\|_F^2$, known as the weight decay term, represents the Frobenius norm on the weights $\mathbf{W}$. This term is used to prevent the network from overfitting. The training problem is therefore to estimate the parameters $\mathbf{W}$ associated with all of the convolutional layers. By minimizing equation (1), the network is able to simultaneously predict segmentation maps and landmark locations. The definitions of the loss functions $L_D(\mathbf{W})$ and $L_L(\mathbf{W})$, used for predicting landmarks and segmentation labels, have been described previously[50].

The FCN segmentations are used to perform a non-rigid registration using cardiac atlases built from >1,000 high-resolution images[51], allowing shape constraints to be inferred. This approach produces accurate, high-resolution and anatomically smooth segmentation results from input images with low through-slice resolution, thus preserving clinically important global anatomical features. The data were split in the ratio 70:30 for training and evaluation, respectively. Motion tracking was performed for each subject using a 4D spatio-temporal B-spline image registration method with a sparseness regularization term[52]. The motion field estimate is represented by a displacement vector at each voxel and at each time frame $t=1$, ..., 20. Temporal normalization was performed before motion estimation to ensure consistency across the cardiac cycle.

Spatial normalization of each patient's data was achieved by registering the motion fields to a template space. A template image was built by registering the high-resolution atlases at the end-diastolic frame and then computing an average intensity image. In addition, the corresponding ground-truth segmentations for these high-resolution images were averaged to form a segmentation of the template image. A template surface mesh was then reconstructed from its segmentation using a 3D surface reconstruction algorithm. The motion field estimate lies within the reference space of each subject and so to enable inter-subject comparison all the segmentations were aligned to this template space by non-rigid B-spline image registration[53]. We then warped the template mesh using the resulting

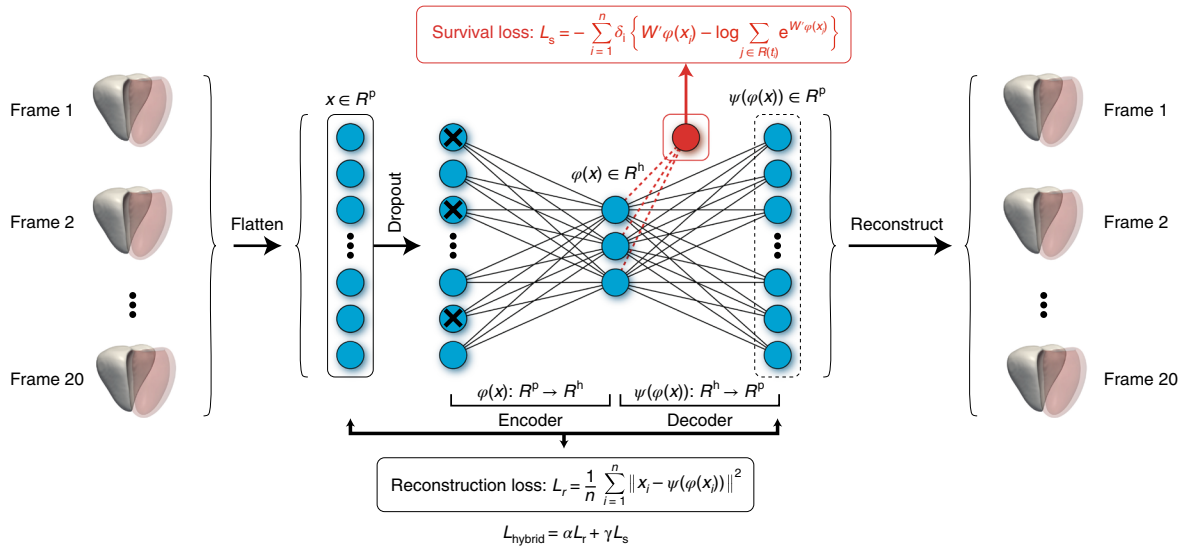


Fig. 6 | The architecture of the prediction network. The prediction network (4Dsurvival) is a DAE that takes time-resolved cardiac motion meshes as its input (right ventricle shown in solid white, left ventricle in red). For the sake of simplicity, two hidden layers, one immediately preceding and the other immediately following the central layer (latent code layer), have been excluded from the diagram. The autoencoder learns a task-specific latent code representation trained on observed outcome data, yielding a latent representation optimized for survival prediction that is robust to noise. The actual number of latent factors is treated as an optimizable parameter.

non-rigid deformation and mapped it back to the template space. Twenty surface meshes, one for each temporal frame, were subsequently generated by applying the estimated motion fields to the warped template mesh accordingly. Consequently, the surface mesh of each subject at each frame contained the same number of vertices (18,028), which maintained their anatomical correspondence across temporal frames, and across subjects (Fig. 5).

Characterization of RV motion. The time-resolved 3D meshes described in the previous section were used to produce a relevant representation of cardiac motion—in this example of right-side heart failure limited to the RV. For this purpose, we utilized a sparser version of the meshes (down-sampled by a factor of ~90) with 202 vertices. Anatomical correspondence was preserved in this process by utilizing the same vertices across all meshes. To characterize motion, we adapted an approach outlined in Bai et al.[54].

This approach is used to produce a simple numerical representation of the trajectory of each vertex (that is, the path each vertex traces through space during a cardiac cycle (see Fig. 1b)). Let (x_{vt}, y_{vt}, z_{vt}) represent the Cartesian coordinates of vertex v ($v = 1, ..., 202$) at the tth time frame ($t = 1, ..., 20$) of the cardiac cycle. At each time frame $t = 2, 3, \ldots, 20$, we compute the coordinate-wise displacement of each vertex from its position at time frame 1. This yields the following 1D input vector:

$$\mathbf{x} = (x_{vt} - x_{v1},\quad y_{vt} - y_{v1},\quad z_{vt} - z_{v1})_{2 \le t \le 20}^{1 \le v \le 202} \tag{2}$$

Vector $\mathbf{x}$ has length 11,514 ($3 \times 19 \times 202$), and was used as the input feature for our prediction network.

4Dsurvival network model

Our 4Dsurvival network structure is summarized in Fig. 6.

Network design and training. We aimed to produce an architecture capable of learning a low-dimensional representation of RV motion that robustly captures prognostic features indicative of poor survival. The architecture's hybrid design combines a DAE[55], with a Cox proportional hazards model (described below)[56].

As before, we denote our input vector by $\mathbf{x} \in \mathbb{R}^{d_p}$, where $d_p = 11{,}514$, the input dimensionality. Our network is based on a DAE, an autoencoder variant that learns features robust to noise[55]. The input vector $\mathbf{x}$ feeds directly into a stochastic masking filter layer that produces a corrupted version of $\mathbf{x}$. The masking is implemented using random dropout[57]; that is, we randomly set a fraction m of the elements of vector $\mathbf{x}$ to zero (the value of m is treated as an optimizable network parameter). The corrupted input from the masking filter is then fed into a hidden layer, the output of which is in turn fed into a central layer. This central layer represents the latent code (that is, the encoded/compressed representation of the input). This central layer is referred to as the 'code', or 'bottleneck' layer. Therefore, we may consider the encoder as a function $\phi(\cdot)$ mapping the input $\mathbf{x} \in \mathbb{R}^{d_p}$ to a latent code $\phi(\mathbf{x}) \in \mathbb{R}^{d_h}$, where $d_h \ll d_p$ (for notational convenience, we consider the corruption step as part of the encoder). This produces a compressed representation whose dimensionality is much lower than that of the input (an undercomplete representation)[58]. Note that the number of units in the encoder's hidden layer, and the dimensionality of the latent code (d_h) are not predetermined but, rather, treated as optimizable network parameters. The latent code $\phi(\mathbf{x})$ is then fed into the second component of the DAE, a multilayer decoder network that upsamples the code back to the original input dimension d_p. Like the encoder, the decoder has one intermediate hidden layer that feeds into the final layer, which in turn outputs a decoded representation (with dimension d_p matching that of the input). The size of the decoder's intermediate hidden layer is constrained to match that of the encoder network, to give the autoencoder a symmetric architecture. Dissimilarity between the original (uncorrupted) input $\mathbf{x}$ and the decoder's reconstructed version (denoted here by $\psi(\phi(\mathbf{x}))$) is penalized by minimizing a loss function of general form $L(\mathbf{x}, \psi(\phi(\mathbf{x})))$. Herein, we chose a simple mean squared error form for L:

$$L_r = \frac{1}{n} \sum_{i=1}^{n} \| \mathbf{x}_i - \psi(\phi(\mathbf{x}_i)) \|^2 \tag{3}$$

where n represents the sample size. Minimizing this loss forces the autoencoder to reconstruct the input from a corrupted/incomplete

version, thereby facilitating the generation of a latent representation with robust features. Further, to ensure that these learned features are actually relevant for survival prediction, we augmented the auto-encoder network by adding a prediction branch. The latent representation learned by the encoder $\phi(\mathbf{x})$ is therefore linked to a linear predictor of survival (see equation (4) below), in addition to the decoder. This encourages the latent representation $\phi(\mathbf{x})$ to contain features that are simultaneously robust to noisy input and salient for survival prediction. The prediction branch of the network is trained with observed outcome data (that is, survival/follow-up time). For each subject, this is the time elapsed from MRI acquisition until death (all-cause mortality), or if the subject is still alive, the last date of follow-up. Furthermore, patients receiving surgical interventions were censored at the date of surgery. This type of outcome is called a right-censored time-to-event outcome[59], and is typically handled using survival analysis techniques, the most popular of which is Cox's proportional hazards regression model[56]:

$$\log\frac{h_i(t)}{h_0(t)} = \beta_1 z_{i1} + \beta_2 z_{i2} + \ldots + \beta_p z_{ip} \tag{4}$$

Here, $h_i(t)$ represents the hazard function for subject i; that is, the 'chance' (normalized probability) of subject i dying at time t. The term $h_0(t)$ is a baseline hazard level to which all subject-specific hazards $h_i(t)$ ($i=1, \ldots, n$) are compared. The key assumption of the Cox survival model is that the hazard ratio $h_i(t)/h_0(t)$ is constant with respect to time (proportional hazards assumption)[56]. The natural logarithm of this ratio is modelled as a weighted sum of a number of predictor variables (denoted here by $z_{i1}, \ldots, z_{ip}$), where the weights/coefficients are unknown parameters denoted by $\beta_1, \ldots, \beta_p$. These parameters are estimated via maximization of the Cox proportional hazards partial likelihood function:

$$\log\mathcal{L}(\beta) = \sum_{i=1}^{n} \delta_i \left\{ \beta' \mathbf{z}_i - \log \sum_{j \in R(t_i)} \mathrm{e}^{\beta' \mathbf{z}_j} \right\} \tag{5}$$

In the expression above, $\mathbf{z}_i$ is the vector of predictor/explanatory variables for subject i, δ_i is an indicator of subject i's status (0 = alive, 1 = dead) and $R(t_i)$ represents subject i's risk set (that is, subjects still alive (and thus at risk) at the time subject i died or became censored ($\{j : t_j > t_i\}$)).

We adapt this loss function for our neural network architecture as follows:

$$L_s = -\sum_{i=1}^{n} \delta_i \left\{ \mathbf{W}' \phi(\mathbf{x}_i) - \log \sum_{j \in R(t_i)} \mathrm{e}^{\mathbf{W}' \phi(\mathbf{x}_j)} \right\} \tag{6}$$

The term $\mathbf{W}'$ denotes a $(1 \times d_h)$ vector of weights, which when multiplied by the d_h-dimensional latent code $\phi(\mathbf{x})$ yields a single scalar ($\mathbf{W}'\phi(\mathbf{x}_i)$) representing the survival prediction (specifically, natural logarithm of the hazard ratio) for subject i. Note that this makes the prediction branch of our 4Dsurvival network essentially a simple linear Cox proportional hazards model, and the predicted output may be seen as an estimate of the log hazard ratio (see equation (4)).

For our network, we combine this survival loss with the reconstruction loss from equation (3) to form a hybrid loss given by:

$$\begin{aligned} L_{\text{hybrid}} &= \alpha L_r + \gamma L_s \\ &= \alpha \left[\frac{1}{n} \sum_{i=1}^{n} \| \mathbf{x}_i - \psi(\phi(\mathbf{x}_i)) \|^2 \right] \\ &\quad + \gamma \left[-\sum_{i=1}^{n} \delta_i \left\{ \mathbf{W}' \phi(\mathbf{x}_i) - \log \sum_{j \in R(t_i)} e^{\mathbf{W}' \phi(\mathbf{x}_j)} \right\} \right] \end{aligned} \tag{7}$$

The terms α and γ are used to calibrate the contributions of each term to the overall loss (that is, to control the trade-off between survival prediction versus accurate input reconstruction). During network training, they are treated as optimizable network hyperparameters, with γ chosen to equal $1-\alpha$ for convenience.

The loss function was minimized via backpropagation. To avoid overfitting and to encourage sparsity in the encoded representation, we applied L1 regularization. The rectified linear unit activation function was used for all layers, except the output layers (linear activation was used for these layers). Using the adaptive moment estimation (Adam) algorithm, the network was trained for 100 epochs with a batch size of 16 subjects. The learning rate is treated as a hyperparameter (see Table 2). During training, the random dropout (input corruption) was repeated at every backpropagation pass. The network was implemented and trained in the Python deep-learning libraries TensorFlow[60] and Keras[61], on a high-performance computing cluster with an Intel Xeon E5-1660 CPU and NVIDIA TITAN Xp GPU. The entire training process (including hyperparameter search and bootstrap-based internal validation (see subsections below)) took a total of 131 h.

Hyperparameter tuning. To determine optimal hyperparameter values, we utilized particle swarm optimization (PSO)[62], a gradient-free meta-heuristic approach to finding optima of a given objective function. Inspired by the social foraging behaviour of birds, PSO is based on the principle of swarm intelligence, which refers to problem-solving ability that arises from the interactions of simple information-processing units[63]. In the context of hyperparameter tuning, it can be used to maximize the prediction accuracy of a model with respect to a set of potential hyperparameters[64]. We used PSO to choose the optimal set of hyperparameters from among predefined ranges of values (summarized in Table 2). We ran the PSO algorithm for 50 iterations, at each step evaluating candidate hyperparameter configurations using sixfold cross-validation. The hyperparameters at the final iteration were chosen as the optimal set. This procedure was implemented via the Python library Optunity[65].

Model validation and comparison. *Predictive accuracy metric.* Discrimination was evaluated using Harrell's concordance index[66], an extension of area under the receiver operating characteristic curve to censored time-to-event data:

$$C = \frac{\sum_{i,j} \delta_i \times I(\eta_i > \eta_j) \times I(t_i < t_j)}{\sum_{i,j} \delta_i \times I(t_i < t_j)} \tag{8}$$

In the above equation, the indices i and j refer to pairs of subjects in the sample and I denotes an indicator function that evaluates to 1 if its argument is true (and 0 otherwise). The symbols η_i and η_j denote the predicted risks for subjects i and j. The numerator tallies the number of subject pairs (i, j) where the pair member with greater predicted risk has shorter survival, representing agreement (concordance) between the model's risk predictions and ground-truth survival outcomes. Multiplication by δ_i restricts the sum to subject pairs where it is possible to determine who died first (that is, informative pairs). The C index therefore represents the fraction of informative pairs exhibiting concordance between predictions and outcomes. The index has a similar interpretation to the area under the receiver operating characteristic curve (and consequently, the same range).

Internal validation. To get a sense of how well our model would generalize to an external validation cohort, we assessed its predictive accuracy within the training sample using a bootstrap-based procedure recommended in the guidelines for transparent reporting of a multivariable model for individual prognosis or diagnosis[67]. This

procedure attempts to derive realistic, 'optimism-adjusted' estimates of the model's generalization accuracy using the training sample[68]. Below, we outline the steps of the procedure.

In step 1, a prediction model was developed on the full training sample (size n), utilizing the hyperparameter search procedure discussed above to determine the best set of hyperparameters. Using the optimal hyperparameters, a final model was trained on the full sample. Then the Harrell's concordance index (C) of this model was computed on the full sample, yielding the apparent accuracy (that is, the inflated accuracy obtained when a model is tested on the same sample on which it was trained/optimized).

In step 2, a bootstrap sample was generated by carrying out n random selections (with replacement) from the full sample. On this bootstrap sample, we developed a model (applying exactly the same training and hyperparameter search procedure used in step 1) and computed C for the bootstrap sample (henceforth referred to as bootstrap performance). Then the performance of this bootstrap-derived model on the original data (the full training sample) was also computed (henceforth referred to as test performance).

In step 3, for each bootstrap sample, the optimism was computed as the difference between the bootstrap performance and the test performance.

In step 4, steps 2 and 3 were repeated B times (where $B=100$).

In step 5, the optimism estimates derived from steps 2–4 were averaged across the B bootstrap samples and the resulting quantity was subtracted from the apparent predictive accuracy from step 1.

This procedure yields an optimism-corrected estimate of the model's concordance index:

$$C_{\text{corrected}} = C_{\text{full}}^{\text{full}} - \frac{1}{B}\sum_{b=1}^{B}\left(C_b^b - C_b^{\text{full}}\right) \quad (9)$$

Above, the symbol $C_{s_1}^{s_2}$ refers to the concordance index of a model trained on sample s_1 and tested on sample s_2. The first term refers to the apparent predictive accuracy (that is, the (inflated) concordance index obtained when a model trained on the full sample is then tested on the same sample). The second term is the average optimism (difference between bootstrap performance and test performance) over the B bootstrap samples. It has been demonstrated that this sample-based average is a nearly unbiased estimate of the expected value of the optimism that would be observed in external validation[68–71]. Subtraction of this optimism estimate from the apparent predictive accuracy gives the optimism-corrected predictive accuracy.

