Copyright Implications of the Use of Code Repositories to Train a Machine Learning Model

John A. Rothchild[*] & Daniel H. Rothchild[**]

I. Introduction

GitHub is a Web-based platform that facilitates collaboration among software developers by allowing them to deposit their code in repositories hosted on the site. The company has recently launched a product called Copilot, which assists developers by generating code that they can incorporate into their projects. Copilot generates code using a machine learning model trained to convert developer-provided natural language descriptions of code into actual code. As with any machine learning model, Copilot requires training data, which GitHub sources in part from public repositories available on the site. The creators of these repositories have generally not explicitly licensed GitHub to use their code for this purpose, which raises the question: does GitHub's use of these code bases to train Copilot and to generate code infringe the copyrights of the creators of the code bases?

GitHub's development of Copilot implicates the creators' copyrights both when training the model and when deploying Copilot as a product. We consider each of these possible infringements below, and conclude that Copilot likely does not infringe copyright through these activities. First, it is possible that GitHub's Terms of Service permits GitHub to use the code in both contexts. Second, GitHub's activities are in any event permitted as fair use, or may even involve only de minimis copying that is non-actionable.

Copilot's developer-customers may use code that Copilot generates either by incorporating it directly into their own software or by using it to inspire them to write new code. We conclude that in both cases, developer-customers likely do not infringe the copyright of the creators of the repositories included in Copilot's training data.

II. Background: copyright law, GitHub and machine learning

A. Copyright law

* Professor of Law, Wayne State University. Prof. Rothchild has taught Copyright Law, Constitutional Law, Trademarks, Electronic Commerce Law, and other legal subjects since 2001. He is a coauthor of *Internet Commerce*, one of the first law school casebooks for teaching Internet law. He serves as Chief Privacy Officer of Wayne State University and is Faculty Director of the Master of Studies in Law degree program. A.B., Princeton University; J.D., University of Pennsylvania Law School.

** Daniel Rothchild is a Ph.D. student in the Electrical Engineering and Computer Sciences Department at the University of California, Berkeley. Rothchild's current research interests include sequence modeling for drug discovery using large transformer models, like those employed by Copilot. Previously, he studied large-scale distributed and federated learning on an NSF Graduate Research Fellowship, and machine learning for astronomy on a Churchill Scholarship at the University of Cambridge. Rothchild holds an A.B. in Physics Summa Cum Laude from Harvard University, and an M.Phil. in Astronomy from the University of Cambridge.

The Copyright Act is a U.S. federal law that grants certain legal rights to creators of works of authorship, such as books, artworks, music, and computer code. Original code is automatically protected by copyright as soon as it is written and saved to some tangible medium. The Act grants the author of a protected work the right to exclude others from making certain uses of the work: copying it, making a derivative work based on that work, distributing copies of the work to the public, and publicly performing or displaying the work.

Users of GitHub are generally the authors of the code they write and upload, and therefore own the copyright to it.

The rights that the Copyright Act grants are, however, limited by fair use: if a person's use of someone else's copyrighted material is found to be "fair," then the copyright owner has no right to forbid that use. The Act sets out a four-factor test to determine whether a use is fair: (1) "the purpose and character of the use," (2) "the nature of the copyrighted work," (3) how much of the copyrighted work is used, and (4) the economic effect of the use on the copyright owner. 17 U.S.C. § 107. The vagueness of these factors, and the need to balance multiple factors, means that it is often difficult to determine whether a particular use is fair.

B. GitHub

GitHub provides software as a service for source control management using open-source software called "git" (which tracks changes in a set of files), collaboration among developers, and related activities. Users can create their own public or private code repositories, contribute to other users' repositories, and discuss code changes with other developers. Code in many repositories is documented using comments, which are natural language descriptions of code that often appear in code files alongside the code being documented.

C. Machine learning

Copilot uses a machine learning model trained to generate code snippets based on comments describing the code. Machine learning models require data to train on, called a training set, which Copilot sources in part from public GitHub repositories. During training, Copilot is presented with a comment from the training set and is optimized using a technique called stochastic gradient descent to generate output similar to the code that the comment describes. A number of techniques are used to ensure that the model learns to generate novel code rather than to memorize and output code from the training set, but some amount of memorization is usually unavoidable when training large models. As a result, Copilot sometimes "generates" code that is found verbatim in the training set.

