

New methods for deprecating artificial intelligence systems will preserve history and facilitate research

Tim Johnson & Nick Obradovich

 Check for updates

AI developers and policy makers need to devise methods of commercial model deprecation that balance developers' reasonable interest in restricting access to outdated models with the value of preserving technological developments and enabling ongoing scientific research across versions.

Upon viewing a dynamo at the height of the Industrial Revolution, Henry Adams famously remarked that humanity had arrived, over a brief period of time, in "a new universe which had no common scale of measurement with the old"¹. For those observing advances in artificial intelligence (AI) via newspaper headlines, the past several years may feel a lot like Adams' era. The pace of change seems incomparably rapid.

Yet, just as the steam engine preceded the dynamo, yesterday's AI models have contributed to today's tools and—no matter how rapid the advance—one sees subtler changes looking iteratively across models than when comparing the oldest and youngest of the series. Gaining an accurate historical picture—not to mention an ability to replicate research and perform new comparative investigations—therefore requires ongoing access to older model versions. Current practices of commercial model deprecation, however, often remove all access to outdated models, obviating such possibilities. As a result, AI developers and policymakers need to collaborate on new methods of model deprecation that will balance developers' reasonable interest in restricting access to outdated models with the wider-ranging value of preserving recent technological developments and enabling ongoing scientific research across versions.

Here, we focus our discussion of deprecation methods on large language models (LLMs), due to their prevalent use, but readers will see that the arguments apply to other AI systems such as diffusion models deployed in image generation. Our arguments also do not conflict with previous important measures to preserve information about AI systems, such as the creation of model cards². Instead, we call for new deprecation methods that reinforce such measures by preserving history and facilitating research.

Reasons for current deprecation practices

At first glance, contemporary practices of model deprecation (or "retirement"), which cease all public access, seem hard to understand in light of the open-source ethos that figures prominently in the software development community. However, the rationale for current commercial deprecation practices appears both well-intentioned and understandable.

For one, even though the most up-to-date models can still produce problematic outputs exhibiting bias and other harms (e.g., privacy infringement via jailbreak techniques³), older models might lack safety features⁴ or legal safeguards that new models possess. As a result, protecting users from prejudicial, unsafe, and insecure models stands as a compelling rationale to deprecate outdated models.

Shielding users also protects model developers from liabilities that might come from old models spitting out libelous text or copyright materials. Since the release of popular LLMs such as ChatGPT, news reports have documented instances of individuals suing OpenAI for libelous content⁵, and academic studies have shown models producing text comparable to material under copyright⁶. Accordingly, developers act in a manner consistent with the tort system⁷ when they seek to avoid liability by removing risky models from public use.

Liabilities, however, do not amount to the only financial interest that motivates current practices of deprecation: maintaining outdated models imposes costs on developers that prove difficult to estimate and recoup. Whereas user fees can compensate readily for computing costs and customer support, enterprises likely find it difficult to calculate and charge for the opportunity costs associated with devoting hardware to outdated models or the reputational and brand-related costs resulting from users inadvertently obtaining lower-quality outputs from outdated models instead of using more-capable, newer models. Instead of facing the challenge of calculating and offsetting such costs, developers might find it more efficient just to remove access to older models.

Furthermore, developers might even find it desirable to deprecate models because the persistence of older models could make users unwilling to shift to newer models, thus preventing developers from fulfilling a business objective to supply the models they deem best or most profitable. Taken too far, this lattermost reason might become a cynical form of planned obsolescence—introducing new models to garner market share or attention—but it still resides within the developer's freedom to choose whether to keep their product in the marketplace.

In sum, current practices of model deprecation that restrict public access likely rest on well-meaning or understandable foundations. Older models may lack safeguards, create liabilities, impose tangible costs, and inhibit transitions to newer models that enterprises would prefer that users access. Together, these reasons provide a rationale for restricting public access to older models, yet these current deprecation practices also create problems.