Conventional parameter model. As a benchmark comparison to our RV motion model, we trained a Cox proportional hazards model using conventional RV volumetric indices including RV end-diastolic volume, RV end-systolic volume and the difference between these measures expressed as a percentage of RV end-diastolic volume, RV ejection fraction, as survival predictors. We also trained a model on strain-related measures of mechanical function with tensors in the longitudinal, radial and circumferential directions[72]. A last model was trained on both the CMR parameters and a set of clinical risk factors[73], which comprised age, sex, six-minute walk distance, functional class and mean pulmonary artery pressure using the missForest algorithm to impute any missing values[74]. To account for collinearity among these predictor variables, a regularization term was added to the Cox partial likelihood function:

$$\log L(\beta) = \sum_{i=1}^{n}\delta_i\left\{\beta'\mathbf{x}_i - \log\sum_{j\in R(t_i)} e^{\beta'\mathbf{x}_j}\right\} + \frac{1}{2}\lambda\ \|\beta\|^2 \quad (10)$$

In the equation above, λ is a parameter that controls the strength of the penalty. λ was treated as a hyperparameter and its optimal value was selected via cross-validation. Internal validation of these models was carried out using the bootstrap-based procedure outlined in the previous section. Model comparisons were carried out using the R package survcomp[75] to compare concordance index measures (see Supplementary Information for further details).

Model interpretation. To facilitate interpretation of our 4Dsurvival network, we used Laplacian eigenmaps to project the learned latent code into two dimensions[8], allowing latent space visualization. Neural networks derive predictions through multiple layers of non-linear transformations on the input data. This complex architecture does not lend itself to straightforward assessment of the relative importance of individual input features. To tackle this problem, we used a simple regression-based inferential mechanism to evaluate the contribution of motion in various regions of the RV to the model's predicted risk. For each of the 202 vertices in our RV mesh models, we computed a single summary measure of motion by averaging the displacement magnitudes across 19 frames. This yielded one mean displacement value per vertex. This process was repeated across all subjects. Then we regressed the predicted risk scores onto these vertex-wise mean displacement magnitude measures using a mass univariate approach; that is, for each vertex v ($v=1, \ldots, 202$), we fitted a linear regression model where the dependent variable was predicted risk score, and the independent variable was average displacement magnitude of vertex v. Each of these 202 univariate regression models was fitted on all subjects and yielded 1 regression coefficient representing the effect of motion at a vertex on predicted risk. The absolute values of these coefficients, across all vertices, were then mapped onto a template RV mesh to provide a visualization of the differential contribution of various anatomical regions to predicted risk.

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data and code availability

Algorithms, motion models and statistical analysis are publicly available on Github under a GNU General Public License (https://github.com/UK-Digital-Heart-Project/4Dsurvival)[76]. A training simulation is available as a Docker image with an interactive Jupyter notebook hosted on Code Ocean (https://doi.org/10.24433/CO.8519672.v1)[77]. Personal data are not available due to privacy restrictions.

Received: 8 October 2018; Accepted: 9 January 2019;
Published online: 11 February 2019

References

1. Wang, L., Zhao, G., Cheng, L. & Pietikäinen, M. *Machine Learning for Vision-Based Motion Analysis: Theory and Techniques* (Springer, London, 2010).
2. Mei, T. & Zhang, C. Deep learning for intelligent video analysis. *Microsoft*; https://www.microsoft.com/en-us/research/publication/deep-learning-intelligent-video-analysis/ (2017).
3. Liang, F., Xie, W. & Yu, Y. Beating heart motion accurate prediction method based on interactive multiple model: an information fusion approach. *Biomed. Res. Int.* **2017**, 1279486 (2017).
4. Savarese, G. & Lund, L. H. Global public health burden of heart failure. *Card. Fail. Rev.* **3**, 7–11 (2017).
5. Galie, N. et al. 2015 ESC/ERS guidelines for the diagnosis and treatment of pulmonary hypertension: The Joint Task Force for the Diagnosis and Treatment of Pulmonary Hypertension of the European Society of Cardiology (ESC) and the European Respiratory Society (ERS): Endorsed by: Association for European Paediatric and Congenital Cardiology (AEPC), International Society for Heart and Lung Transplantation (ISHLT). *Eur. Heart J.* **37**, 67–119 (2016).
6. Puyol-Antón, E. et al. A multimodal spatiotemporal cardiac motion atlas from MR and ultrasound data. *Med. Image Anal.* **40**, 96–110 (2017).
7. Scatteia, A., Baritussio, A. & Bucciarelli-Ducci, C. Strain imaging using cardiac magnetic resonance. *Heart Fail. Rev.* **22**, 465–476 (2017).

8. Belkin, M. & Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems* 14 (eds Dietterich, T. G. et al.) 585–591 (MIT Press, Cambridge, 2002).
9. Li, K., Javer, A., Keaveny, E. E. & Brown, A. E. X. Recurrent neural networks with interpretable cells predict and classify worm behaviour. Preprint at https://doi.org/10.1101/222208 (2017).
10. Walker, J., Doersch, C., Gupta, A. & Hebert, M. An uncertain future: forecasting from static images using variational autoencoders. Preprint at https://arxiv.org/abs/1606.07873 (2016).
11. Bütepage, J., Black, M., Kragic, D. & Kjellström, H. Deep representation learning for human motion prediction and classification. Preprint at https://arxiv.org/abs/1702.07486 (2017).
12. Johnson, K. W. et al. Enabling precision cardiology through multiscale biology and systems medicine. *JACC Basic Transl. Sci.* **2**, 311–327 (2017).
13. Cikes, M. & Solomon, S. D. Beyond ejection fraction: an integrative approach for assessment of cardiac structure and function in heart failure. *Eur. Heart J.* **37**, 1642–1650 (2016).
14. Ahmad, T. et al. Clinical implications of chronic heart failure phenotypes defined by cluster analysis. *J. Am. Coll. Cardiol.* **64**, 1765–1774 (2014).
15. Shah, S. J. et al. Phenomapping for novel classification of heart failure with preserved ejection fraction. *Circulation* **131**, 269–279 (2015).
16. Awan, S. E., Sohel, F., Sanfilippo, F. M., Bennamoun, M. & Dwivedi, G. Machine learning in heart failure: ready for prime time. *Curr. Opin. Cardiol.* **33**, 190–195 (2018).
17. Tripoliti, E. E., Papadopoulos, T. G., Karanasiou, G. S., Naka, K. K. & Fotiadis, D. I. Heart failure: diagnosis, severity estimation and prediction of adverse events through machine learning techniques. *Comput. Struct. Biotechnol. J.* **15**, 26–47 (2017).
18. Ambale-Venkatesh, B. et al. Cardiovascular event prediction by machine learning: the multi-ethnic study of atherosclerosis. *Circ. Res.* **121**, 1092–1101 (2017).
19. Yousefi, S. et al. Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. *Sci. Rep.* **7**, 11707 (2017).
20. Ching, T., Zhu, X. & Garmire, L. X. Cox-nnet: an artificial neural network method for prognosis prediction of high-throughput omics data. *PLoS Comput. Biol.* **14**, 1–18 (2018).
21. Katzman, J. et al. DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Med. Res. Methodol.* **18**, 1–12 (2018).
22. Esteva, A. et al. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **542**, 115–118 (2017).
23. Ching, T. et al. Opportunities and obstacles for deep learning in biology and medicine.*J. R. Soc. Interface* **15**, 20170387 (2018).
24. Litjens, G. et al. A survey on deep learning in medical image analysis. *Med. Image Anal.* **42**, 60–88 (2017).
25. Shen, D., Wu, G. & Suk, H. I. Deep learning in medical image analysis. *Annu. Rev. Biomed. Eng.* **19**, 221–248 (2017).
26. Bai, W. et al. Automated cardiovascular magnetic resonance image analysis with fully convolutional networks. *J. Cardiovasc. Magn. Reson.* **20**, 65 (2018).
27. Piras, P. et al. Morphologically normalized left ventricular motion indicators from MRI feature tracking characterize myocardial infarction. *Sci. Rep.* **7**, 12259 (2017).
28. Zhang, X. et al. Orthogonal decomposition of left ventricular remodeling in myocardial infarction. *Gigascience* **6**, 1–15 (2017).
29. Zhang, X. et al. Atlas-based quantification of cardiac remodeling due to myocardial infarction. *PLoS ONE* **9**, e110243 (2014).
30. Dawes, T. et al. Machine learning of three-dimensional right ventricular motion enables outcome prediction in pulmonary hypertension: a cardiac MR imaging study. *Radiology* **283**, 381–390 (2017).
31. Rifai, S., Vincent, P., Muller, X., Glorot, X. & Bengio, Y. Contractive auto-encoders: explicit invariance during feature extraction. In *Proc. 28th International Conference on Machine Learning*, 833–840 (Omnipress, 2011).
32. Rolfe, J. T. & LeCun, Y. Discriminative recurrent sparse auto-encoders. Preprint at 1301.3775 (2013).
33. Huang, R., Liu, C., Li, G. & Zhou, J. Adaptive deep supervised autoencoder based image reconstruction for face recognition. *Math. Probl. Eng.* **2016**, 14 (2016).
34. Du, F., Zhang, J., Ji, N., Hu, J. & Zhang, C. Discriminative representation learning with supervised auto-encoder. *Neur. Proc. Lett.* https://doi.org/10.1007/s11063-018-9828-2 (2018).
35. Zaghbani, S., Boujneh, N. & Bouhlel, M. S. Age estimation using deep learning. *Comp. Elec. Eng.* **68**, 337–347 (2018).
36. Beaulieu-Jones, B. K. & Greene, C. S. Semi-supervised learning of the electronic health record for phenotype stratification. *J. Biomed. Inform.* **64**, 168–178 (2016).
37. Shakeri, M., Lombaert, H., Tripathi, S. & Kadoury, S. Deep spectral-based shape features for Alzheimer's disease classification. In *International Workshop on Spectral and Shape Analysis in Medical Imaging* (eds Reuter, M. et al.) 15–24 (Springer, 2016).
38. Biffi, C. et al. Learning interpretable anatomical features through deep generative models: Application to cardiac remodeling. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* Vol. 11071 (eds Frangi, A., Schnabel, J., Davatzikos, C., Alberola-López, C. & Fichtinger, G.) (Springer, 2018).
39. Dawes, T. J. W., Bello, G. A. & O'Regan, D. P. Multicentre study of machine learning to predict survival in pulmonary hypertension. *Open Science Framework* https://doi.org/10.17605/OSF.IO/BG6T9 (2018).
40. Grapsa, J. et al. Echocardiographic and hemodynamic predictors of survival in precapillary pulmonary hypertension: seven-year follow-up. *Circ. Cardiovasc. Imaging* **8**, 45–54 (2015).
41. Bao, W., Yue, J. & Rao, Y. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS ONE* **12**, e0180944 (2017).
42. Lim, B. & van der Schaar, M. Disease-atlas: navigating disease trajectories with deep learning. Preprint at https://arxiv.org/abs/1803.10254 (2018).
43. Lee, C., Zame, W. R., Yoon, J. & van der Schaar, M. DeepHit: a deep learning approach to survival analysis with competing risks. In *32nd Association for the Advancement of Artificial Intelligence (AAAI) Conference* (2018).
44. Gopalan, D., Delcroix, M. & Held, M. Diagnosis of chronic thromboembolic pulmonary hypertension. *Eur. Respir. Rev.* **26**, 160108 (2017).
45. Kramer, C., Barkhausen, J., Flamm, S., Kim, R. & Nagel, E. Society for cardiovascular magnetic resonance board of trustees task force on standardized protocols. Standardized cardiovascular magnetic resonance (CMR) protocols 2013 update. *J. Cardiovasc. Magn. Reson.* **15**, 91 (2013).
46. Woodbridge, M., Fagiolo, G. & O'Regan, D. P. MRIdb: medical image management for biobank research. *J. Digit. Imaging* **26**, 886–890 (2013).
47. Schulz-Menger, J. et al. Standardized image interpretation and post processing in cardiovascular magnetic resonance: society for cardiovascular magnetic resonance (SCMR) board of trustees task force on standardized post processing. *J. Cardiovasc. Magn. Reson.* **15**, 35 (2013).
48. Baggen, V. J. et al. Cardiac magnetic resonance findings predicting mortality in patients with pulmonary arterial hypertension: a systematic review and meta-analysis. *Eur. Radiol.* **26**, 3771–3780 (2016).
49. Hulshof, H. G. et al. Prognostic value of right ventricular longitudinal strain in patients with pulmonary hypertension: a systematic review and meta-analysis. *Eur. Heart J. Cardiovasc. Imaging* https://doi.org/10.1093/ehjci/jey120 (2018).
50. Duan, J. et al. Automatic 3D bi-ventricular segmentation of cardiac images by a shape-constrained multi-task deep learning approach. Preprint at 1808.08578 (2018).
51. Bai, W. et al. A bi-ventricular cardiac atlas built from 1000+ high resolution MR images of healthy subjects and an analysis of shape and motion. *Med. Image Anal.* **26**, 133–145 (2015).
52. Shi, W. et al. Temporal sparse free-form deformations. *Med. Image Anal.* **17**, 779–789 (2013).
53. Rueckert, D. et al. Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Trans. Med. Imaging* **18**, 712–721 (1999).
54. Bai, W et al. Learning a global descriptor of cardiac motion from a large cohort of 1000+ normal subjects. In *8th International Conference on Functional Imaging and Modeling of the Heart (FIMH'15)* Vol. 9126 (Springer, Cham, 2015).
55. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. & Manzagol, P.-A. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **11**, 3371–3408 (2010).
56. Cox, D. Regression models and life-tables. *J. R. Stat. Soc. B* **34**, 187–220 (1972).
57. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).
58. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, Cambridge MA, 2016).
59. Faraggi, D. & Simon, R. A neural network model for survival data. *Stat. Med.* **14**, 73–82 (1995).
60. Abadi, M. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems* (TensorFlow, 2015); http://download.tensorflow.org/paper/whitepaper2015.pdf
61. Chollet, F. et al. *Keras* https://keras.io (2015).
62. Kennedy, J. & Eberhart, R. Particle swarm optimization. *Proc. IEEE Int. Conf. Neural Net.* **4**, 1942–1948 (1995).
63. Engelbrecht, A. *Fundamentals of Computational Swarm Intelligence* (Wiley, Chichester, 2005).
64. Lorenzo, P. R., Nalepa, J., Kawulok, M., Ramos, L. S. & Pastor, J. R. Particle swarm optimization for hyper-parameter selection in deep neural networks. In *Proc. Genetic and Evolutionary Computation Conference, GECCO '17*, 481–488 (2017).
65. Claesen, M., Simm, J., Popovic, D. & De Moor, B. Hyperparameter tuning in Python using Optunity.*In Proc. International Workshop on Technical Computing for Machine Learning and Mathematical Engineering* Vol. 9 (2014).

66. Harrell, F., Califf, R., Pryor, D., Lee, K. & Rosati, R. Evaluating the yield of medical tests.*J. Am. Med. Assoc.* **247**, 2543–2546 (1982).
67. Moons, K. et al. Transparent Reporting of a multivariable prediction model for Individual Prognosis or Diagnosis (TRIPOD): explanation and elaboration. *Ann. Intern. Med.* **162**, W1–W73 (2015).
68. Harrell, F., Lee, K. & Mark, D. Tutorial in biostatistics: multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Stat. Med.* **15**, 361–387 (1996).
69. Efron, B. Estimating the error rate of a prediction rule: some improvements on cross-validation. *J. Am. Stat. Assoc.* **78**, 316–331 (1983).
70. Efron, B. & Tibshirani, R. in *An Introduction to the Bootstrap* Ch. 17 (Chapman & Hall, New York, 1993).
71. Smith, G., Seaman, S., Wood, A., Royston, P. & White, I. Correcting for optimistic prediction in small data sets. *Am. J. Epidem.* **180**, 318–324 (2014).
72. Liu, B. et al. Normal values for myocardial deformation within the right heart measured by feature-tracking cardiovascular magnetic resonance imaging. *Int. J. Cardiol.* **252**, 220–223 (2018).
73. Gall, H. et al. The Giessen pulmonary hypertension registry: survival in pulmonary hypertension subgroups. *J. Heart Lung. Transplant.* **36**, 957–967 (2017).
74. Stekhoven, D. J. & Buhlmann, P. missForest–non–parametric missing value imputation for mixed-type data. *Bioinformatics* **28**, 112–118 (2011).
75. Schroder, M. S., Culhane, A. C., Quackenbush, J. & Haibe-Kains, B. survcomp: an R/Bioconductor package for performance assessment and comparison of survival models. *Bioinformatics* **27**, 3206–3208 (2011).
76. Bello, G. A. & O'Regan, D. Deep learning cardiac motion analysis for human survival prediction (4Dsurvival) *Zenodo* https://doi.org/10.5281/zenodo.1451540 (2019).
77. Bello, G. et al. Deep learning cardiac motion analysis for human survival prediction (4Dsurvival). *Code Ocean* https://doi.org/10.24433/CO.8519672.v1 (2018).

Acknowledgements

The research was supported by the British Heart Foundation (NH/17/1/32725, RE/13/4/30184); the National Institute for Health Research Biomedical Research Centre based at Imperial College Healthcare NHS Trust and Imperial College London; and the Medical Research Council, UK. The TITAN Xp GPU used for this research was kindly donated by the NVIDIA Corporation.

Author contributions

G.A.B., C.B. and T.J.W.D. contributed to methodology, software, formal analysis and writing original draft. J.D. contributed to methodology, software and writing original draft; A.d.M. was involved with formal analysis; L.S.G.E.H., J.S.R.G., M.R.W. and S.A.C. were involved in investigation; D.R. contributed to software and supervision; D.P.O. was responsible for conceptualization, supervision, writing (review and editing) and funding acquisition. All authors reviewed the final manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s42256-019-0019-2.