III. Legal analysis

A. GitHub's Terms of Service may permit it to use submitted code as a training set for Copilot and to output portions of submitted code to Copilot users

Users who wish to deposit their code into a GitHub repository must agree to the website's Terms of Service. These terms grant GitHub certain rights to use the code in ways that might otherwise infringe the user's copyright. Specifically, the user grants GitHub "the right to store, archive, parse, and display Your Content, and make incidental copies, as necessary to provide the Service," where the "Service" is defined to include all services provided by GitHub, including Copilot. The license also grants GitHub the right to "copy [code] to our database and make backups," "show it to you and other users," and "parse it into a search index or otherwise

analyze it on our servers."

It is not clear to what extent the training and operation of Copilot implicate the copyright rights of the code submitters. Training Copilot is a form of analysis carried out on GitHub's servers that probably entails copying beyond what occurred on GitHub pre-Copilot, since training will involve copying code into a computer's random access memory. Code generated by Copilot might or might not contain verbatim snippets from the training data. Even assuming that these operations involve copying, they seem to be explicitly allowed in the Terms of Service ("copy" and "show"). Thus it might be argued that users have authorized GitHub to use their code to create and operate Copilot.

However, a court might find the Terms of Service ambiguous. Prior to Copilot, "pars[ing]...into a search index or otherwise analyz[ing]" and "copy[ing] to our database" were likely understood to refer to low-level analysis to allow basic functionality of GitHub such as searching, locating definitions, merging new code, etc. Similarly, "show[ing code] to you and other users" was likely understood as displaying the original repository to users who navigated to it on GitHub's website, not displaying unattributed snippets to arbitrary Copilot users. A court might find that Copilot is an unanticipated new technological use of the code repositories and that the license fails to unambiguously convey the intent of the parties. In such a situation, some courts will find that the license includes only such uses as fall within the unambiguous core meaning of the license terms (i.e. simple parsing and display). Other courts will find that the novel use (i.e. training Copilot) is allowed as long as it may reasonably be said to fall within the scope of the license terms. Bartsch v. Metro-Goldwyn-Mayer, Inc., 391 F.2d 150 (2d Cir. 1968).

One last possibility is that a court might find that the code Copilot generates is a derivative work based on the deposited code, and that the license terms do not specifically allow creation of derivative works. Caselaw on this point is sparse.

B. GitHub's use of the code repositories to train its machine learning model is likely fair use

In order to use the contents of the code repositories in its training set, GitHub must make copies of that code. If a court finds that the code submitters have not authorized such copying by their acceptance of the Terms of Service, then the permissibility of GitHub's use would turn on whether the use is "fair" under the terms of the Copyright Act. To perform a fair use analysis a court will apply and balance the four statutory factors.

1. Purpose and character of the use

This factor has two components: (i) whether the use is for a commercial purpose, and (ii) whether the use is transformative. The U.S. Supreme Court has instructed that the second of these components carries the most weight: the goal of analysis under the first factor is to determine whether the use merely "supersedes" the purpose of the original work (such as by making exact copies of the work and selling them in competition with the originals), "or instead adds something new, with a further purpose or different character, altering the first with new expression, meaning, or message." Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569 (1994).

There is a strong argument that GitHub's use of the code repositories as training data is a transformative use. The original code was written to accomplish a particular purpose of the developer — say, to sort a list of elements, or to perform a mathematical calculation. GitHub uses the code for an entirely different purpose: to teach its AI how to generate new code based

on a natural language description. An analogy may be found in a case in which students were required to submit their papers to an online plagiarism-detection service. After comparing a submitted paper to papers available on the Internet and those in its own database, the company would add the student's paper to its database and use it when analyzing future submissions. The court found that the company's use of the papers was highly transformative: the purpose of a paper is to convey its expressive content, while the purpose of the database is to detect plagiarism. A.V. ex rel. Vanderhye v. iParadigms, LLC, 562 F.3d 630 (4th Cir. 2009). Likewise, GitHub is not using the copied code to sort lists, but rather to train its AI to create code that will accomplish a particular purpose.

