Before the
National Telecommunications and Information Administration
Washington, D.C. 20230

Dual Use Foundational Artificial Intelligence) Docket No. 240216-0052
Models With Widely Available Model Weights) RIN 0660-XC060

COMMENT

This comment primarily responds to questions 1, 6a, 6c, and 6d presented in the request for comment. It is grounded in the experience of the open source hardware community, a community that has worked to apply frameworks of openness developed in the context of software and culture to a much more complex hardware landscape.[1] This translation process holds potential lessons for open foundation models.

Specifically, this comment focuses on the role that open licenses play in determining the relative openness of a foundation model. Open licenses can provide useful information in evaluating the openness of a foundation model. However, unlike in areas such as open source software, at this stage they cannot be used as simple heuristics to quickly sort open and not-open models. As a result, NTIA should not adopt a definition of open foundation model that incorporates or requires specific licenses. Instead, it should look to a function-based definition of open that is agnostic towards the licensing structure used to meet it.[2]

The wisdom of avoiding a license-specific definition is grounded in differences between open foundation models and other open works. Open source software and open cultural license requirements are effective in other contexts because those licenses are mature, have been widely adopted by the relevant communities over an extended period of time, and can be used to quickly assess the openness of a work. What constitutes infringement of copyright-protected open source software is also relatively settled.

Open foundation models are comparatively more complex in composition and intellectual property protection. They are also less mature, and far from standardized within the relevant community. Infringement may or may not take a wide range of untested forms. While open licenses can be used to help achieve functional openness, specific open licenses cannot be used as a proxy for it.

[1] Commenter is a long-time board member of the Open Source Hardware Association (OSHWA) and director of OSHWA's certification program. However, this comment is submitted in Commenter's personal capacity and not submitted on behalf of OSHWA.
[2] This approach is in line with other Federal frameworks for evaluating openness. For example, the Federal Source Code policy defines Open Source Software as meeting definitions of open maintained by the Open Source Initiative or Free Software Foundation. *See* OMB Memorandum M-16-21, *Federal Source Code Policy: Achieving Efficiency, Transparency, and Innovation through Reusable and Open Source Software*, https://obamawhitehouse.archives.gov/sites/default/files/omb/memoranda/2016/m_16_21.pdf at 14.

These characteristics make open foundation models much more conceptually similar to open source hardware than open source software. Like open foundation models, open source hardware consists of a number of elements with varying relationships to copyright and other intellectual property protections. Also, like the open foundation model community, the open source hardware community has worked to translate core concepts of openness into field-specific licenses. While open hardware thrives, its experience suggests that it may be some time before the open foundation model community coalesces around a stable definition of the components of an open foundation model, or a consensus regarding how those elements should be licensed. That is why a functionality-based approach to defining open foundation models may be more effective at this time.

Open Source Software and Creative Commons Licenses as Proxies for Openness

Open licenses can make it easy to evaluate compliance with broader open requirements. Many funders and governments require software to be licensed under specific licenses in order to meet open requirements.[3] Creative Commons licenses play similar roles in open requirements related to journal articles and other written works.[4] In each of these cases, if works meet two criteria - actual accessibility of the work itself and legal accessibility enabled by a qualifying open license - it qualifies as "open".[5]

These requirements work because both elements are fairly easy to evaluate in these traditional "open" contexts. Code is "accessible" if it can be accessed without significant barriers. Often this means it is downloadable from a repository such as github, and written in a human-readable (and editable) programming language. Similarly, articles are "accessible" if they can be downloaded from an online repository without onerous financial or technological costs. In both cases, the work is usually easy to identify in a standardized form - a code repository or article. If the code cannot be run, it is not accessible. If an article cannot be read, it is not accessible.

The licensing requirement is similarly straightforward. Software, prose, and images, the types of works most often covered by traditional open policies, are completely and categorically protected by copyright, both in the United States and internationally. In their canonical form, a single copyright covers the entire work. In these cases, an open copyright license is required to, and effectively does, remove any legal barriers to openness.

[3] *See, e.g*. Open Source Initiative, *International Authority & Recognition*, https://opensource.org/authority.
[4] *See, e.g*. Creative Commons, *Government*, https://creativecommons.org/government/.
[5] These two characteristics can be used to evaluate compliance with the OSI-hosted open source definition (https://opensource.org/osd) or the Free Software Foundation's definition of "Free Software" (https://www.gnu.org/philosophy/free-sw.html).

The licensing requirement is further simplified by a mature, widely adopted, and widely recognized set of licenses. While requirements do not always limit creators to these specific licenses, the OSI-approved licenses[6] for open source software and Creative Commons licenses[7] for works such as journal articles are well understood and pervasive in their relevant spheres.

In practice, this means that many institutions use the presence of a compliant open license as a proxy for evaluating openness. If the work is openly licensed, it meets their requirements. Compliance checks are quick, easy, and do not require specialized knowledge about the work being evaluated.

Open Data and a Complex International Copyright Regime

Data is also often subject to open requirements that are similar to software and articles.[8] While these policies often operate similarly to standard open policies, they do differ in one important way. Unlike software or a journal article, data itself is not usually eligible for copyright protection in the United States.[9] Without a legal barrier to reuse, there is no need for a license to remove that barrier.