Reprints and permissions information is available at www.nature.com/reprints.

Correspondence and requests for materials should be addressed to D.P.O.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature Limited 2019

nature research

Corresponding author(s): D.P. O'Regan

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see Authors & Referees and the Editorial Policy Checklist.

Statistical parameters

When statistical analyses are reported, confirm that the following items are present in the relevant location (e.g. figure legend, table legend, main text, or Methods section).

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒	An indication of whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☐	☒	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistics including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated
☐	☒	Clearly defined error bars *State explicitly what error bars represent (e.g. SD, SE, CI)*

Our web collection on statistics for biologists may be useful.

Software and code

Policy information about availability of computer code

Data collection	No software was used
Data analysis	Manual volumetric analysis of images (acquired from cardiac magnetic resonance imaging) was performed using the proprietary software cmr42 (Circle Cardiovascular Imaging, Calgary, Canada) Training and validation of deep learning and conventional parameter models was carried out using custom algorithms (available at: https://github.com/UK-Digital-Heart-Project/4Dsurvival) implemented with open-source Python language libraries Keras (v2.1.3 [http://keras.io]), Tensorflow-GPU (v1.4.0 [https://www.tensorflow.org/]), Optunity (v1.1.1 [https://github.com/claesenm/optunity]) and Lifelines (v0.14.6 [https://lifelines.readthedocs.io/en/latest/]).

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors/reviewers upon request. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

Algorithms, motion models and statistical analysis are publicly available under a GNU General Public License. A training simulation is available as a Docker image with an interactive Jupyter notebook hosted on Binder. Personal data are not available due to privacy restrictions.

Field-specific reporting

Please select the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/authors/policies/ReportingSummary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Our power calculations use classifier performance estimates obtained from preliminary data in 256 PH patients comparing the sensitivity for identifying high-risk patients using supervised learning of cardiac motion versus conventional risk factors [Dawes et al. (2017) Radiology; 283(2):381-390]. A bootstrap cross-validation of our feasibility data using nested multivariable models demonstrated an incremental benefit of ML using complex phenotypes in outcome prediction (ANOVA, Hazard Ratio: F=80.2, p<0.001; AUC: F=94.2, p<0.001).
Data exclusions	Criteria for inclusion were a documented diagnosis of Group 4 pulmonary hypertension (PH) investigated by right heart catheterization (RHC) and non-invasive imaging. Subjects with congenital heart disease were excluded.
Replication	Results have not been replicated in an external cohort. The current work represents the preliminary stage of a multicentre study that will involve external validation in a future study (for further details, see published prospective study design: Dawes TJW, Bello GA, O'Regan DP. Multicentre study of machine learning to predict survival in pulmonary hypertension. Open Science Framework (2018). DOI 10.17605/OSF.IO/BG6T9 [https://osf.io/qvx69/])
Randomization	No experimental groups were used in this study
Blinding	Blinding is not relevant to this study, as we did not utilize experimental groups.

Reporting for specific materials, systems and methods

Materials & experimental systems

n/a	Involved in the study
☒	☐ Unique biological materials
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology
☒	☐ Animals and other organisms
☐	☒ Human research participants

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☐	☒ MRI-based neuroimaging

Human research participants

Policy information about studies involving human research participants

Population characteristics	Age (years): Mean=62.9, SD=14.5 Body surface area (m2): Mean=1.92, SD=0.25 Male: n=169 (56%); Female: n=133 (34%) Race: Caucasian 71.2%, Asian 2.3%, Black 4.3%, Other 9.3%, Unknown 12.9%

WHO functional class: Class I 0%, Class II 15%, Class III 71%, Class IV 14%
Systolic BP (mmHg): Mean=131.5, SD=25.2
Diastolic BP (mmHg): Mean=75, SD=13
Heart rate (beats/min): Mean=69.8, SD=22.5
Mean right atrial pressure (mmHg): Mean=9.9, SD=5.8
Mean pulmonary artery pressure (mmHg): Mean=44.1, SD=12.6
Pulmonary vascular resistance (Wood units): Mean=8.9, SD=5.0
Cardiac output (l/min): Mean=4.3, SD=1.5
LV ejection fraction (%): Mean=61, SD=11.1
LV end diastolic volume (ml): Mean=110, SD=37.4
LV end systolic volume (ml): Mean=44, SD=22.9
RV ejection fraction (%): Mean=38, SD=13.7
RV end diastolic volume (ml): Mean=194, SD=62
RV end systolic volume (ml): Mean=125, SD=59.3

Recruitment

This study was part of a continuous prospective research program into the prognosis of patients with PH by using conventional clinical and imaging biomarkers. Our study used data (cross-sectional) collected from patients referred to the National Pulmonary Hypertension Service (at the Imperial College Healthcare NHS Trust) for routine diagnostic assessment and cardiac imaging.

Magnetic resonance imaging

Experimental design

Design type

Indicate task or resting state; event-related or block design.

Design specifications

Specify the number of blocks, trials or experimental units per session and/or subject, and specify the length of each trial or block (if trials are blocked) and interval between trials.

Behavioral performance measures

State number and/or type of variables recorded (e.g. correct button press, response time) and what statistics were used to establish that the subjects were performing the task as expected (e.g. mean, range, and/or standard deviation across subjects).

Acquisition

Imaging type(s)

Structural imaging (Cardiac)

Field strength

1.5

Sequence & imaging parameters

A standard clinical protocol for cardiac MRI was followed according to published international guidelines (Kramer, JCMR 2013;15:91). Cardiac ventricular function was assessed using balanced-steady state free precession (b-SSFP) cine imaging acquired in conventional cardiac short- and long- axis planes typically with: repetition time msec/echo time msec, 3.2/1.6; voxel size, 1.5 x 1.5 x 8 mm; flip angle, 60°; sensitivity encoding factor (SENSE), 2.0; bandwidth, 962 Hz/pixel; temporal resolution 29 msec; slice thickness 10mm; field of view 400 x 400 mm, 30 time phases.

Area of acquisition

Protocolised, three-plane, low-resolution localizer images were used to define the ventricles as the region of interest, after which long- and short-axes planes were described using a line from the apex of the heart to centre of the left ventricular base. Margins of ~1cm in all planes were added to allow for variable breath-holding.

Diffusion MRI ☐ Used ☒ Not used

Preprocessing

Preprocessing software

Provide detail on software version and revision number and on specific parameters (model/functions, brain extraction, segmentation, smoothing kernel size, etc.).

Normalization

If data were normalized/standardized, describe the approach(es): specify linear or non-linear and define image types used for transformation OR indicate that data were not normalized and explain rationale for lack of normalization.

Normalization template

Describe the template used for normalization/transformation, specifying subject space or group standardized space (e.g. original Talairach, MNI305, ICBM152) OR indicate that the data were not normalized.

Noise and artifact removal

Describe your procedure(s) for artifact and structured noise removal, specifying motion parameters, tissue signals and physiological signals (heart rate, respiration).

Volume censoring

Define your software and/or method and criteria for volume censoring, and state the extent of such censoring.

Statistical modeling & inference

Model type and settings

Specify type (mass univariate, multivariate, RSA, predictive, etc.) and describe essential details of the model at the first and second levels (e.g. fixed, random or mixed effects; drift or auto-correlation).

Effect(s) tested

Define precise effect in terms of the task or stimulus conditions instead of psychological concepts and indicate whether ANOVA or factorial designs were used.

Specify type of analysis: ☐ Whole brain ☐ ROI-based ☐ Both

Statistic type for inference
(See Eklund et al. 2016)

Specify voxel-wise or cluster-wise and report all relevant parameters for cluster-wise methods.

Correction

Describe the type of correction and how it is obtained for multiple comparisons (e.g. FWE, FDR, permutation or Monte Carlo).

Models & analysis

n/a	Involved in the study
☒	☐ Functional and/or effective connectivity
☒	☐ Graph analysis
☐	☒ Multivariate modeling or predictive analysis

Multivariate modeling and predictive analysis

Feature extraction was carried out via an image processing pipeline consisting of segmentation, co-registration and mesh generation. The output of this pipeline was a set of high-resolution, three-dimensional surface mesh representations of the heart's right ventricle (RV) at various phases of the cardiac cycle (total of 20 phases). These were used to derive point-wise displacement values representing the distance traveled by each mesh vertex (corresponding to an anatomical location on the RV) from frame to frame. These displacement values were fed as independent variables into a predictive neural network model. The neural network architecture used in this study was a 'supervised autoencoder', which combines dimension reduction with survival prediction via a Cox Proportional Hazards model. Training and validation metrics included the hazard ratio and Harrell's concordance index.

nature machine intelligence

ARTICLES

https://doi.org/10.1038/s42256-019-0017-4

Feedback GAN for DNA optimizes protein functions

Anvita Gupta[1] and James Zou[1,2]*

Generative adversarial networks (GANs) represent an attractive and novel approach to generate realistic data, such as genes, proteins or drugs, in synthetic biology. Here, we apply GANs to generate synthetic DNA sequences encoding for proteins of variable length. We propose a novel feedback-loop architecture, feedback GAN (FBGAN), to optimize the synthetic gene sequences for desired properties using an external function analyser. The proposed architecture also has the advantage that the analyser does not need to be differentiable. We apply the feedback-loop mechanism to two examples: generating synthetic genes coding for antimicrobial peptides, and optimizing synthetic genes for the secondary structure of their resulting peptides. A suite of metrics, calculated in silico, demonstrates that the GAN-generated proteins have desirable biophysical properties. The FBGAN architecture can also be used to optimize GAN-generated data points for useful properties in domains beyond genomics.

Synthetic biology refers to the systematic design and engineering of biological systems, and is a growing domain that promises to revolutionize areas such as medicine, environmental treatment and manufacturing[1]. However, current technologies for designing biological products are mostly manual and require significant domain experience. Artificial intelligence can transform the process of designing biological products by helping scientists leverage large existing genomic and proteomic datasets; by uncovering patterns in these datasets, artificial intelligence can help scientists design optimal biological molecules. Generative models, such as generative adversarial networks (GANs), can automate the process of designing DNA sequences, proteins and additional macromolecules for usage in medicine and manufacturing.

Solutions for using GANs for synthetic biology require a framework both for the GAN to generate novel sequences, and also to optimize the generated sequences for desired properties. These properties may include binding affinity of the sequence for a particular ligand, or secondary structure of the generated macromolecule. The possession of such properties by the synthetic molecules may be necessary to enable their real-world use.

Here, we present a novel feedback-loop mechanism for generating DNA sequences using a GAN and then optimizing these sequences for desired properties using a separate predictor, denoted as a function analyser.

The proposed feedback-loop mechanism is applied to train a GAN to generate protein-coding sequences (genes), and then enrich the produced genes for ones coding for antimicrobial peptides (AMPs), and ones coding for α-helical peptides. AMPs are typically lower-molecular-weight peptides with broad antimicrobial activity against bacteria, viruses and fungi[2]. They are an attractive area to apply GANs to since they are commonly short, less than 50 amino acids, and have promising applications in fighting drug-resistant bacteria[3].

Similarly, optimizing for secondary structure is a common task in protein design[4] and is feasible since common secondary structures, such as helices and β-sheets, arise even in short peptides. We optimize for α-helices in particular since these structures are more stable and robust to mutations than β-sheets[5] and AMPs are often helical.

Optimizing for these two properties provides a proof of concept that the proposed feedback-loop architecture FBGAN can be used to effectively optimize a diverse set of properties, regardless of whether a differentiable analyser is available for that property.

Related works

Generative models such as variational autoencoders and recurrent neural networks (RNNs) have also shown promise in producing sequences for synthetic biology applications.

RNNs have shown to be successful in generating SMILES sequences for de novo drug discovery[6] and recent work also showed that RNN outputs could be optimized for specific properties through transfer-learning and fine-tuning on desired sequences[7]. A similar methodology has been applied to generate AMPs[8]. RNNs have been combined with reinforcement learning to produce molecules optimized for biological activity[9].

GANs have the attractive property over RNNs that they allow for latent space interpolation through input codes provided to the generator[10]. GANs are increasingly being used to generate realistic biological data. Recently, GANs have been used to morphologically profile cell images[11], to generate time-series intensive care unit data[12] and to generate single-cell RNA-seq data from multiple cell types[13]. GANs have also been used to generate images of cells imaged by fluorescent microscopy, uniquely using a separable generator where one channel of the image was used as input to generate another channel[14]. The authors of ref. [15] explored using GAN to improve active learning; in contrast, FBGAN uses the feedback from the analyser to improve the GAN process itself.

In independent and concurrent work, Killoran et al. use GANs to generate generic DNA sequences[16]. This work used a popular variant of the GAN known as the Wasserstein GAN, which optimizes the earth mover distance between the generated and real samples[17]. In this approach, the generator was first pretrained to produce DNA sequences, and then the discriminator was replaced with a differentiable analyser. The analyser in this approach was a deep neural network that predicted, for instance, whether the input DNA sequence bound to a particular protein. By backpropagation through the analyser, the authors modified the input noise into the generator

[1]Department of Computer Science, Stanford University, Stanford, CA, USA. [2]Department of Biomedical Data Science, Stanford University, Stanford, CA, USA. *e-mail: jamesz@stanford.edu

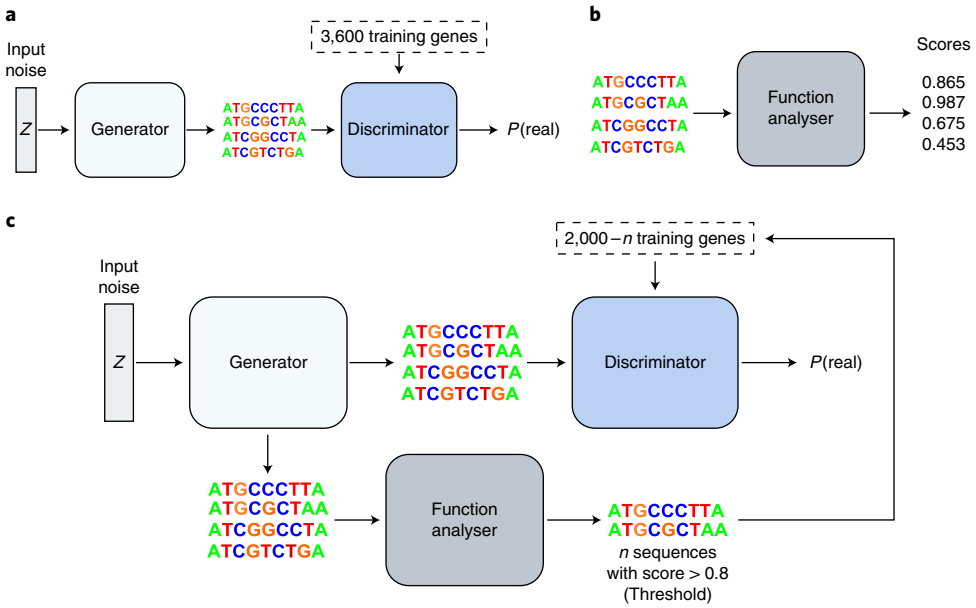


Fig. 1 | FBGAN architecture and training. a, A WGAN pretrained to produce valid genes. **b**, The general form of the function analyser, any black box that takes a sequence and produces a score. **c**, FBGAN's feedback-loop training mechanism. At every epoch, several sequences are sampled from the generator and input into the analyser. The analyser scores each sequence, and sequences above some cutoff are selected to be input back into the discriminator. They replace the least-recently added sequences in the training dataset of the discriminator, so gradually the discriminator's training data are replaced by synthetic data.

to yield desirable DNA sequences. The authors empirically demonstrated that GANs outperformed variational autoencoders for this task of generating DNA sequences. However, the approach of ref. [16] does not extend to non-differentiable analysers, and does not change the generator itself, but rather its input.

Here, we propose a novel feedback-loop architecture, FBGAN, to enrich a GAN's outputs for user-desired properties; an overview of the FBGAN architecture is provided in Fig. 1. FBGAN employs an external predictor to optimize generated data points for the desired property, and this predictor has the added benefit that it does not need to be differentiable. We present a proof of concept of the feedback-loop architecture by first generating realistic genes, or protein-coding DNA sequences, up to 50 amino acids in length (156 nucleotides); feedback is then used to enrich the generator for genes coding for AMPs, and genes coding for α-helical peptides.

Feedback GAN design and training

The basic formulation of a GAN as proposed by Goodfellow et al. consists of two component networks, a generator G and a discriminator D, where the generator G creates new data points from a vector of input noise z, and the discriminator D classifies those data points as real or fake[18]. The end goal of G is to produce data points so realistic that D is unable to classify them as fake. Each pass through the network includes a backpropagation step, where the parameters of G are improved so the generated data points appear more realistic. G and D are playing a minimax game with the loss function in equation (1)[18].

$$\min_G \max_D V(D,G) = \mathbf{E}_{x \in P_{\text{data}}(x)}[\log(D(x)] + \mathbf{E}_{z \in P(z)}[\log(1-D(G(z))] \quad (1)$$

Concretely, the discriminator seeks to maximize the probability $D(x)$ that x is real when x comes from a distribution of real data, and minimize the probability that the data point is real, $D(G(z))$, when $G(z)$ is the generated data.

The Wasserstein GAN (WGAN) is a variant of the GAN thath instead minimizes the earth mover (Wasserstein) distance between the distribution of real data and the distribution of generated data[17]. A gradient penalty is imposed for gradients above one to maintain a Lipshitz constraint[19].

WGANs have been shown empirically to be more stable during training than the vanilla GAN formulation. Moreover, the Wasserstein distance corresponds well to the quality of the generated data points[17].