Problems with current deprecation practices

Several challenges result from current LLM deprecation practices—from the mundane difficulty of transitioning applications from a particular model to the more substantial issue of losing the opportunity to perform historical, replicative, and comparative research. Exploring

these problems highlights the need for new deprecation methods that balance the concerns of developers with those of users.

To users who have built projects around soon-to-be-deprecated models, current practices of deprecation force a transition on an uncontrollable timeline. Users must assess whether advanced models will function in a manner comparable to older models in their applications; if projects remain viable across model iterations, users must adapt their projects to new models on a timeline that they do not create. Yet, although difficult, these challenges differ little from software users' perennial challenge of dealing with version updates. The distinctive issues related to the deprecation of LLMs mainly implicate the importance of outdated models for history and research.

Popular LLMs have gained users rapidly, thus making them significant artifacts of socio-cultural history. They also have exceeded the sophistication of past models to such an extent that they can be considered salient objects in the history of science and technology, if not the history of humanity. Maintaining this record becomes difficult when enterprises halt public access. Users can no longer observe, document, and catalog the precise attributes responsible for a model's place in history.

An inability to examine deprecated models directly also impedes research efforts. Researchers cannot characterize the evolution of model behavior over time if developers restrict access to a substantial number of past models⁸. Moreover, such restrictions impede scientists' ability to replicate past research involving deprecated models.

One response to this challenge is that researchers should not expect to replicate work using the same models. However, if research focuses on contrasting particular LLMs' outputs due to their model attributes, then one needs access to prior model versions to replicate that research. For example, our work has studied the text that LLMs generate when presented with settings in which they can dictate sharing behavior. We found that outputs consistently specifying sharing behavior first emerged in one particular model from OpenAI's GPT-3.5 series, text-davinci-003. After 4 January 2024, when OpenAI halted public access to text-davinci-003 and much of the GPT-3 series, other researchers lost the opportunity to replicate the process of generating our study data. That is, other researchers can inspect the data we collected and they can run our computer code on those data but they can no longer submit the prompts used in our study to text-davinci-003 and other deprecated LLM versions to observe whether the models generate outputs similar to those collected in our original study. Developing an archive of models, as we discuss later, can address this problem.

Current deprecation practices, however, not only impede the inspection of past research, but they also prevent new discoveries. Consider, for instance, research on whether⁹ or how^{10,11} large language models might exhibit understanding. Researchers have made great progress in this line of work by creating data sets of tasks that can be tested across a series of models (e.g., the GLUE and SuperGLUE data sets^{12,13}). However, experience shows that once-promising tests can contain issues that, when surfaced, raise questions about their effectiveness¹⁴ thus necessitating the development of novel tests that researchers must apply not only to recent models but also to old models so as to facilitate calibration by creating overlap with prior testing instruments.

Ultimately, such comparative studies have scientific and historical value: they shed light on how model features influence performance and they develop a chronology of that performance trajectory. Ideally, conducting such studies in the future would resemble a historian's archival research: a researcher would access models in a well-

organized, carefully curated library to test new hypotheses about model versions across time. However, current deprecation practices portend a future more akin to archeology: a process of spending considerable time sleuthing the whereabouts of past models—or their traces—with the hopes of being able to dust them off, preserving some semblance of their past essence, and testing hypotheses with whatever fragments persist. The possibility of such difficult circumstances resulting from current deprecation methods suggests a reason for new approaches that strike a balance between commercial and research needs.

New methods of deprecation

Finding compromise amidst conflicting commercial and user interests appears possible, thus we encourage developers and policymakers to forge this compromise. Outdated models impose liabilities and create opportunity costs on developers, harming the very enterprises that users rely on to provide their AI toolkit. At the same time, current deprecation practices force alterations to currently functional software applications on uncertain time intervals and inhibit historical research, replication, and novel scientific investigations, creating hassles for a key user group that developers hope to serve—namely, high-volume, professional users apt to influence wider usage trends among the mass-user base. In other words, developers and users include each other's welfare in their own interests, thereby exhibiting the type of interdependence that has amplified human cooperation throughout history¹⁵. In this situation, cooperation means users accepting restrictions for the benefit of continued access and paying for that access, while developers allow such ongoing use for the benefits of limited liability and an ongoing stream of new insights derived from continued use of outdated models.