However, many policies require or encourage an open license, both in order to remove barriers to reuse that copyright might present in other jurisdictions, and as a fallback in the event that specific elements of specific datasets are eligible for copyright reasons.

As such, open data represents an example of how open requirements apply to a situation where the intellectual property protection applies less uniformly to a category (data).

Open Hardware Represents Complex Accessibility and Intellectual Property Protection

Open Source Hardware provides an even more complex challenge for open policies. With its complexity, it serves as a model of the challenges related to using licenses to identify open foundational models.

[6] Open Source Initiative, *OSI Approved Licenses*,https://opensource.org/licenses.
[7] Creative Commons, *Licenses List*, https://creativecommons.org/licenses/list.en.
[8] Many funders, including the National Science Foundation, require grantees to prepare a "data management plan" to make raw data publicly available. *See* National Science Foundation, *Today's Data, Tomorrow's Discoveries: Increasing Access to the Results of Research Funded by the National Science Foundation* (March 18, 2015), https://www.nsf.gov/pubs/2015/nsf15052/nsf15052.pdf at 5.
[9] *See Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340 (1991).

First, the hardware itself is not simply a single repository or document. Instead, open hardware is more accurately conceived of as some combination of a) the hardware itself, b) the hardware design files, c) documentation related to the hardware's use, assembly, and operation, d) and any software required to operate the hardware.[10] This is often a significantly more complex combination than a software repository or journal article.

Second, each of these elements may or may not be eligible for copyright protection (or other types of intellectual property protection). Any given hardware product may have copyrightable hardware, design files, documentation, and software. However, there are other instances where some or none of these elements are protectable by copyright at all.[11]

This creates a challenge when developing easy-to-use tests for open compliance. When it comes to accessibility, it may not be immediately obvious if a specific piece of hardware requires design files, or documentation, or software, or if those components are accessible and well documented enough to facilitate reuse. Similarly, it is hard for non-experts to evaluate the copyright status of each of those elements, and therefore to evaluate the adequacy of a license in granting permission (or even evaluate the necessity of a license in the first place).[12]

Over ten years after the creation of the Open Hardware Definition,[13] the open source hardware community has developed a certification program to identify compliant hardware,[14] as well as a growing suite of licenses.[15] The shared definition, certification program, and suite of licenses have meaningfully improved the process of identifying open hardware.

However, they are significantly less precise when compared with methods used in the open source software community. At the same time, they are significantly more mature than what is currently available for foundational models.

[10] OSHWA, *Best Practices for Open Source Hardware 1.0*, Licensing your Designs, https://www.oshwa.org/sharing-best-practices/

[11] For example, the physical hardware could consist of an assemblage of purely functional, non-expressive mechanical components, or the design files could merely be a series of coordinates for cutting and drilling a material. Neither of these are likely to be eligible for copyright protection.

[12] There is an ongoing debate within the open hardware community as to whether it is better to err on the side of over- or under-application of licenses to works with ambiguous copyright protection statuses. *See, e.g.* Michael Weinberg, *Is it Better to Over License?* (January 19, 2017), https://michaelweinberg.org/post/156095370255/is-it-better-to-over-license.

[13] OSHWA, *Definition*, https://www.oshwa.org/definition/.

[14] OSHWA, *Open Hardware Certification Program*, https://certification.oshwa.org/.

[15] OSHWA, *Recommended Licenses for Hardware*, https://certification.oshwa.org/process/hardware.html.

Licenses Should Not Act as a Primary Criteria for Identifying Open Foundational Models

Foundational models are more complex, and their use is less mature, than open hardware, let alone open software. This complexity applies to the elements of a model, and those elements' relationship to intellectual property protections. In fact, model weights - in many ways the heart of foundational models - are unlikely to be protectable by copyright at all. As such, a license will be even less effective as a tool to quickly evaluate a model's openness.

There is not a universally-agreed upon definition of what elements need to be accessible for a model to qualify as open

Like open hardware, foundational models can be described as containing a number of different components. These include the model itself, the code for running the model, the weights that make up the model, and the data used to train the model. There are not universal ways to think about these components, or blanket statements that can be made about their relationship to copyright protection.

The definition of "open" in the context of foundational models continues to evolve, with many different approaches making a reasonable claim to being open depending on the context and intended use.[16] A model that is open for the purposes of allowing users to run the model independently of its creator may not be open for the purposes of interrogating the data used to train it, all of which is distinct from a model that is open in a way that can be tuned, modified, or built upon by others. Other commenters will likely submit detailed discussions of these differences.

Without a universal standard for open, it is impossible to evaluate whether or not the actual availability of model elements meets a general "open" standard. This proceeding may make an important contribution to establishing a functional framework for evaluating the openness of foundational models.

[16] Irene Solaiman *The Gradient of Generative AI Release: Methods and Considerations* (February 2023), https://arxiv.org/abs/2302.04844 at 4 described a gradient of system access for open models. This spectrum has since been modified, including in Rishi Bommasani, et. al. *Considerations for Governing Open Foundation Models* (December 2023), https://hai.stanford.edu/sites/default/files/2023-12/Governing-Open-Foundation-Models.pdf at 3.