Our generative models for producing genes follow the WGAN architecture with the gradient penalty proposed by Gulrajani et al.[19]. The model has five residual layers with two one-dimensional convolutions of size 5×1 each. The softmax in the final layer is replaced with a Gumbel softmax operation with temperature $t = 0.75$. When sampling from the generator, the argmax of the probability distribution is taken to output a single nucleotide at each position. Models were coded in Pytorch and initially trained for 70 epochs with a batch size $B = 64$.

GAN dataset. We first assembled a training set of naturally observed protein sequences. More than 3,655 proteins were collected from the Uniprot database, where each protein was less than 50 amino acids in length[20]. These proteins were selected from the set of all reviewed proteins in Uniprot with length from 5–50 residues, and the protein sequences were then clustered by sequence similarity ≥0.5. One representative sequence was selected from each cluster to form a diverse dataset of short peptides. The dataset was limited to proteins up to 50 amino acids in length since this length allows for observations of protein properties such as secondary structure and binding activity, while limiting the long-term dependencies the GAN would have to learn to generate protein-coding sequences.

The Uniprot peptides were then converted into complementary DNA sequences by translating each amino acid to a codon (where a random codon was selected when multiple codons mapped to one amino acid); the canonical start codon and a random stop codon were also added to each sequence. All sequences were padded to length 156, which was the maximum possible length.

Feedback-loop training mechanism. The feedback-loop mechanism consists of two components (Fig. 1). The first component is the WGAN, which generates novel gene sequences not yet enriched for any properties. The second component is the analyser; in our first use case, the analyser is a differentiable neural network that takes in a gene and predicts the probability that the gene codes for an AMP. However, the analyser can be any black box that takes in a sequence and assigns a 'favourability' score to the sequence. In our second use case, the analyser is a web server that returns the number of α-helical residues a gene will code for. An advantage of FBGAN is that it can work with any analyser and does not require the analyser to be differentiable.

The GAN and analyser are linked by the feedback mechanism after an initial number of pretraining epochs; pretraining is done so that the generator is producing valid sequences for input into the analyser. Once feedback starts, every epoch several sequences (here 15 batches) are sampled from the generator and input into the analyser. The analyser assigns a score to each sequence, and all sequences with a score above the cutoff are input back into the discriminator's dataset. The generated sequences replace the least-recently added (oldest) genes in the discriminator's training dataset. The generator's ability to mimic these new sequences is thus factored into the loss.

The GAN is then trained as usual for one epoch (one pass through this training set). As the feedback process continues, the entire training set of the discriminator is replaced repeatedly by high-scoring generated sequences.

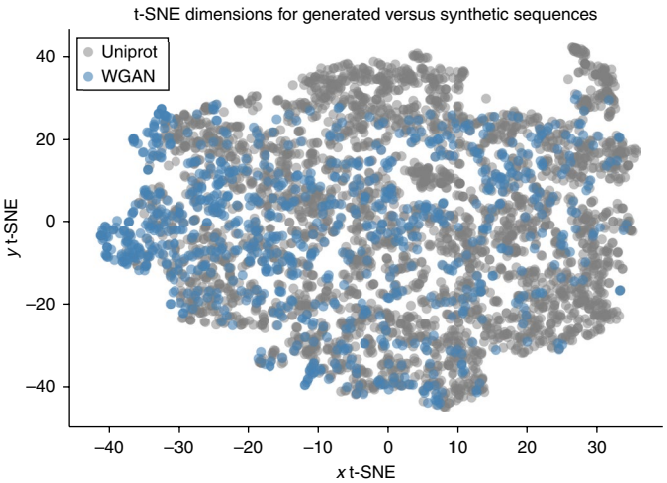


Fig. 2 | t-SNE visualization of synthetic genes. A set of 500 valid synthetic genes were randomly sampled from the trained WGAN, and 10 physiochemical features were calculated for the encoded proteins. The same ten features were also calculated for the cDNA sequences from Uniprot proteins. Dimensionality reduction was conducted through t-SNE with perplexity 40. The synthetic proteins cover a similar data manifold as the Uniprot proteins.

Analyser for AMP-coding genes. The first analyser is a recurrent classifier with gene input and output probability that the gene codes for an AMP.

Dataset. The AMP classifier was trained on 2,600 experimentally verified AMPs from the APD3 database[21], and a negative set of 2,600 randomly extracted peptides from UniProt from 10 to 50 amino acids (filtered for unnatural amino acids). The dataset was loaded using the Modlamp package[22]. As above, the proteins were translated to cDNA by translating each amino acid to a codon (a random codon in the case of redundancy), and by adding a start codon and random stop codon. The AMP training dataset was split into 60% training, 20% validation and 20% test sequences.

Classifier architecture. Using Pytorch, we built and trained a RNN classifier to predict whether a gene sequence would produce an AMP. The architecture of the RNN consisted of two gated recurrent unit (GRU) layers with hidden state $h = 128 \times 1$. The second GRU layer's output at the final time step was fed to a dense output layer, with the number of neurons equal to the number of output classes minus one. This dense layer had a sigmoid activation function, such that the output corresponded to the probability of the gene sequence being in the positive class. To reduce overfitting and improve generalization, we added dropout with $p = 0.3$ in both layers.

Using the Adam optimizer with learning rate $\mathrm{lr} = 0.001$, we optimized the binary cross-entropy loss of this network. The network was trained using minibatch gradient descent with batch size $B = 64$, and 60% of the data were retained for training, 20% for validation and 20% for testing. The model was trained for 30 epochs.

Secondary-structure black-box analyser. To optimize synthetic genes for secondary structure, a wrapper was written around the PSIPRED predictor of secondary structure[23]. The PSIPRED predictor takes in an amino-acid sequence and tags each amino acid in the sequence with known secondary structures, such as α-helix or β-sheet. The wrapper takes in a gene sequence (sampled from the generator), converts it into a protein sequence and predicts the secondary structure of the amino acids in that protein. The wrapper then outputs the total number of α-helix tagged residues. If the gene cannot be converted into a protein (due to a lack of stop/start codon, incorrect length and so on), the wrapper outputs zero. The analyser selects all sequences with helix length above some cutoff to move to the discriminator's training set. In this case, the cutoff was arbitrarily set to five residues.

Results

WGAN architecture to generate protein-coding sequences. Synthetic genes up to 156 nucleotides (50 amino acids) in length are produced from the WGAN; after training, 3 batches of sequences (192) sequences were sampled from the generator. Three batches were also sampled before one epoch of training as a comparison point.

Correct gene structure was defined as a string starting with the canonical start codon 'ATG', followed by an integer number of codons of length 3, and ending with one of three canonical stop codons ('TAA', 'TGA', 'TAG'). Before training, 3.125% of sequences initially followed the correct gene structure. After training, 77.08% of sampled sequences contained the correct gene structure.

To further examine whether the synthetic genes were similar to natural cDNA sequences from Uniprot, physiochemical features such as length, molecular weight, charge, charge density, hydrophobicity and so on were calculated.

t-distributed stochastic neighbour embedding (t-SNE) was conducted on these physiochemical features for the natural and synthetic sequences (Fig. 2). The synthetic proteins occupy a similar data manifold as the Uniprot proteins. In addition, the relative amino-acid frequencies of the synthetic sequences mirror the relative frequencies of the natural cDNA sequences from Uniprot (Supplementary Fig. 2). The WGAN for gene sequences is then used as the generator for the FBGAN architecture. We describe the analyser in the architecture below.

Deep RNN analyser for antimicrobial properties. The AMP analyser is a recurrent classifier that assigns each gene sequence a probability of coding for an AMP. The architecture of the RNN classifier consisted of two GRU layers with hidden state $h = 128 \times 1$

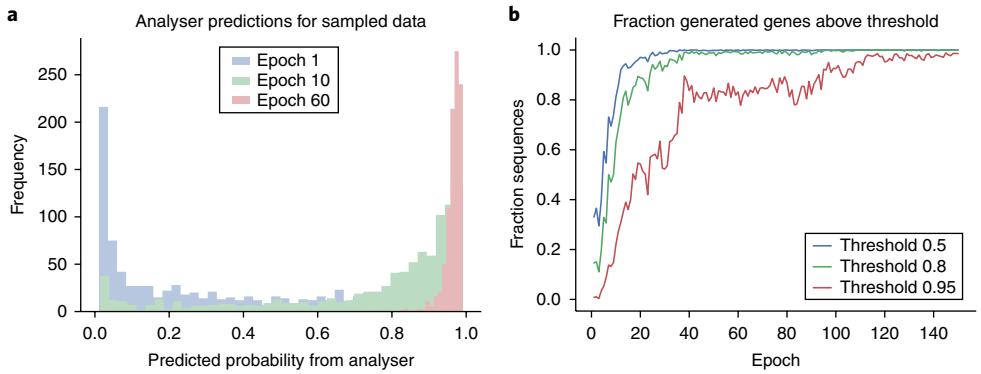


Fig. 3 | AMP analyser predictions over training epochs. a, Histograms showing the predicted probability that generated genes are antimicrobial, as the closed-loop training progresses. While most sequences are initially assigned 0.1 probability of being antimicrobial, as training progresses, nearly all sequences are eventually predicted to be antimicrobial with probability > 0.99. **b**, Percentage of sequences predicted to be antimicrobial with probability above three thresholds: [0.5,0.8,0.99]. While 0.8 was used as the cutoff for feedback, the percentage of sequences above 0.99 also continues to rise during training with feedback.

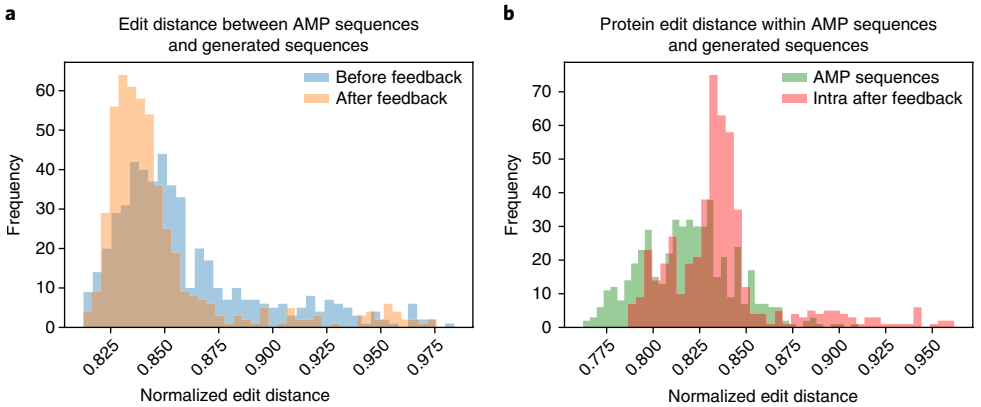


Fig. 4 | Sequence similarity for synthetic AMPs and known AMPs. a, Between-group edit distance (normalized Levenstein distance) between known antimicrobial sequences (AMPs) and proteins coded for by synthetic genes produced without feedback, and between known antimicrobial sequences (AMPs) and proteins coded for by synthetic genes produced after feedback. To calculate between-group edit distance, the distance between each synthetic protein and each AMP was calculated and the means were then plotted. **b**, Within-group edit distance for AMPs and for proteins produced after feedback, to evaluate the variability of GAN-generated genes after the feedback loop. Within-group edit distance was computed by selecting 500 sequences from the group and computing the distance between each sequence and every other sequence in the group; the mean of these distances was then taken and plotted.

Table 1 | Physiochemical properties for known and synthetic AMPs

	Known AMP	Before feedback	After feedback
	N **= 2,600**	***N*** **= 898**	***N*** **= 18,816**
Length	32.367 ± 0.00692	21.560 ± 0.01484	36.984 ± 0.00090
MW	3,514.007 ± 0.76177	2,427.960 ± 1.65405	4,023.285 ± 0.09831
Charge	3.858 ± 0.00115	2.346 ± 0.00270	2.712 ± 0.00012
ChargeDensity	0.001 ± 0.00000	0.001 ± 0.00000	0.001 ± 0.00000
pI	10.270 ± 0.00079	10.174 ± 0.00274	9.477 ± 0.00010
InstabilityInd	27.174 ± 0.01028	38.255 ± 0.03973	53.156 ± 0.00157
Aromaticity	0.082 ± 0.00002	0.063 ± 0.00008	0.078 ± 0.00000
AliphaticInd	91.859 ± 0.01817	83.751 ± 0.05176	84.890 ± 0.00185
BomanInd	0.770 ± 0.00058	1.812 ± 0.00193	0.886 ± 0.00006
HydrophRatio	0.435 ± 0.00005	0.386 ± 0.00016	0.441 ± 0.00001

Mean ± standard error (sample size given) of physiochemical features, before and after training with feedback. Properties in bold are those for which the mean after feedback is closer to the mean of the known AMPs than before feedback.

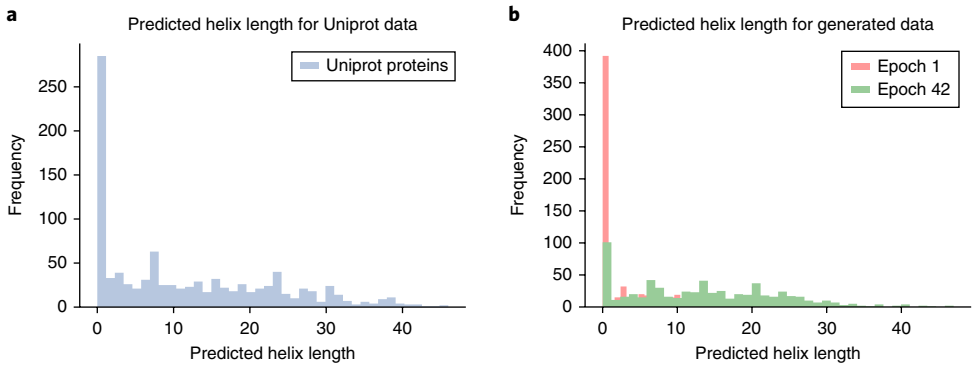


Fig. 5 | α-helix lengths of known versus synthetic proteins. a, Distribution of α-helix lengths for natural proteins under 50 amino acids scraped from Uniprot. **b**, Distribution of α-helix lengths from synthetic gene sequences after 1 and 40 epochs of training. The predicted helix length from the generated sequences quickly shifts to be higher than the helix length of the natural proteins.

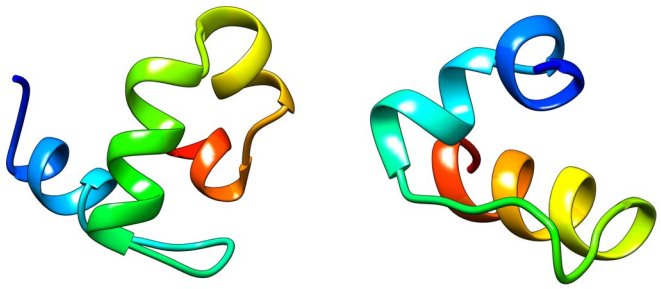


Fig. 6 | Sample α-helices from FBGAN. Example peptides from the synthetic genes output by our WGAN model with feedback from the PSIPRED analyser. Both proteins show a clear helix structure. The peptide on the left was predicted to have 10 residues arranged in helices, while the peptide on the right was predicted to have 22 residues in helices; accordingly, the peptide on the right appears to have more residues arranged in helices.

and dropout $p = 0.3$ in both layers. The function analyser achieved a training accuracy of 0.9447 and a validation accuracy of 0.8613. The test accuracy was 0.842, and the area under the receiver operating characteristic curve on the test set was 0.908. The precision and recall on the test set were 0.826 and 0.8608, respectively, and the area under the precision-recall curve was 0.88 (Supplementary Fig. 3).

Feedback loop to optimize antimicrobial properties. After both the WGAN and function analyser were trained, the two were linked with the feedback-loop mechanism; at each epoch of training, sequences were sampled from the generator and fed into the analyser. The analyser then assigned each sequence a probability of being antimicrobial, and the sequences crossing a user-defined threshold (here with $P(Antimicrobial) > 0.8$) were fed into the discriminator and labelled as 'real' sequences. The n top ranking sequences took the place of the n oldest sequences in the discriminators data set.

Evaluation of FBGAN for AMPs. FBGAN was trained with the feedback mechanism for 150 epochs (until the WGAN distance converged), and almost 20,000 sequences were sampled. The performance of FBGAN was evaluated on several criteria.

The first criterion was whether the percentage of predicted AMPs increased with feedback (without sacrificing gene structure). We examine the analyser's predictions as the training with feedback progresses. As shown in Fig. 3, after only ten epochs of closed-loop training, the analyser predicts the majority of sequences as being antimicrobial. After 60 epochs, nearly all of the sequences are predicted to be antimicrobial with high probability (greater than 0.95).

The threshold for feedback is at 0.8; however, the generator continues to improve beyond the threshold, suggesting that the closed-loop training is robust to changes in threshold. A total of 93.3% of the generated sequences after closed-loop training have the correct gene structure, showing that the reading-frame structure was not sacrificed but rather reinforced.

We also evaluated the FBGAN-generated sequences on an independent AMP predictor to measure whether the feedback loop increased the number of valid AMPs in the generated proteins. Since FBGAN as implemented here relied only on the AMP predictions of a long short-term memory (LSTM) classifier for feedback, we used the open-access AMP prediction tool from the CAMP server as a second, orthogonal test[24]. Among the sequences produced by the WGAN without feedback, 22.2% of the sequences were predicted by CAMP to be valid AMPs. If the WGAN was trained directly on positive AMP sequences without feedback, 31.9% of the sequences are predicted to be valid AMPs. The relatively low agreement could due to discrepancies between the APD3 database of antimicrobial proteins and the CAMP classifier. The outputs of FBGAN saw a significant enrichment, 40.2%, of valid CAMP AMPs.