Specifically, the first step in devising new methods of deprecation involves ensuring that the commercial developers of LLMs do not face penalties for providing ongoing access to outdated models. Capping or eliminating liability for any model that an enterprise designates as outdated, as well as eliminating all warranties for such models, would be necessary—much like liability limits designed in the U.S. tobacco settlement of the 1990s¹⁶. To ensure that developers do not deprecate models strategically to avoid liability or costly warranties, the designation of a model as “outdated” would limit the model's use solely to non-commercial, historical, and scientific purposes. This designation would impose a further cost on users as such restrictions would require users both to certify that their deployments accord with usage limits and to accept monitoring to verify adherence to those limits. Users, reasonably, would need to fund this monitoring and pay the costs associated with gaining access to outdated models. Agreeing to those costs would reflect the user's expectation that the benefits of accessing outdated models for narrow historical and scientific purposes exceed the costs themselves. Moreover, user fees could include additional expenses to redress uncompensated economic and environmental costs—i.e., negative externalities—that result from model use¹⁷, thus preventing inefficient overconsumption of deprecated models.

But what would developers get from the costs they will incur should they provide continued access? First, one might note that developers could limit the ongoing costs they face by simply releasing the technical details (e.g., weights, architecture details, etc.) of outdated models. Some developers already take an open-source approach by providing access to technical details and letting users run models on their own hardware (e.g., Meta's Llama series). Making such an approach universal would offer a straightforward means of perpetual model access. Alternatively, sending that information to a centralized


archive—such as a government office or a third-party host—presents another way in which developers could outsource ongoing access to others. Such options address the positive externalities—that is, uncompensated benefits—that would result from developers providing ongoing access: it would distribute the costs across the community that would obtain the diffuse benefits resulting from ongoing access.

The prospect of a public or third-party entity hosting deprecated models also could take advantage of economies of scale by storing large troves of models that could be retrieved episodically according to user demand, thus limiting the massive financial and environmental costs that would accrue from running the models perpetually. Even if it ran models episodically, however, a public or third-party entity building and maintaining the hardware necessary to run deprecated models would face an expensive bill. Yet, such public or non-commercial computing infrastructure appears increasingly necessary for the purposes of national security and scientific competitiveness. Using it residually to access deprecated models would advance those objectives, while alleviating developers' costs, preserving AI history, and promoting scientific replicability.

Open-source approaches, however, might reveal proprietary information, thus making the release of model details undesirable. If so, developers still stand to benefit even if they take a more involved approach in which they continue to keep technical details proprietary and provide API access to models running on their own hardware. Historical and scientific research will enhance understanding of LLMs, thus providing the type of insights that often seep into commercial activity downstream. Also, by enabling historical and scientific research opportunities associated with their products, commercial providers expand the scope of their business activities and create more opportunities for user engagement with their products. Such engagement, it warrants mentioning, would stem largely from professional users, not a mass-consumer audience. Although this group would include fewer users, its per-user usage rate likely would exceed that of the average user and that usage has the promise of yielding further scientific benefits that enterprises could capture.

Together, these reasons suggest a clear path forward. Across jurisdictions, developers and policymakers can convene to hammer out the details of a system akin to the one outlined above and, in so doing, they can create model legislation or administrative rules that will bring to life a new approach to model deprecation. A system of ongoing, monitored access to liability-limited and warranty-free models for historical and scientific research purposes offers benefits to developers and users at a modest cost; commercial interests and public officials should make it happen.