There is not a standard, mature, broadly applicable licensing framework for licensing foundational models or their elements

However, even if NTIA is able to develop a standard framework for describing open foundational models, we do not currently have a broadly applicable set of mature licenses for foundational models. There is no equivalent of software's MIT or Apache license that effectively controls the entire model.

One reason for this is the potential variability in how and if each element of the model is protectable by copyright. The code for running the model itself is most likely to be protected by copyright law, and therefore controllable by a license.

The copyright status for model weights is much less clear. Conceived of as a collection of facts or data points, they are not protectable under United States copyright law.[17] Without copyright protection, there is no license necessary to use them, and no license terms that can control their use.[18] This represents a fundamental departure from the open framework developed around copyrightable software and other cultural goods.

Licensing the data used to train the model is potentially even more complex. An entity that trains a model rarely relies exclusively on data fully within its control, or even that it has a formal agreement to make use of.[19] Furthermore, the training data is sometimes distributed as a collection of data itself (likely containing individually copyrightable elements created by a range of third parties), and sometimes merely as a collection of links and pointers to locations where the data can be retrieved (less likely to contain individually copyrightable elements).[20] One could conceive of a definition of an open data set requiring individual copyright licenses for each piece of data in the set, or no license at all for an unstructured list of five billion urls.

The complex relationship that each element of a model has with copyright protection is further complicated by variability in how necessary each of these elements are for a model to qualify as open. In *The Gradient of Generative AI Release: Methods and Considerations*, Irene Solaiman describes a "gradient of system access" where "fully open" requires "all aspects of the system are accessible and downloadable, including all components."[21] Bommasani, et. al. modify that definition by describing any "models with widely available weights" as "open foundation models".[22] This proceeding is likely to produce yet another definition of what it means to be an open foundation model.

[17] Feist, 499 U.S. 340 (1991).

[18] There may be other legal ways to control use, with varying degrees of effectiveness. But none of those give you a simple way to evaluate the openness of a model.

[19] Although this point is subject to active litigation, most model trainers appear to rely on fair use with regard to third party training data.

[20] For example, the LAION-5B dataset is structured as "5.85 billion pairs of image URLs and the corresponding metadata". Users of the dataset are expected to reconstruct the actual images by following the links and downloading the images themselves. https://laion.ai/blog/laion-5b/

[21] Solaiman at 5, 6.

[22] Bommasani at 3.

Separate from the relative merits of these approaches, they represent significantly different licensing requirements. If widely available weights are all that are required to qualify as open, and weights themselves are not subject to copyright protection, then no license would be necessary for an open model. Conversely, if all elements of the system must be accessible and downloadable, a number of licenses from a number of different parties could conceivably be required.

In light of this variability, it is not a surprise that the community has not developed and adopted a standard suite of models on par with the OSI-approved open source software licenses or Creative Commons licenses. Efforts such as the RAIL licenses are commendable efforts to create model-specific licenses that also expand the scope of what open licensing can be in this context.[23] However, it would not be accurate to say that the community has yet developed a consensus about their use. For example, the three models used as categorical examples of open foundation models in Bommasani, et. at., each use a different license: Meta's Llama 2 uses a custom license,[24] BigScience's BLOOM uses a custom RAIL license,[25] and EleutherAI's GPT-NeoX uses Apache 2.0[26] (a standard open source software license).

While it is possible that a consensus around licensing evolves within the open foundational model community in the future, it does not exist yet. As a result, NTIA's definition of open foundational models should not include specific licensed-based requirements.

NTIA Should Focus on a Functionality-Based Definition of Open Foundational Models

License-based definitions of open are popular because compliance with them is straightforward, as is evaluating that compliance. Without that option, NTIA should focus its definition of open foundation models on functionality. Specifically, the definition should describe what others must be able to do with the model in order to meet the definition. How that is achieved will be the responsibility of the model developers.

This approach imposes burdens on both model developers and those hoping to evaluate openness compliance. Model developers will need to do their own legal review, and to choose between (or develop new) licensing options for their models. Similarly, evaluators will need to examine models more precisely in order to identify if specific actions are technically and legally possible.

[23] *Responsible AI Licenses*, https://www.licenses.ai/.
[24] *Request Access to Llama Agreement*, https://llama.meta.com/llama-downloads/.
[25] *BigScience RAIL License v 1.0*, https://huggingface.co/spaces/bigscience/license.
[26] https://github.com/EleutherAI/gpt-neox/blob/main/LICENSE.

These burdens need not be permanent. Over time, model developers may develop more standardized ways to meet the standards and communicate their compliance. At some point, those methods might be incorporated into the definition itself.

However, until those ways have been standardized, NTIA should not be overly reliant on approaches that work well in open source hardware when developing its open foundational model test. Doing so would likely freeze development at a moment when more experimentation is beneficial to the community.

/s/
Michael Weinberg
hello@michaelweinberg.org
3/26/24