As an additional baseline comparison, we randomly sampled 3-mers of amino acids from known AMPs and assembled them into new proteins. Only 20.2% of these proteins were predicted by CAMP to be valid AMPs.

Evaluation of physiochemical properties. The generated genes were examined for sequence-level and physiochemical-level similarity to known antimicrobial genes. Figure 4a shows a histogram of the mean edit distance between the known AMPs and proteins from synthetic genes before feedback, and the distance between the AMPs and proteins from synthetic genes produced after feedback.

Figure 4b shows the intrinsic edit distance within the AMP proteins, and within the proteins coded for by the synthetic gene sequences after feedback. All edit distances were normalized by the length of the sequences, in order to not penalize longer sequences unfairly.

The distribution shifts after feedback towards a lower edit distance from the AMP sequences. The sequences after feedback also have a higher edit distance within themselves than the AMP sequences do with each other; this demonstrates that the model has not reached the failure mode of replicating a single data point.

The physiochemical properties of the resulting proteins were measured, and are shown in Table 1. As can be seen in the table, the proteins produced with analyser feedback shift to be closer to the

positive AMPs in six out of ten physiochemical properties such as length, hydrophobicity and aromaticity. The proteins produced after feedback remain as similar as proteins produced without feedback for properties like charge and aliphatic index. Violin plots demonstrating shifts for selected properties are shown in Supplementary Fig. 1, and a principal component analysis of physiochemical properties is visualized in Supplementary Fig. 4. The shift occurs even though the analyser operates directly on the gene sequence rather than physiochemical properties, and so the feedback mechanism does not directly optimize for the physiochemical properties that demonstrate a shift.

Optimizing secondary structure with black-box PSIPRED analyser. FBGAN was then applied to produce synthetic genes coding for peptides with a particular secondary structure, here, α-helical peptides. Secondary structure is attractive to optimize since it arises and can be predicted even in short peptides, and is an extremely important property for determining protein function.

The analyser used to optimize for helical peptides is a black-box secondary-structure predictor from the PSIPRED server, which tags protein sequences with predicted secondary structure at each amino acid[23]. All gene sequences with more than five α-helical residues were input back into the discriminator as described in the feedback mechanism.

Evaluation of FBGAN for α-helices. After 43 epochs of feedback, the helix length in the generated sequences was significantly higher than the helix length without feedback and the helix length of the original Uniprot proteins, as illustrated by Fig. 5.

In addition, we independently folded several of the peptides to verify that overfitting was not occurring. Folded examples of peptides generated are shown in Fig. 6; these three-dimensional peptide structures were produced by ab initio folding from our generated gene sequences, using knowledge-based force-field template-free folding from the QUARK server[25]. The edit distance within the DNA sequences generated after feedback was in the same range as the edit distance within the Uniprot natural cDNA sequences (Supplementary Fig. 5), and higher than the edit distance within the sequences from WGAN with no feedback.

As a baseline comparison to evaluate the difficulty of generating α-helices, we sampled 3-mer residues from actual protein sequences with known amphipatic helices. We then combined these 3-mers together into synthetic peptides. These sequences had an average of only 1.42 helical residues, well below the length of helices generated by FBGAN.

Discussion

In this work, we have developed the FBGAN architecture to produce protein-coding sequences for peptides under 50 amino acids in length, and demonstrated a novel feedback-loop mechanism to optimize those sequences for desired properties. We use a function analyser to assign a score to sampled sequences from the generator at every epoch, and input sequences above some threshold back into the discriminator as 'real' data points. In this way, the outputs from the generator slowly shift over time to outputs that are assigned a high score by the function analyser. We have shown that the feedback-loop training mechanism is robust to the type of analyser used; as shown in the secondary-structure case, the analyser does not have to be differentiable for the feedback mechanism to be successful.

We have demonstrated the usefulness of the feedback-loop mechanism in two use cases: optimizing for genes that code for AMPs, and optimizing for genes that code for α-helical peptides. For the first use case, we built our own deep RNN analyser as well as evaluating on an independent AMP predictor; for the second, we employed the existing PSIPRED analyser in a black-box manner. In both cases, we were able to significantly shift the generator to produce genes likely to have the desired properties.

FBGAN produced a higher proportion of valid gene sequences that were predicted to be AMPs, when compared to both a kmer baseline and a WGAN trained directly on known AMPs.

Being able to optimize synthetic data for desired properties without a differentiable analyser is useful for two reasons. The first is that the user might not have enough positive data to effectively train a differentiable classifier to distinguish positive data points. In FBGAN, the analyser can be any model that takes in a data point and assigns it a score; the analyser may now even be a machine carrying out experiments in a laboratory.

The second reason is that many existing models in bioinformatics are based on non-differentiable operations, such as BLAST searches or homology detection algorithms. This feedback-loop mechanism allows such models to integrate smoothly with GANs.

While we were able to extend the GAN architecture to produce genes up to 156 base pairs in length while maintaining the correct start codon/stop codon structure, it was noticeably more difficult to maintain the structure at 156 base pairs than at 30 base pairs or 50. To allow the generator to learn patterns in the sequence over longer lengths, we might investigate using a recurrent architecture or even dilated convolutions in the generator, which have been shown to be effective in identifying long-term genomic dependencies[26]. It is still challenging to use GAN architectures to produce long, complex sequences, which currently limits the usefulness of GANs in designing whole proteins, which can be thousands of amino acids long.

Here, to make the training process for the GAN easier, we focus on producing gene sequences that have a clear start/stop codon structure and only four nucleotides in the vocabulary. However, in the future, we might focus on producing protein sequences directly (with a vocabulary of 26 amino acids).

While we have shown that the proteins from the synthetic genes have shifted after training to be more physiochemically similar to known AMPs, we would like to conduct additional experimental validation on the generated peptides. The same holds true for the predicted α-helical peptides.

In future work, we would also like to apply and further validate the currently proposed method on additional application areas in genomics and personalized medicine, such as noncoding DNA and RNA. In addition, FBGAN's proposed feedback-loop mechanism for training GANs is not limited to sequences or to synthetic biology applications; thus, our future work also includes applying this methodology to non-biological use cases of GANs.

Data availability

Demo, instructions and code for FBGAN are available at https://github.com/av1659/fbgan. All of the data used in this paper are publicly available and can be accessed at the references cited[22].

Received: 20 June 2018; Accepted: 4 January 2019;
Published online: 11 February 2019

References

1. Benner, S. & Sismour, M. Synthetic biology. *Nat. Rev. Genet.* **6**, 533–543 (2005).
2. Izadpanah, A. & Gallo, R. Antimicrobial peptides. *J. Am. Acad. Dermatol.* **52**, 381–390 (2005).
3. Papagianni, M. Ribosomally synthesized peptides with antimicrobial properties: biosynthesis, structure, function, and applications. *Biotechnol. Adv.* **21**, 465–499 (2003).
4. Rocklin, G. J. et al. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science* **357**, 168–175 (2017).
5. Abrusán, G. & Marsh, J. Alpha helices are more robust to mutations than beta strands. *PLoS Comput. Biol.* **12**, e1005242 (2016).
6. Segler, M., Kogej, T., Tyrchan, C. & Waller, M. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent. Sci.* **4**, 120–131 (2018).

7. Gupta, A. et al. Generative recurrent networks for de novo drug design. *Mol. Inform.* **37**, 1700111 (2018).
8. Muller, A. T., Hiss, J. A. & Schneider, G. Recurrent neural network model for constructive peptide design. *J. Chem. Inf. Model.* **58**, 472–479 (2018).
9. Olivecrona, M., Blaschke, T., Engkvist, O. & Chen, H. Molecular de novo design through deep reinforcement learning. *J. Chemoinformatics* **9**, 48 (2017).
10. Salimans, T. et al. Improved techniques for training GANs. Preprint at abs/1606.03498 (2016).
11. Goldsborough, P., Pawlowski, N., Caicedo, J., Singh, S. & Carpenter, A. Cytogan: generative modeling of cell images. Preprint at https://www.biorxiv.org/content/10.1101/227645v1 (2017).
12. Esteban, C., Hyland, S. & Rätsch, G. Real-valued (medical) time series generation with recurrent conditional GANs. Preprint at https://arxiv.org/abs/1706.02633 (2017).
13. Ghahramani, A., Watt, F. & Luscombe, N. Generative adversarial networks uncover epidermal regulators and predict single cell perturbations. Preprint at https://www.biorxiv.org/content/10.1101/262501v2 (2018).
14. Osokin, A., Chessel, A., Carazo-Salas, R. & Vaggi, F. GANs for biological image synthesis. Preprint at http://arxiv.org/abs/1708.04692 (2017).
15. Zhu, J. & Bento, J. Generative adversarial active learning. Preprint at https://arxiv.org/abs/1702.07956 (2017).
16. Killoran, N., Lee, L., Delong, A., Duvenaud, D. & Frey, B. Generating and designing DNA with deep generative models. Preprint at https://arxiv.org/abs/1712.06148 (2017).
17. Arjovsky, M., Chintala, S. & Bottou, L. Wasserstein generative adversarial networks. In *Proc. 34th International Conference on Machine Learning* (eds Precup, D. & Teh, Y. W.) 214–223 (PMLR, 2017).
18. Goodfellow, I. et al. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27* (eds Ghahramani, Z. et al.) 2672–2680 (Curran Associates, 2014).
19. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. & Courville, A. Improved training of Wasserstein GANs. Preprint at https://arxiv.org/abs/1704.00028 (2017).
20. Apweiler, R. et al. Uniprot: the universal protein knowledgebase. *Nucleic Acids Res.* **46**, 2699 (2018).
21. Wang, G., Li, X. & Wang, Z. Apd3: the antimicrobial peptide database as a tool for research and education. *Nucleic Acids Res.* **44**, D1087–D1093 (2016).
22. Muller, A., Gabernet, G., Hiss, J. & Schneider, G. modlamp: python for antimicrobial peptides. *Bioinformatics* **33**, 2753–2755 (2017).
23. Buchan, D., Minneci, F., Nugent, T., Bryson, K. & Jones, D. Scalable web services for the PSIPRED protein analysis workbench. *Nucleic Acids Res.* **41**, W349–W357 (2013).
24. Waghu, F., Barai, R., Gurung, P. & Idicula-Thomas, S. Campr3: a database on sequences, structures and signatures of antimicrobial peptides. *Nucleic Acids Res.* **44**, D1094–D1097 (2016).
25. Xu, D. & Zhang, Y. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge based force field. *Proteins* **80**, 1715–1735 (2012).
26. Gupta, A. & Rush, A. Dilated convolutions for modeling long-distance genomic dependencies. Preprint at https://arxiv.org/abs/1710.01278 (2017).

Acknowledgements

The authors thank A. Kundaje for guidance when initiating the research on GANs and DNA. J.Z. is supported by a Chan-Zuckerberg Biohub Investigator grant and National Science Foundation (NSF) grant CRII 1657155.

Author contributions

J.Z. conceived the objective of using GANs to generate genes and optimize protein functions; A.G. conceived of and implemented the feedback-loop architecture and conducted the experiments and analysis. Both authors wrote the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s42256-019-0017-4.

Reprints and permissions information is available at www.nature.com/reprints.

Correspondence and requests for materials should be addressed to J.Z.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature Limited 2019

ARTICLES
https://doi.org/10.1038/s42256-019-0018-3
nature machine intelligence

An integrated iterative annotation technique for easing neural network training in medical image analysis

Brendon Lutnick[1], Brandon Ginley[1], Darshana Govind[1], Sean D. McGarry[2], Peter S. LaViolette[3], Rabi Yacoub[4], Sanjay Jain[5], John E. Tomaszewski[1], Kuang-Yu Jen[6] and Pinaki Sarder[1]*

Neural networks promise to bring robust, quantitative analysis to medical fields. However, their adoption is limited by the technicalities of training these networks and the required volume and quality of human-generated annotations. To address this gap in the field of pathology, we have created an intuitive interface for data annotation and the display of neural network predictions within a commonly used digital pathology whole-slide viewer. This strategy used a 'human-in-the-loop' to reduce the annotation burden. We demonstrate that segmentation of human and mouse renal micro compartments is repeatedly improved when humans interact with automatically generated annotations throughout the training process. Finally, to show the adaptability of this technique to other medical imaging fields, we demonstrate its ability to iteratively segment human prostate glands from radiology imaging data.

In the current era of artificial intelligence, robust automated image analysis is attained using supervised machine-learning algorithms. This approach has been gaining considerable ground in virtually every domain of data analysis, mainly since the advent of neural networks[1–4]. Neural networks are a broad range of graphical models, whose nodes are variably activated by a nonlinear operation on the sum of their inputs[3,5]. The connections between nodes are modulated by weights, which are adjusted to alter the contribution of that node to the network output. These weights are iteratively tuned via backpropagation so that the input of data leads to a desired output (usually a classification of the data)[6]. Particularly useful for image analysis are convolutional neural networks (CNNs)[2,3], a specialized subset of neural networks. CNNs leverage convolutional filters to learn spatially invariant representations of image regions specific to the desired image classification. This allows high-dimensional filtering operations to be learned automatically, a task that has traditionally been performed through hand-engineering. The potential of neural networks exceeds that of other machine-learning techniques[7], but they are problematic in certain applications. Namely, they require significant amounts of annotated data to provide generalized high performance.

Easing the burden of data annotation is arguably as important as generating state-of-the-art network architectures, which without sufficient data are unusable[8,9]. Many large-scale modern machine-learning applications are based on cleverly designed crowd-sourced active-learning pipelines. In an era of constant firmware updates, this advancement comes in the form of human-in-the-loop training[10–12]. Initiated by low classification probabilities, machine-learning applications, such as automated teller machine character recognition, self-driving cars and Facebook's automatic tagging, all rely on user-refined training sets for fine-tuning neural network applications post deployment[3]. These 'active learning' techniques require users to 'correct' the predictions of a network, identifying gaps in network performance[13].

Although computational strategies for image analysis are increasingly being translated to biological research, the application of neural networks to biological datasets has lagged their implementation in computer science[14,15]. This late adoption of CNN-based methods is largely due to the lack of centrally curated and annotated biological training sets[16]. Due to the specialized nature of medical datasets, the expert annotation needed to generate training sets is less feasible than for traditional datasets[17]. This issue creates challenges when trying to apply CNNs to medical imaging databases, where domain-expert knowledge is required to perform image annotation. This annotation is expensive, time-consuming and labour-intensive, and there are no technical media that enable easy transference of this information from clinical practice to training sets[18].

Despite the challenges, using neural networks to segment and classify tissue slides can aid clinical diagnosis and help create improved diagnostic guidelines based on quantitative computational metrics. Moreover, neural networks can generate searchable data repositories[19], providing practicing clinicians and students access to previously unavailable collections of domain knowledge[20–22], such as labelled images and associated clinical outcomes. Achieving such access on a large scale will require a combination of curated pathological datasets, machine-learning classifiers[3], automatic anomaly detection[23,24] and efficiently searchable data hierarchies[21]. Finally, pipelines will be needed for creating easily viewable annotations on pathology images. Towards this aim, we have developed an iterative interface between the successful semantic segmentation network DeepLab v2[25] and the widely used whole-slide image (WSI) viewing software Aperio ImageScope[26], which we have termed Human AI Loop (H-AI-L) (Fig. 1). Put simply, the algorithm converts annotated regions stored in XML format (provided in ImageScope) into

[1]Department of Pathology & Anatomical Sciences, SUNY Buffalo, New York, NY, USA. [2]Department of Biophysics, Medical College of Wisconsin, Wauwatosa, WI, USA. [3]Department of Radiology and Biomedical Engineering, Medical College of Wisconsin, Wauwatosa, WI, USA. [4]Department of Medicine, Nephrology, SUNY Buffalo, New York, NY, USA. [5]Department of Medicine, Nephrology, Washington University School of Medicine, St Louis, MO, USA. [6]Department of Pathology, University of California, Davis Medical Center, Sacramento, CA, USA. *e-mail: pinakisa@buffalo.edu

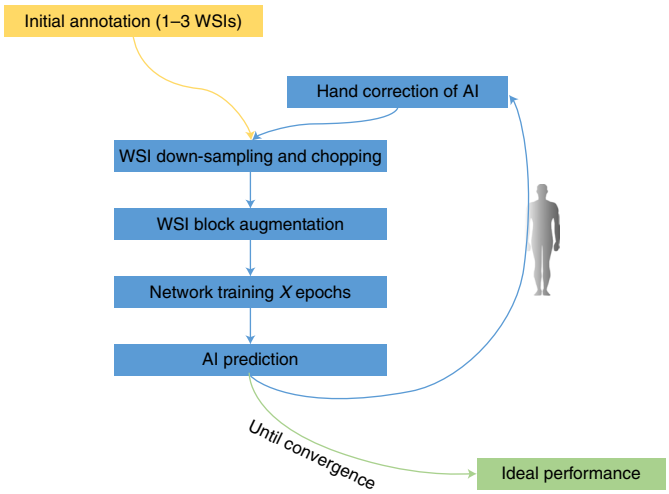


Fig. 1 | Iterative H-AI-L pipeline overview. Schematic representation of the H-AI-L pipeline for training semantic segmentation of WSIs. Several rounds of training are performed using human expert feedback to optimize ideal performance, resulting in improved efficiency in network training with limited numbers of initial annotated WSIs.

image region masks. These masks are used to train the semantic segmentation network, whose predictions are converted back to XML format for display in ImageScope. This graphical display of the network output is an ideal visualization tool for making segmentation predictions on WSIs. It allows the entire tissue slide to be viewed, with panning and zooming, and it uses the efficient JPG2000 decompression[27] of WSI files provided by ImageScope. Note that while the current code works only in ImageScope, the proposed system can easily be adapted for other WSI viewers, such as the universal viewer Pathcore Sedeen[28], as well as ImageJ. Note also that ImageScope and the DeepLab architecture are not currently approved for diagnostic procedures. Therefore, for any potential application of our system in a clinical workflow, our pipeline needs to be adopted using annotation and machine-learning tools that are currently approved for clinical diagnosis.