Tim Johnson ¹ & **Nick Obradovich**²

¹Atkinson Graduate School of Management, Willamette University, Salem, OR, USA. ²Laureate Institute for Brain Research, Tulsa, OK, USA.  e-mail: tjohnson@willamette.edu

Received: 27 May 2024; Accepted: 13 November 2024;
Published online: 26 November 2024

References

1. Adams, H. *The Education of Henry Adams* (Project Gutenberg, 2011/1918).
2. Mitchell, M. et al. Model cards for model reporting. In *FAT* '19: Proceedings of the Conference on Fairness, Accountability, and Transparency*, 220–229 <https://doi.org/10.1145/3287560.3287596> (2019).

3. Wang, B. et al. DecodingTrust: a comprehensive assessment of trustworthiness in GPT models. *Proc. Thirty-Seventh Conference on Neural Information Processing Systems Data Sets and Benchmarks Track* (2023).
4. Kumar, S., Balachandran, V., Njoo, L., Anastasopoulos, A. & Tsvetkov, Y. In *Proc. 17th Conference of the European Chapter of the Association for Computational Linguistics* (eds Andreas Vlachos & Isabelle Augenstein) 3299–3321 (Association for Computational Linguistics, 2023).
5. Vincent, J. OpenAI sued for defamation after ChatGPT fabricates legal accusations against radio host. *The Verge* (2023).
6. Henderson, P. et al. Foundation models and fair use. *J. Mach. Learn. Res.* **24**, 1–79 (2023).
7. Calabresi, G. Some thoughts on risk distribution and the law of torts. *Yale Law J.* **70**, 499–553 (1961).
8. Rahwan, I. et al. Machine behaviour. *Nature* **568**, 477–486 (2019).
9. Mitchell, M. & Krakauer, D. C. The debate over understanding in AI's large language models. *Proc. Natl. Acad. Sci.* **120**, e2215907120 (2023).
10. Piantadosi, S. T. & Hill, F. Meaning without reference in large language models. In *Proc. NeurIPS 2022 Workshop on Neuro Causal and Symbolic AI (nCSI)* (2022).
11. Lake, B. M. & Murphy, G. L. Word meaning in minds and machines. *Psychol. Rev.* **130**, 401–431 (2023).
12. Wang, A. et al. Superglue: a stickier benchmark for general-purpose language understanding systems. *Adv. Neural Inf. Process. Syst.* **32** (2019).
13. Wang, A. et al. GLUE: a multi-task benchmark and analysis platform for natural language understanding. In *Proc. EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 353–355 (2018).
14. Niven, T. & Kao, H.-Y. In *Proc. 57th Annual Meeting of the Association for Computational Linguistics* 4658–4664 (2019).
15. Tomasello, M., Melis, A. P., Tennie, C., Wyman, E. & Herrmann, E. Two key steps in the evolution of human cooperation: the interdependence hypothesis. *Curr. Anthropol.* **53**, 673–692 (2012).
16. Bulow, J. & Klemperer, P. The tobacco deal. *Brook. Pap. Econom. Act. Microecon.* **1998**, 323–394 (1998).
17. Bender, E. M., Gebru, T., McMillan-Major, A. & Shmitchell, S. In *Proc. ACM Conference on Fairness, Accountability, and Transparency* 610–623 (Association for Computing Machinery, Virtual Event, Canada, 2021).

Acknowledgements

The authors thank Arunas Radzvilavicius and two anonymous reviewers for providing insights, suggestions, and edits that greatly improved this commentary. The authors also thank Manuel Cebrian for illuminating conversations that have shaped their thinking on AI policy, including ideas conveyed in this commentary.

Author contributions

T.J. came up with the idea, wrote the first draft of the paper, and edited the paper. NO helped clarify key ideas, contributed amendments and edited the paper.

Competing interests

The authors declare the following competing interests: T.J. and N.O. have received in-kind resources from OpenAI's Researcher Access Program for previously conducted research.

Additional information

Correspondence and requests for materials should be addressed to Tim Johnson.

Peer review information *Nature Communications* thanks the anonymous reviewer(s) for their contribution to the peer review of this work.

Reprints and permissions information is available at <http://www.nature.com/reprints>

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

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2024