Using this open-sourced pipeline, a supervising domain expert can correct the network predictions (deleting false positives and annotating false-negative regions) before initiating further training using the newly annotated data. Thus, networks can be trained either 'on demand' or as the data become available. Using H-AI-L, we are able to significantly reduce the annotation effort required to learn robust segmentations of large microscopy images[28]. Adapting this technique to other modes of medical imaging is highly feasible, which we demonstrate using magnetic resonance imaging (MRI) data.

Results

To evaluate the utility of H-AI-L, we first quantified its performance and efficiency in segmenting histologic sections of kidney tissue, beginning with glomerular localization in mouse kidney WSIs[4,29–32]. This glomeruli segmentation network was trained for five iterations, using a combination of periodic acid–Schiff (PAS) and haematoxylin and eosin (H&E)-stained murine renal sections. For more data variation, streptozotocin (STZ)-induced diabetic nephropathy[33–36] murine data were included in iteration 4 (Table 1). To validate the performance of our network, we use four holdout WSIs, including one STZ-induced WSI.

During the training process, we observed approximately four- to tenfold increases in average glomerular annotation speed between the initial and end iterations (Fig. 2a). Compared to each annotator's baseline speed, these increases represent time savings of 81.4, 82 and 72.7% for annotators 1, 2 and 3, respectively. The prediction

Table 1 | H-AI-L segmentation mouse WSI training and testing datasets

H-AI-L dataset							
Annotation iteration		0	1	2	3	4	Test
WSIs added		1	2	4	6	4	4
Total glomeruli	Normal	32	84	86	418	0	138
	STZ	0	0	0	0	293	96

Mouse WSI training set used to train the glomerular segmentation network. Data presenting structural damage from STZ-induced diabetes[1] were introduced in iteration 4. The test dataset included three normal and one STZ-induced murine renal WSI.

performance increase is shown in Fig. 2b, where the network reaches nearly perfect performance on a holdout dataset by annotation iteration 4. One side effect of using iterative annotation is intuitive qualification of network performance after each interaction. That is, an expert interacts with the network predictions after each training round, visualizing network biases and shortcomings on holdout data. Two examples of evolving network predictions are highlighted in Supplementary Video 1.

To improve network prediction efficiency, we designed a two-stage segmentation approach. This uses two segmentation networks, first identifying hotspot regions at 1/16th scale and then segmenting them at the highest resolution. This approach (which we call multi-pass segmentation) provides a better F-measure (F1 score)[37,38] (Fig. 2b) than a full-resolution pass, as well as approximately 4.5-times faster predictions (Fig. 2c). An overview of this method can be found in Supplementary Fig. 1.

Quantification of the performance achieved by our method in WSIs is a challenge due to the imbalance between class distributions[39]. Therefore, we choose to report the F-measure, which considers both precision and recall (sensitivity) simultaneously[37], as specificity and accuracy are always high because the negative region is large with respect to the positive class. This choice of using the F-measure is particularly important considering the performance characteristics of multi-pass segmentation. During testing we found that the multi-pass approach trades segmentation sensitivity for increased precision, while outperforming full analysis overall, with an improved F1 score (Fig. 2). This result is due to a lower false-positive rate achieved by multi-pass segmentation as a result of the low-resolution network pre-pass, which limits the amount of background region seen by the high-resolution network. Overall (on four holdout WSIs), our network achieved its best performance after the fifth iteration of training using multi-pass segmentation, with a sensitivity of 0.92 ± 0.02, specificity of 0.99 ± 0.001, precision of 0.93 ± 0.14 and accuracy of 0.99 ± 0.001.

Network performance analysis is further complicated by human annotation errors. We note several instances where network predictions outperformed human annotators, despite being trained using flawed annotations. This phenomenon is highlighted in Fig. 3, where glomerular regions annotated manually in iteration 0 are compared to the iteration 5 network predictions. Such errors are more prevalent in WSIs annotated in early iterations, where network predictions need the most correction.

To qualitatively prove the effectiveness and extendibility of our method, we show its extension to multi-class detection by segmenting glomerular nuclei types[40,41] and interstitial fibrosis and tubular atrophy (IFTA)[42,43], as well as by differentiating sclerotic and non-sclerotic glomeruli[44]. This analysis is performed in mouse kidney and human renal biopsies. Figure 4 shows the glomeruli detection network from Fig. 2 adapted for nuclei detection. This study was carried out by retraining the high-resolution network using a set of 143 glomeruli with labelled podocyte and non-podocyte nuclei, marked via immunofluorescence labelling. For this analysis, the

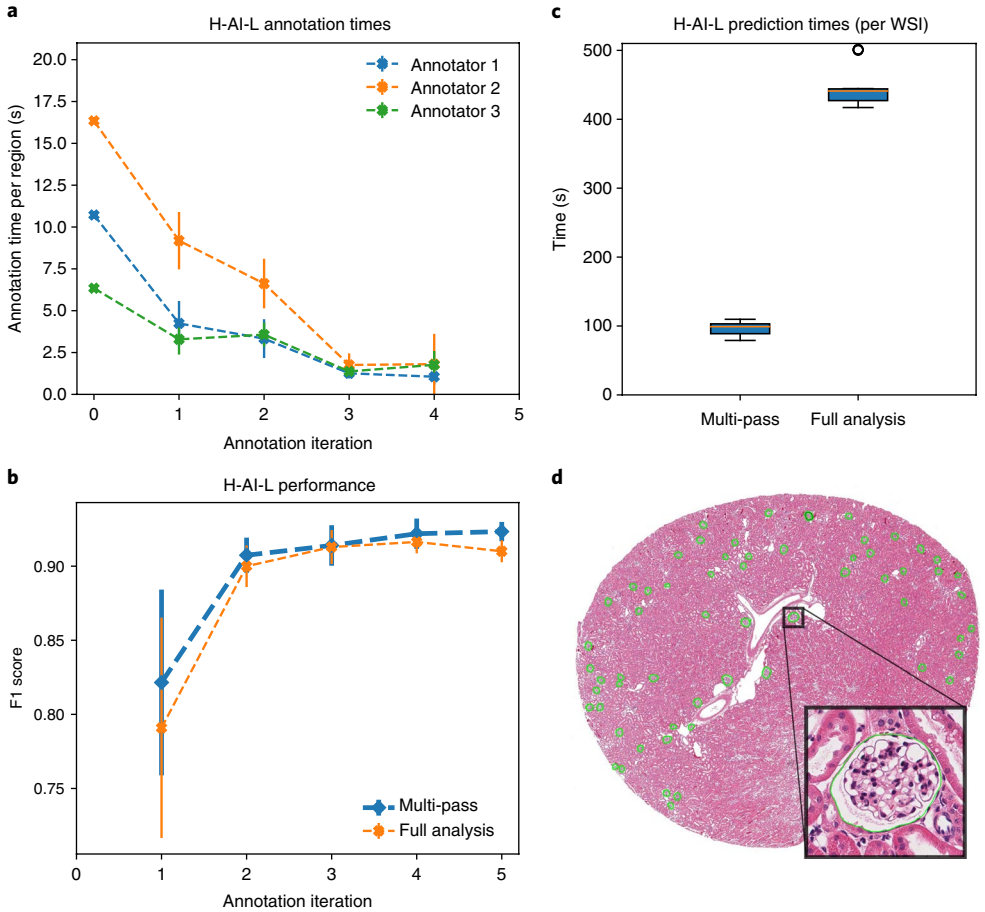


Fig. 2 | H-AI-L pipeline performance analysis for glomerular segmentation on holdout mouse WSIs. a, Average annotation time per glomerulus as a function of annotation iteration. The data are averaged per WSI and normalized by the number of glomeruli in each WSI. The 0th iteration was performed without pre-existing predicted annotations, whereas subsequent iterations use network predictions as an initial annotation prediction that can be corrected by the annotator. **b**, F1 score of glomerular segmentation of four holdout mouse renal WSIs as a function of training iteration. **c**, Run times for glomerular segmentation prediction on holdout mouse renal WSIs using H-AI-L with multi-pass (two-stage segmentation) versus full-resolution segmentation. **d**, Example of a mouse WSI with segmented glomeruli (×40, H&E-stained). Network predictions are outlined in green. The error bars indicate ±1 standard deviation.

low-resolution network from Fig. 2 was kept unchanged to identify the glomerular regions in the mouse WSI.

Due to the non-sparse nature of IFTA regions in some human WSIs, we forgo our multi-pass approach to generate the results shown in Fig. 5. The development of this IFTA network has been limited due to the biological expertise required to produce these multi-class annotations. However, preliminary segmentation results on holdout WSIs are promising, even though only 15 annotated biopsies were used for training (Fig. 5). We note that this is a small training set, as human biopsy WSIs contain much less tissue area than the mouse kidney sections used to train the glomerular segmentation network above.

Finally, to show the adaptability of the H-AI-L pipeline to other medical imaging modalities, we quantify the use of our approach for the segmentation of human prostate glands from T2 MRI data. These data were oriented and normalized as described in ref. [45] and saved as a series of TIFF image files. These images can be opened in ImageScope and are compatible with our H-AI-L pipeline. This analysis was completed using a training set of data from 39 patients, with an average of 32 slices per patient (512×512 pixels) (Fig. 6d); 509 of the total 1,235 slices contained prostate regions of interest. Iterative training was completed by adding data from four new patients to the training set before each iteration. Data from the remaining seven patients were used as a holdout testing set (a full breakdown is available in Supplementary Table 1). The newly annotated/corrected training data were augmented ten times, and a full-resolution network was trained for two epochs during each iteration: the results of this training are presented in Fig. 6. While the network performs well after just one round of training, the performance on holdout patient data continues to improve with the addition of training data (Fig. 6a), achieving a sensitivity of 0.88 ± 0.04, specificity of 0.99 ± 0.001, precision of 0.9 ± 0.03 and accuracy of 0.99 ± 0.001. This trend is also loosely reflected in the network prediction on newly added training data, where an upward trend in prediction performance is observed in Fig. 6b. Notably, when our iterative training pipeline is applied to this dataset, annotation is reduced by approximately 90% percent after the second iteration; only 10% of the MRI slices containing prostate fall below our segmentation performance threshold (Fig. 6c). We note that careful conversion between the DICOM and TIFF format (considering orientation and colour scaling) is essential for this analysis.

Conclusions

We have developed an intuitive pipeline for segmenting structures from WSIs commonly used in pathology, a field where there is often a large disconnect between domain experts and engineers.

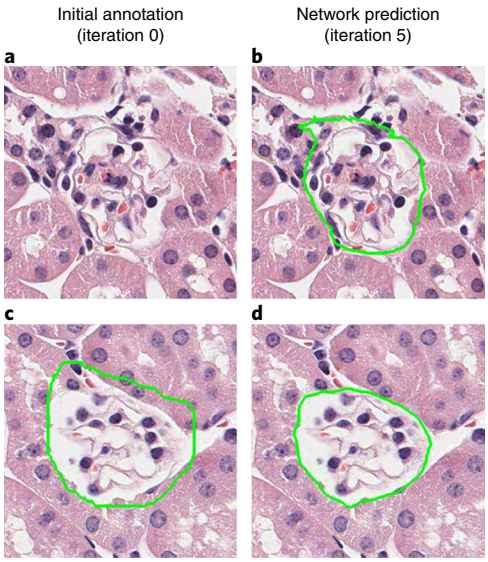


Fig. 3 | H-AI-L human annotation errors (mouse data). a–d, Comparison of initial manual annotations from iteration 0 (**a**,**c**) with their respective final network predictions from iteration 5 (**b**,**d**). These examples were selected due to poor manual annotation, where the glomerulus was not annotated (**a**) or showed poorly drawn boundaries (**c**). These images are captured at ×40, and tissue was stained using H&E.

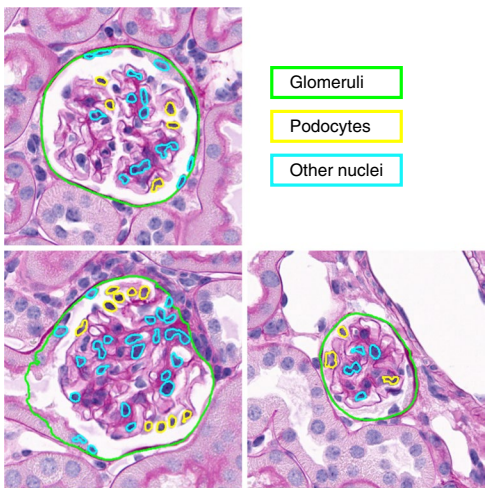


Fig. 4 | Multiclass nuclei prediction on a mouse WSI. Several examples of multi-class nuclei predictions are visualized on a mouse WSI (×40, PAS-stained). Here, transfer learning was used to adapt the high-resolution network from above (Fig. 2) to segment nuclei classes. This network was trained using 143 labelled mouse glomeruli. The low-resolution network was kept unchanged for the initial detection of glomeruli. We expect the results to significantly improve using more labelled training data.

To bridge this gap, we seek to provide pathologists with robust data analytics provided by state-of-the-art neural networks. We have developed an intuitive library for the adaptation of DeepLab v2[25], a semantic segmentation network, to WSI data commonly used in the field. This library uses annotation tools from the common WSI viewing software Aperio ImageScope[26] to annotate and display network predictions. Training, prediction and validation of the network are performed via a single Python script with a command line interface, making data management as simple as dropping data into a pre-determined folder structure.

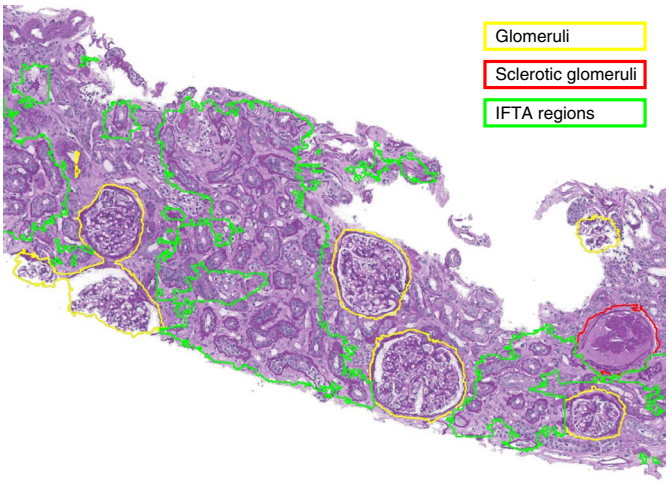


Fig. 5 | Multiclass IFTA prediction on a holdout human renal WSI. Segmentation of healthy and sclerotic glomeruli, as well as IFTA regions from human renal biopsy WSI (×40, PAS-stained). Due to the non-sparse nature of IFTA regions, these predictions were made using only a high-resolution pass. This is a screenshot of Aperio ImageScope, which we use to interactively visualize the network predictions.

Our iterative, human-in-the-loop training allows considerably faster annotation of new WSIs (or similar imaging data), because network predictions can easily be corrected in ImageScope before incorporation into the training set. With this approach, network performance can be qualitatively assessed after each iteration. Newly added data act as a holdout validation set, where predictions are easily viewed during correction. The theoretical performance achievable by this method is bounded by the training set used, and is therefore the same as the current state-of-the-art (manual annotation of all training data). However, due to the increased speed of annotation and the intuitive visualization of network performance (allowing selection of poorly predicted new data after each iteration), H-AI-L training can converge to the upper bound of performance more efficiently than the traditional method. That is, H-AI-L achieves state-of-the-art segmentation performance much faster than traditional methods, which are limited by data annotation speed (Fig. 7). Our H-AI-L approach offers an ideal viewing environment for network predictions on WSIs, using the fast pan and zoom functionality provided by ImageScope[27], improving the accuracy and ease of expert annotation.

The ability to transfer parameters from a trained network (repurposing it for a different task) ensures that segmentation of tissue structure can be tailored to any clinical or research definition, including other biomedical imaging modalities. Our two-stage segmentation (multi-pass) analysis allows rapid prediction of sparse regions from large WSIs, without sacrificing accuracy due to low-resolution analysis alone. Inspired by the way pathologists scan tissue slides, multi-pass approaches have been successfully described in digital pathology for detecting cell nuclei[46]. We believe that this technique offers the perfect compromise between speed and specificity, producing high-resolution sparse segmentations ideal for display in ImageScope. Our method provides non-sparse segmentation of WSIs by forgoing multi-pass analysis. However, in the future we plan to change how the class hierarchy is defined in our algorithm, offering easy functionality to search for low-resolution regions with high-resolution sub-compartments.

In the future, we will also extensively test our method in a clinical research setting. This testing will evaluate both the segmentation performance and ergonomic aspects affecting a clinician's ease of use. We will extend our method to provide anomaly detection,

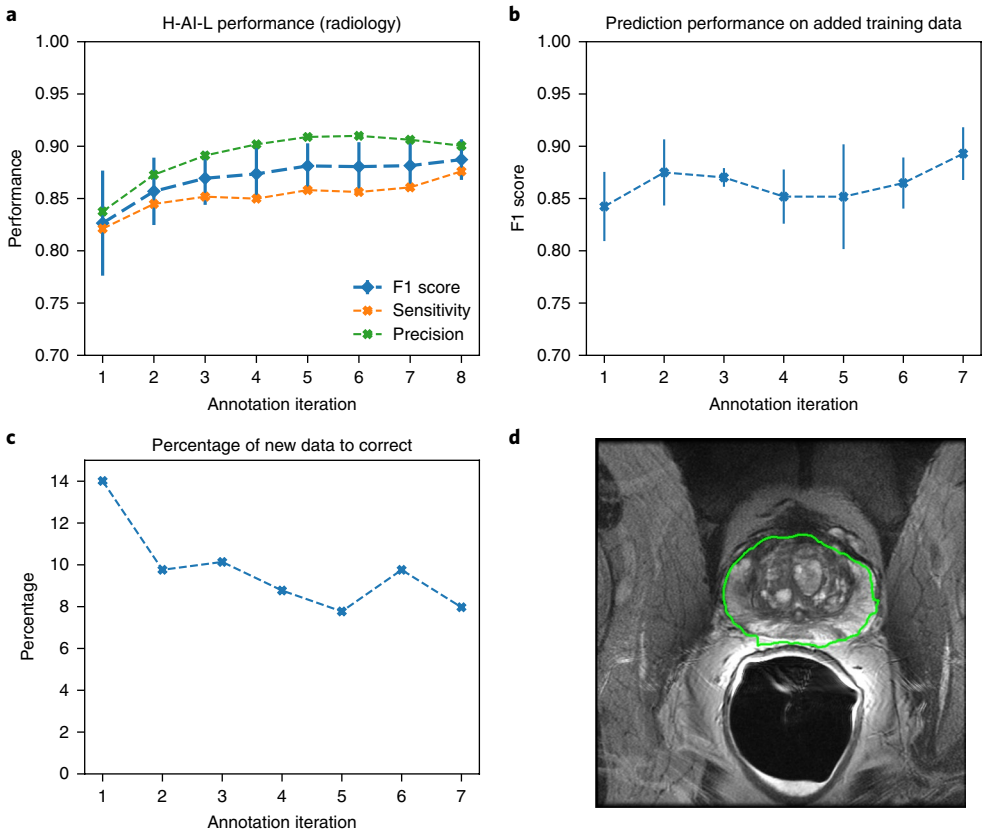


Fig. 6 | H-AI-L method performance analysis for human prostate segmentation from T2 MRI slices. a, Segmentation performance as a function of training iteration, evaluated on 7 patient holdout MRI images (224 slices). Performance was evaluated on a patient basis. We note that despite the decline in network precision after iteration 6, the F1 score improves as a result of increasing sensitivity. **b**, The prediction performance on added training data, before network training. This figure shows the prediction performance on newly added data with respect to the expert-corrected annotation, and is evaluated on a patient basis (data from four new patients were added at the beginning of each training iteration). **c**, The percentage of prostate regions where network prediction performance (F1 score) fell below an acceptable threshold (percentage of slices that needed expert correction) as a function of training iteration. We define acceptable performance as F1 score > 0.88. Using this criterion, expert annotation of new data is reduced by 92% by the fifth iteration. **d**, A randomly selected example of a T2 MRI slice with segmented prostate; the network predictions are outlined in green. The error bars indicate ±1 standard deviation. A detailed breakdown of the training and validation datasets is available in Supplementary Table 1.

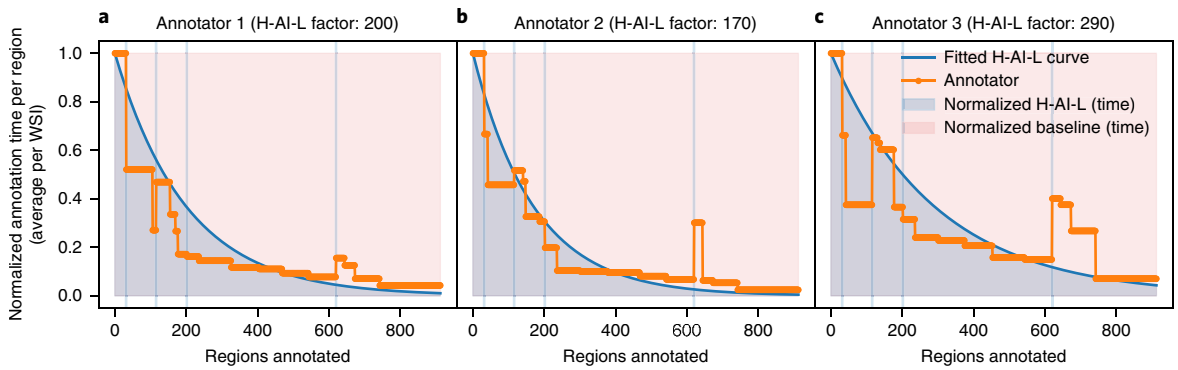


Fig. 7 | Annotation time-savings using the H-AI-L method while comparing to baseline segmentation speed. H-AI-L plots showing the annotation time per region normalized with respect to the baseline annotation speed of each annotator for the result shown in Fig. 2a. An exponential decay distribution (H-AI-L curve) is fitted to each annotator, where the H-AI-L factor is the exponential time constant: a derivation can be found in the Methods. The vertical lines are gaps between iterations (where the network was trained). The area under the H-AI-L curve represents the normalized annotation time per annotator. This can be compared to the area of the normalized baseline region, which represents the normalized annotation time without the H-AI-L method. **a**, The time-savings by annotator 1 (calculated to be 81.3%) when creating the training set used to train the glomerular segmentation network in Fig. 2. **b**, Annotator 2 was 82.0% faster. **c**, Annotator 3 was 72.7% faster. While the *y* axis in these plots is not a direct measure of network performance, it is highly correlated. The spike in annotation time seen at 600 regions is data from a WSI with severe glomerular damage from diabetic nephropathy. Future work will involve deriving optimal iterative training strategies based on information mined via such plots, with a goal of reducing annotation burdens for expert annotators.

defining a confidence metric and threshold where WSIs are flagged for further evaluation. Further, to minimize the expert's time, we will create an algorithm to predict the optimal amount of annotation performed in each iteration, using a curve fitting similar to Fig. 7. We will also adapt our method for native use with a DICOM viewer and a three-dimensional CNN for segmentation, allowing easier workflows for segmentation of radiology datasets, and mitigating the issues of data orientation and gamut mapping when converting to 8-bit TIFF images. Given these tools, we foresee a segmentation approach similar to our H-AI-L method underpinning efforts to build searchable medical image databases for research and education.

Methods

All animal tissue sections were collected in accordance with protocols approved by the Institutional Animal Care and Use Committee at the University at Buffalo, and in a manner consistent with federal guidelines and regulations and in accordance with recommendations of the American Veterinary Medical Association guidelines on euthanasia. Human renal biopsy samples were collected from the Kidney Translational Research Center at Washington University School of Medicine, directed by S.J., following a protocol approved by the Institutional Review Board at the University at Buffalo before commencement. Digital MRI images of human prostate glands were provided by P.S.L., following a protocol approved by the Institutional Review Board at the Medical College of Wisconsin. All human methods were performed in accordance with the relevant federal guidelines and regulations. All patients provided written informed consent.

For mouse pathology sample preparation, C57BL/6J background mice were euthanized, and their kidneys were perfused, extracted and embedded in paraffin. Mice were either treated with STZ to induce diabetic nephropathy or with an STZ vehicle for control. The murine WSIs used (Figs. 2 and 3) were sliced from paraffin-embedded kidney sections at 2 µm, stained with either PAS or H&E, and bright-field imaged at 0.25 µm per pixel resolution and ×40 magnification using a whole-slide scanner (Aperio Scan Scope, Leica). The sections used for podocyte segmentation (Fig. 4) were prepared similarly: stained first using immunofluorescence labels targeting WT1 (to generate training labels for podocyte detection), and then imaged via a whole-slide fluorescence scanner at 0.16 µm per pixel resolution and ×40magnification (Aperio Versa, Leica). These tissue sections were then post-stained using PAS, and bright-field imaged as described above. The human pathology WSIs used (Fig. 5) were obtained from 2–5-µm-thick biopsy sections, stained with PAS and bright-field imaged in a manner similar to that discussed above.

For digital MRI images of human prostate glands, 39 patients were recruited for an MRI scan befre a radical prostatectomy, using a 3T GE scanner (GE Healthcare) and an endorectal coil. The MRI included an axial T2-weighted image, collected with 3 mm slice thickness, 0.234 × 0.234 mm^2 voxel resolution, and a 4,750/123 ms TR/TE. The DICOM files were converted to NIFTI format using the mri_convert command from the Freesurfer library of tools (surfer.nmr.mgh.harvard.edu). Prostate masks were then manually annotated using AFNI by P.S.L. and verified by a board-certified radiologist for an unrelated study[47]. The prostate images and annotations were then converted into TIFF format using MATLAB (Mathworks Inc) for analysis by the SUNY Buffalo team.

In the H-AI-L pipeline, an annotator labels a limited number of WSIs using annotation tools in ImageScope[26], which provides the input for network training. The resulting trained network is then used to predict the annotations on new WSIs. These predictions are used as rough annotations, which are corrected by the annotator and sent back for incorporation into the training set; improving network performance and optimizing the amount of expert annotation time required. As this technique makes the adaptation of network parameters to new data easy, adapting a trained network to new data generated in different institutions is extremely feasible.

At the heart of H-AI-L is the conversion between mask and XML[48] formats, which are used by DeepLab v2[25] and ImageScope[26], respectively. Training any semantic segmentation architecture relies on pixel-wise image annotations that are input to the network for training and output after network predictions as mask images. In the case of DeepLab, the mask images take the form of indexed greyscale 8-bit PNG files, where each unique value pertains to an image class. On the other hand, annotations performed in ImageScope are saved in text format, as XML files[48], where each region is saved as a series of boundary points or vertices. Determining the vertices of a mask image is a common image processing task, known as image contour detection[49,50]. As opposed to edge detection, contour detection can have hierarchal classifications[50], lending itself ideally to conversion into the hierarchal XML format used by ImageScope.

To facilitate the transfer between ImageScope XML and greyscale mask images, we use the OpenCV-Python library (cv2)[49], specifically the function cv2.findContours to convert from masks to contours. Using this function, we are able to automatically convert DeepLab predictions to XML format, which can be viewed in ImageScope, and thus easily evaluate and correct network performance. Furthermore, we have written a library for converting an XML file into mask regions, using cv2.fillPoly. This library follows the OpenSlide-Python[51] conventions for reading WSI regions, returning a specified mask region from the WSI.

Using OpenSlide[51] and our XML to mask libraries allows for efficient chopping of WSIs into overlapping blocks for network training and prediction; similar sliding-window approaches are common in predicting semantic segmentations on large medical images[52,53]. To simplify the iterative training process, and complement the easy annotation pipeline proposed, we have created a callable function that handles operations automatically, prompting the user to initiate the next step. This function needs two flags [--option] and [--project], which are the parameters identifying the iterative step and the project to train, respectively. Initially created using [--option] 'new', a new project is trained iteratively by alternating the [--option] flag between 'train' and 'test'.

Multi-pass approach. Our algorithm uses our multi-pass approach by default. This approach is inspired by the way that pathologists scan WSIs at progressively higher resolutions. This process is accomplished by training two DeepLab segmentation networks using image regions and masks cropped from the training set. A high-resolution and a separate low-resolution network are respectively trained with full-resolution and down-sampled cropped regions. Prediction using this approach is performed serially; the low-resolution network identifies WSI regions to be passed to the high-resolution network for further refinement. This method is outlined in Supplementary Fig. 1.

Full-resolution analysis alone is achievable by setting the [--one_network] flag to 'True' during training and prediction. This analysis trains only the high-resolution network, which is exclusively used to segment WSIs during prediction. More information on the training and prediction is explained below.

Training. To streamline the training process, we created a pipeline where a user places new WSIs and XML annotations in a project folder structure, and then calls a function to train the project. This automatically initiates data chopping and augmentation, and then loads parameters from the most recently trained network (if available) before starting to train. For faster convergence, we utilize transfer learning, automatically pulling a pre-trained network file whenever a new project is created, which is used to initialize the network parameters before training. We have also included functionality to specify a pre-trained file from an existing project using the [--transfer] flag. For ease of use, the network hyper-parameters can be changed using command line flags, but are set automatically by default.

When [--option] 'train' is specified, WSIs and XML annotations are chopped into a training set containing 500 × 500 blocks with 50% overlap. This training set is then augmented via random flipping, hue and lightness shifts, and piecewise affine transformations, all accomplished using the imgaug Python library[54]. To keep the network unbiased, the total number of blocks containing each class is tabulated and used to augment less frequent classes with a higher probability[55]. Our multi-pass approach performs these steps for both high- and low-resolution patches separately to generate two training sets. The 500 × 500 low-resolution patches cover a greater receptive field, emphasizing information that occurs in the lower spatial image frequencies.

Once the training data have been assembled, the networks are trained for the specified number of epochs. The user is then prompted to upload new WSIs and run the [--option] 'predict' flag. This produces XML predictions that can be corrected using ImageScope before incorporation into the training set.

Multi-pass prediction. Due to the sparse nature of the structures we attempt to segment from renal WSIs, we limit the search space, using a low-resolution pass to determine hotspot regions before segmentation at full resolution. In this multi-pass approach, thresholding and morphological processing first determine which WSI blocks contain tissue, eliminating background regions. Second, down-sampled blocks (1/16th resolution, 500 × 500 pixels with 50% overlap) are extracted and tested, using the low-resolution segmentation network to roughly segment structures. The output predictions of the preprocessing steps are then stitched back into a hotspot map, which is 1/16th the WSI size. For multi-class cases, this stitching can be performed by finding the maximum class number between overlapping prediction maps, which is assigned to each pixel in the hotspot map. In this way, multi-class hierarchies are defined by assigning subclasses to higher mask indices. For example, conducting the stitching for the nuclear segmentation in Fig. 4 requires the definition of background, glomeruli, nuclei and podocyte classes to be 0, 1, 2 and 3, respectively, where nuclei and podocytes are compartments of glomeruli. The result in Fig. 5 was obtained using a similar procedure. This stitching operation is outlined in Supplementary Fig. 2 for two classes. The results in Figs. 2, 3 and 6 were obtained using a similar two-class stitching operation.

The hotspot map is then used to determine the locations for performing pixel-wise segmentation using the high-resolution DeepLab network (trained using full-resolution image patches). Hotspot indices are calculated, scaled back to full resolution (×16), and used to extract these regions at full resolution. The XML annotation file is then assembled from the high-resolution predictions on these regions.

Full-resolution prediction. When the [--one_network] flag is set to *'True'*, the initial extraction of overlapping blocks is performed at full resolution. Prediction

on these blocks uses the high-resolution DeepLab network, and the resulting hotspot map is stitched using the same method as above. Unlike above, this map (which is the same size as the WSI) is used to directly assemble the XML annotation file.

Post prediction processing. To limit possible false-positive predictions of small regions, we implemented a size threshold that tests the area of each predicted region, eliminating regions smaller than the set threshold using morphological operations. This threshold can be adjusted via the [--min_size] flag, and is easily estimated using the area displayed in the Annotations tab in ImageScope to determine the minimum regions size. By default, this threshold is set to 625 pixels, which was used for the analysis in this paper.

Validation. While the performance of the network is easily visualized after prediction on new WSIs, we have included functionality for explicitly evaluating performance metrics and prediction time on a holdout dataset. This is accomplished using the [--option] 'validate' flag. When called, it evaluates the network performance on holdout images for every annotation iteration by automatically pulling the latest models. To perform this performance comparison, ground-truth XML annotations of the holdout set are required to calculate the sensitivity, specificity, accuracy and precision performance metrics[38].

Estimating H-AI-L performance (Fig. 7). To quantify the time-savings of our H-AI-L method, we plot the normalized annotation time per region versus the number of regions annotated. Here we define the normalized annotation time per region A as $A=\frac{t}{t_0}$, where t is the annotation time per region (averaged per WSI) and t_0 is the average annotation time per region in iteration 0. A is bounded from [0,1], where 1 is the normalized time required to annotate one region fully. Although the annotation time is reduced as a piecewise function of the training iteration, in Fig. 7 we use a continuous exponential decay distribution to approximate $A(r)$:

$A(r)=e^{\frac{-r}{\tau}}$, where r is the number of regions annotated and τ is the exponential time constant, which we call the H-AI-L factor.

The normalized annotation time of our H-AI-L method (H) can therefore be estimated as

$$H=\int_0^R A(r)\mathrm{d}r=\tau[1-e^{\frac{-R}{\tau}}]$$

where R is the total number of regions annotated. Likewise, the normalized baseline annotation time (B) can be calculated as

$$B=\int_0^R 1\mathrm{d}r=R$$

Therefore, the time-savings performance (P) of our H-AI-L method can be estimated as a percentage:

$$P=\left(1-\frac{H}{B}\right)\times 100=\left(1+\frac{\tau}{R}[e^{\frac{-R}{\tau}}-1]\right)\times 100$$

The H-AI-L factor τ reflects the effectiveness of iterative network training, where lower values of τ represent training curves that decay faster. In the future, algorithms to select the optimal amount of annotation and identify data outliers to be annotated at each iteration will improve the performance of the H-AI-L method by reducing τ.

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

We have made the data used for analysing the performance of H-AI-L method available at https://goo.gl/cFVxjn. The folder contains a detailed note describing the data. Namely, the folder contains pathology and radiology image data used for training and testing our H-AI-L method, ground-truth and predicted segmentations of the test image data, network corections and respective annotations of the training image data for different iterations, and the network models trained at different iterations. We have made our code openly available online at https://github.com/SarderLab/H-AI-L.

Received: 12 October 2018; Accepted: 7 January 2019;
Published online: 11 February 2019

References

1. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60**, 84–90 (2017).
2. LeCun, Y. & Bengio, Y. in *The Handbook of Brain Theory and Neural Networks* (ed. Michael, A. A.) 255–258 (MIT Press, Cambridge, 1998).
3. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
4. Pedraza, A. et al. Glomerulus classification with convolutional neural networks. In *Proc. Medical Image Understanding and Analysis: 21st Annual Conference, MIUA 2017* (eds Valdés Hernández, M. & González-Castro, V.) 839–849 (Springer, 2017).
5. Schmidhuber, J. Deep learning in neural networks: an overview. *Neural Netw.* **61**, 85–117 (2015).
6. Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proc. COMPSTAT'2010* (eds Lechevallier, Y. & Saporta, G.) 177–186 (Springer, 2010).
7. Szegedy, C. et al. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2015).
8. Swingler, K. *Applying Neural Networks: A Practical Guide* (Morgan Kaufmann, Burlington, 1996).
9. Ronneberger, O., Fischer, P. & Brox, T. U-net: convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (eds Navab, N., Hornegger, J., Wells, W. M. & Frangi, A. F.) (Springer, 2015).
10. Zhang, T. & Nakamura, M. Neural network-based hybrid human-in-the-loop control for meal assistance orthosis. *IEEE Trans. Neural Syst. Rehabil. Eng.* **14**, 64–75 (2006).
11. Krogh, A. & Vedelsby, J. in *Advances in Neural Information Processing Systems* (1995).
12. Cohn, D., Atlas, L. & Ladner, R. Improving generalization with active learning. *Mach. Learn.* **15**, 201–221 (1994).
13. Gosselin, P. H. & Cord, M. Active learning methods for interactive image retrieval. *IEEE Trans. Image Process.* **17**, 1200–1211 (2008).
14. Shi, L. & Wang, X.-c. Artificial neural networks: current applications in modern medicine. In *Computer and Communication Technologies in Agriculture Engineering, 2010 International Conference* (IEEE, 2010).
15. Madabhushi, A. & Lee, G. Image analysis and machine learning in digital pathology: challenges and opportunities. *Med. Image Anal.* **33**, 170–175 (2016).
16. Baxevanis, A. D. & Bateman, A. The importance of biological databases in biological discovery. *Curr. Protoc. Bioinformatics* **50**, 1.1.1-8 (2015).
17. Cheplygina, V. et al. in *Deep Learning and Data Labeling for Medical Applications* 209–218 (Springer, New York, 2016).
18. Szolovits, P., Patil, R. S. & Schwartz, W. B. Artificial intelligence in medical diagnosis. *Ann. Intern. Med.* **108**, 80–87 (1988).
19. Orthuber, W. et al. Design of a global medical database which is searchable by human diagnostic patterns. *Open Med. Inform. J.* **2**, 21 (2008).
20. Smeulders, A. W. et al. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 1349–1380 (2000).
21. Müller, H. et al. A review of content-based image retrieval systems in medical applications—clinical benefits and future directions. *Int. J. Med. Inform.* **73**, 1–23 (2004).
22. Gong, T. et al. *Automatic pathology annotation on medical images: a statistical machine translation framework*. In *Proc. 20th International Conference on Pattern Recognition* (IEEE, 2010).
23. Abe, N., Zadrozny, B. & Langford, J. *Outlier detection by active learning*. In *Proc. 12th ACM SIGKDD International Conference on Knowledge discovery and Data mining* (ACM, 2006).
24. Doyle, S. & Madabhushi, A. *Consensus of Ambiguity: Theory and Application of Active Learning for Biomedical Image Analysis* (Springer, Berlin, 2010).
25. Chen, L.-C. et al. Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**, 834–848 (2018).
26. Aperio Imagescope (Leica Biosystems); https://www.leicabiosystems.com/digital-pathology/manage/aperio-imagescope/
27. Skodras, A., Christopoulos, C. & Ebrahimi, T. The JPEG 2000 still image compression standard. *IEEE Signal Process. Mag.* **18**, 36–58 (2001).
28. Sedeen Viewer (Pathcore); https://pathcore.com/sedeen/
29. Ginley, B., Tomaszewski, J. E. & Sarder, P. Automatic computational labeling of glomerular textural boundaries. In *Proc. SPIE 10140, Medical Imaging 2017: Digital Pathology* 101400G (2017).
30. Kato, T. et al. Segmental HOG: new descriptor for glomerulus detection in kidney microscopy image. *BMC Bioinformatics* **16**, 316 (2015).
31. Sarder, P., Ginley, B. & Tomaszewski, J. E. Automated renal histopathology: digital extraction and quantification of renal pathology. In *Proc. SPIE 9791, Medical Imaging 2016: Digital Pathology* 97910F (2016).
32. Simon, O., Yacoub, R., Jain, S., Tomaszewski, J. E. & Sarder, P. Multi-radial LBP features as a tool for rapid glomerular detection and assessment in whole slide histopathology images. *Sci. Rep.* **8**, 2032 (2018).
33. Tesch, G. H & Allen, T. J. Rodent models of streptozotocin-induced diabetic nephropathy. *Nephrology* **12**, 261–216 (2007).
34. Goyal, S. N. et al. Challenges and issues with streptozotocin-induced diabetes - a clinically relevant animal model to understand the diabetes pathogenesis and evaluate therapeutics. *Chem. Biol. Interact.* **244**, 49–63 (2016).
35. Kitada, M., Ogura, Y. & Koya, D. Rodent models of diabetic nephropathy: their utility and limitations. *Int. J. Nephrol. Renov. Dis.* **9**, 279–290 (2016).
36. Wu, K. K. & Huan, Y. Streptozotocin-induced diabetic models in mice and rats.*Curr. Protoc. Pharmacol.* **40**, 5.47 (2008).

37. Hripcsak, G. & Rothschild, A. S. Agreement, the F-measure, and reliability in information retrieval. *J. Am. Med. Inform. Assoc.* **12**, 296–298 (2005).
38. Sokolova, M., Japkowicz, N. & Szpakowicz, S. Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. In *Australasian Joint Conference on Artificial Intelligence* (eds Sattar, A. & Kang, B.-H.) (Springer, 2006).
39. Japkowicz, N. & Stephen, S. The class imbalance problem: A systematic study. *Intell. Data Anal.* **6**, 429–449 (2002).
40. Bariety, J. et al. Parietal podocytes in normal human glomeruli. *J. Am. Soc. Nephrol.* **17**, 2770–2780 (2006).
41. Pavenstadt, H., Kriz, W. & Kretzler, M. Cell biology of the glomerular podocyte. *Physiol. Rev.* **83**, 253–307 (2003).
42. Solez, K. et al. Banff 07 classification of renal allograft pathology: updates and future directions. *Am. J. Transplant.* **8**, 753–760 (2008).
43. Mengel, M. Deconstructing interstitial fibrosis and tubular atrophy: a step toward precision medicine in renal transplantation. *Kidney Int.* **92**, 553–555 (2017).
44. Wang, X. et al. Glomerular pathology in dent disease and its association with kidney function. *Clin. J. Am. Soc. Nephrol.* **11**, 2168–2176 (2016).
45. McGarry, S. D. et al. Radio-pathomic maps of epithelium and lumen density predict the location of high-grade prostate cancer. *Int. J. Radiat. Oncol. Biol. Phys.* **101**, 1179–1187 (2018).
46. Janowczyk, A. et al. A resolution adaptive deep hierarchical (RADHicaL) learning scheme applied to nuclear segmentation of digital pathology images. *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.* **6**, 270–276 (2016).
47. McGarry, S. D. et al. Radio-pathomic maps of epithelium and lumen density predict the location of high-grade prostate cancer. *Int. J. Radiat. Oncol. Biol. Phys.* **101**, 1179–1187 (2018).
48. Bray, T. et al. Extensible markup language (XML). *World Wide Web J.* **2**, 27–66 (1997).
49. Bradski, G. The OpenCV Library. *Dr. Dobb's* http://www.drdobbs.com/open-source/the-opencv-library/184404319 (2000).
50. Klette, R. et al. *Computer Vision* (Springer, New York, 1998)
51. Goode, A. et al. OpenSlide: a vendor-neutral software foundation for digital pathology. *J. Pathol. Inform.* **4**, 27 (2013).
52. Lu, C. & Mandal, M. Automated segmentation and analysis of the epidermis area in skin histopathological images. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (IEEE, 2012).
53. Govind, D. et al. Automated erythrocyte detection and classification from whole slide images. *J. Med. Imaging* **5**, 027501 (2018).
54. Jung, A. imgaug (2017); http://imgaug.readthedocs.io/en/latest/
55. Zhou, Z.-H. & Liu, X.-Y. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Trans. Knowl. Data Eng.* **18**, 63–77 (2006).

Acknowledgements

The project was supported by the faculty start-up funds from the Jacobs School of Medicine and Biomedical Sciences, University at Buffalo, the University at Buffalo IMPACT award, NIDDK Diabetic Complications Consortium grant DK076169 and NIDDK grant R01 DK114485. The prostate imaging data were collected with funds from the State of Wisconsin Tax Check-off Program for Prostate Cancer research. Percent efforts for P.S.L. and S.D.M. were provided by R01 CA218144, and the National Center for Advancing Translational Sciences NIH UL1TR001436 and TL1TR001437. We thank NVIDIA Corporation for the donation of the Titan X Pascal GPU used for this research.

Author contributions

B.L. conceived the H-AI-L method, analysed the data and wrote the manuscript. The code was written by B.L. and B.G. D.G. contributed in generating results for Fig. 4. S.D.M. and P.S.L. provided the radiology data and annotations for the prostate MRI analysis, and edited the manuscript. R.Y. implemented the mouse model. S.J. provided human renal biopsy data. J.E.T. evaluated renal pathology segmentation as a domain expert. K.-Y.J. provided the IFTA annotation for Fig. 5. P.S. is responsible for the overall coordination of the project, mentoring and formalizing the image analysis concept and oversaw manuscript preparation.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s42256-019-0018-3.

Reprints and permissions information is available at www.nature.com/reprints.

Correspondence and requests for materials should be addressed to P.S.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature Limited 2019

nature research

Corresponding author(s): Pinaki Sarder

Last updated by author(s): Dec 24, 2018

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see Authors & Referees and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☒	☐	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	☐	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Leica digital pathology scanners were used for the collection of the digital whole slide images (WSI) of the histology specimens used in this study. Digital MRI image collection was standard and is described in the Methods section of the manuscript.
Data analysis	Aperio Imagescope version 12.3.3 (Leica) was used for digital image annotation and viewing. The Openslide python library was used to view digital images in python. The H-AI-L code is written in python 3. The semantic segmentation network, DeepLab v2 was used for this work.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors/reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

We have made the data used for the performance analysis available here: https://goo.gl/cFVxjn. We have made our code openly available online: https://github.com/SarderLab/H-AI-L

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	NA
Data exclusions	No data were excluded
Replication	Training of neural networks with large datasets takes time, which limited the possibilities for replication. However, during development, we were able to obtain similar performance to that presented in this manuscript using random testing experiments.
Randomization	The samples were randomly assigned to training and holdout testing sets for all analysis done in this manuscript. During training the inclusion of new training data was done by selecting WSIs where the network performed poorly (done to optimize the learning speed).
Blinding	Experts performing annotation for this study were not given information on the disease or condition of the data they were given. This was done to ensure unbiased annotation performance.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology
☐	☒ Animals and other organisms
☐	☒ Human research participants
☒	☐ Clinical data

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Animals and other organisms

Policy information about studies involving animals; ARRIVE guidelines recommended for reporting animal research

Laboratory animals	Mouse, C57BL/6J background, Male, 7-32 weeks of age.
Wild animals	NA
Field-collected samples	NA
Ethics oversight	Institutional Animal Care and Use Committee at University at Buffalo

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Human research participants

Policy information about studies involving human research participants

Population characteristics	18-70 years old. Special populations (vulnerable) such as minors, pregnant women, neonates, prisoners, children, and cognitively impaired patients were not included. There are no restrictions in regards to gender or race. The renal tissue samples were obtained at Kidney Translational Research Center of Washington University School of Medicine under their institutionally approved protocol. Tissue samples were provided to the corresponding author and the research team after de-identification under an institutionally approved protocol at University at Buffalo. The human prostate MRI digital images were provided by co-author Peter S. Laviolette to the research team after de-identification under an institutionally approved protocol at Medical

	College of Wisconsin.
Recruitment	NA. Digital images were used in a de-identified fashion for developing the computational image annotation pipeline. For achieving robustness, several images from several cases were needed. However, any specific recruitment strategy does not impact the computational result produced.
Ethics oversight	Washington University School of Medicine, University at Buffalo, Medical College of Wisconsin.

Note that full information on the approval of the study protocol must also be provided in the manuscript.

A galaxy of data challenges

By organizing Kaggle competitions, astrophysicist Thomas Kitching can focus on asking the right questions.

In science we always have more questions than answers. Finding the right tools to tackle these questions is an essential part of research. This challenge has recently taken on a new shape as many fields transition from data-scarce to data-rich paradigms and individual researchers realize they do not possess the tools to analyse large heterogeneous datasets. One strategy is to pursue interdisciplinary research and collaborations. Another is to seek the wisdom of the crowd.

This is the approach that we took by engaging with Kaggle, the data science platform. Kaggle runs competitions in which organizers upload a dataset and pose a problem or question based on that data. Organizers define a metric for scoring submissions, and provide training data for competitors to play around with and train their models. A prize is also offered, which can be anything from cash to a job interview.

From the organizer's perspective, Kaggle is a source for myriads of data scientists worldwide who can try to solve your data science problem. The key is to ask the right question, and provide the right prize. From the competitor's perspective, you can get access to new datasets, interact with other data scientists, and potentially get rich or land a new job.

We have run three Kaggle competitions since 2010 that have posed the problems of measuring galaxy shapes[1], determining the position of dark matter in noisy data[2] and classifying galaxy types[3], and there is currently a competition to help classify supernovae observations[4]. Running public competitions to spur scientific advances is not a new idea, and has a long heritage dating back to at least the Longitude Prize — a challenge set in 1713 to accurately determine the longitudinal position of a ship, with a maximum prize of up to £20,000 (equivalent to over £2 million in 2018). Fast-forward to the twenty-first century and such competitions thrive by the Internet, through which a nearly unlimited number of potential competitors can be instantly reached. Furthermore, Kaggle competitions are software rather than hardware based,

Credit: ESA/Hubble and NASA

and so solutions can be tested, improved and combined, and evolve at a rapid rate, with no specialist equipment required.

It can be difficult for organizers to offer good prizes. However, we identified a new mode of impact by partnering with a company — Winton Capital Management — who sponsored monetary prizes for the competitions. In return we allowed them to offer job interviews for data science positions to winners of the competitions. This is a win-win-win scenario: astronomy wins because we get good competitors solving our problems, the sponsoring company wins because they can reduce recruitment costs by getting access to people who have proven their data science skills, and the competitors win because they get a chance to solve astronomy mysteries and earn a position on the leader board while (hopefully) having fun.

During our competitions, new ideas were revealed for successfully solving the problems, leading to scientific papers describing the results. While the exact implementation of these ideas needed refinement before applying to real data, the kernel of these new directions for analysis began on the Kaggle leader board. However, it was pointed out in one competition — in a not entirely tongue in cheek manner — that the best way to win would be to model not the data, but the person who created the competition, in order to determine what assumptions were made. Indeed, from my perspective as a competition setter this is an astute observation, and one that future challenges should carefully consider.

But is running competitions the best way to generate new ideas and advance science? Organizing a competition is stressful, and while the majority of competitors are collaborative some competitors get, well… competitive — which is not always a pleasant experience. More broadly, the approach of crowdsourcing scientific advances to a competition could be seen as a step towards a more market-driven approach to science, particularly when prizes are monetary in nature. In a survival of the fittest, almost everyone ends up a loser on the leader board. So perhaps it would be better to collaborate rather than compete, and Kaggle is now moving towards that model by encouraging people to upload and explore datasets together.

A broader question is whether crowdsourced competitions are a step towards a mode of science in which scientists only ever ask the questions, but rely on others to find the answers. Taking this further: we use human competitors now, but perhaps in the future AI competitors will answer the questions we set. This reminds me of Isaac Asimov's Multivac stories, where scientists are those who ask the right questions to an AI, which provides the answers.

But for the moment at least, by admitting our limited knowledge and reaching out for help, these competitions provide a data playground in which problems can be solved and new friendships formed. ❐

Thomas Kitching
Mullard Space Science Laboratory, UCL, London, UK.
e-mail: t.kitching@ucl.ac.uk

Published online: 11 February 2019
https://doi.org/10.1038/s42256-018-0016-x

References

1. Mapping dark matter. *Kaggle* https://www.kaggle.com/c/mdm (2012).
2. Observing dark worlds. *Kaggle* https://www.kaggle.com/c/DarkWorlds (2013).
3. Galaxy zoo. *Kaggle* https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge (2014).
4. PLAsTiCC astronomical classification. *Kaggle* https://www.kaggle.com/c/PLAsTiCC-2018 (2018).

nature machine intelligence

AMENDMENTS

https://doi.org/10.1038/s42256-019-0023-6

Author Correction: Learnability can be undecidable

Shai Ben-David, Pavel Hrubeš, Shay Moran, Amir Shpilka and Amir Yehudayoff

Correction to: *Nature Machine Intelligence* https://doi.org/10.1038/s42256-018-0002-3, published online 7 January 2019.

In the version of this Article originally published, the following text was missing from the Acknowledgements: 'Part of the research was done while S.M. was at the Institute for Advanced Study in Princeton and was supported by NSF grant CCF-1412958.' This has now been corrected.

Published online: 23 January 2019
https://doi.org/10.1038/s42256-019-0023-6