Any potential verbatim copying of the training data by Copilot during deployment is likely transformative as well. Copilot generates relatively short snippets of code that Copilot users incorporate into their own project, which is unlikely to simply supersede whatever repository the snippet was copied from.

Although GitHub's purpose in developing Copilot is presumably commercial, a court would likely find that the transformativeness of the use outweighs the commercial purpose and that this factor weighs in favor of fair use.

2. The nature of the copyrighted work

In most situations this factor has little impact on a fair use analysis, but it has played a prominent role in several cases involving copying of computer code. The U.S. Supreme Court recently held that Google's copying of the declaring code of the Java API (roughly speaking, the names of each of the functions that compose the API) into its Android mobile operating system was fair use. The Court found that the "nature of the copyrighted work" factor supported fair use, explaining that Google's copying of the declaring code was not motivated by laziness but rather by the goal of allowing programmers familiar with the Java API to apply their existing skills to developing apps for the Android platform. Google LLC v. Oracle America, Inc., 141 S. Ct. 1183 (2021). Another court held that it was fair use for game developer Accolade to copy the code from a Sega game cartridge as part of the process of disassembling it in order to determine how to create its own games that would be compatible with the Sega Genesis console. The goal, again, was to promote the creation of new works of software. Sega Enterprises Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992).

Since GitHub's purpose in copying the deposited code is not to sort lists, etc., but rather to enable developers to create new programs more easily, a court might find that this factor supports a finding of fair use.

3. How much of the copyrighted work is used

Generally, the more of a work the second user appropriates, the less likely a court is to find that the use is fair. But the amount that the second user may fairly use depends on the purpose for using it. In many cases the courts have held that the purpose of the use justified the copying of the entirety of the protected work.

For training Copilot, GitHub is presumably copying the entirety of the code repositories into its training set. Given the purpose of the copying, this amount of copying is justified. Copying less than the entirety of the repositories would cause the training set to be less useful in accomplishing its goal of training a code-generating model; machine learning research has shown that the performance of language models such as those used by Copilot scales to

dataset sizes at least as large as all public code on GitHub.

In deployment, Copilot is unlikely to be copying large fractions of copyrighted works, since generated snippets are relatively short, and Copilot rarely quotes verbatim from the training data: according to GitHub, “about 0.1% of the time, [Copilot’s] suggestion may contain some snippets that are verbatim from the training set.” https://copilot.github.com/ (FAQs, linked from “Protecting originality”). That Copilot copies only a small amount of code would contribute to a fair use determination, and a court may even view such a small amount of copying as de minimis and therefore not enough to support a copyright claim at all. We note, however, that the “about 0.1% of the time” statement does not exclude the possibility that Copilot may sometimes output code that is verbatim from a code repository and is of substantial length. In addition, the 0.1% figure excludes some verbatim copying that GitHub deems unimportant, and also excludes near-verbatim copying (e.g., copying a code snippet and changing only the variable names). To the extent this sort of copying occurs, the argument for fair use is weakened.

4. The economic effect of the use on the copyright owner

This factor requires courts to assess the extent to which the unauthorized use interferes with the copyright owner’s ability to derive economic value from the original work. The question is whether the work resulting from the copying is a market substitute for the original work.

In applying this factor, a court might lay primary stress on how transformative the use is. The more transformative the use, the less the second work can act as a market substitute for the original work. A court might find that Copilot does not in any way substitute for the code in the repositories because it has a completely different function: it generates new code, whereas the repository code causes a computer to e.g. sort or calculate.

In some circumstances, courts apply this factor by inquiring whether the unauthorized use causes the copyright owner to lose the ability to command licensing fees. There is an inherent circularity to this approach: a copyright owner may assert that he would demand substantial licensing fees for the challenged use, and therefore the use is not fair; but the copyright owner is entitled to licensing fees only on the assumption that the use is not fair. Some courts have sought to overcome the circularity by holding that an assertion of a right to licensing fees will be credited only where it pertains to “traditional, reasonable, or likely to be developed markets.” American Geophysical Union v. Texaco Inc., 60 F.3d 913 (2d Cir. 1994). A court might reason that there is no established market for contributing code to a training set or providing a snippet of code to include in one’s own project, and that no such market is likely to be feasible, and conclude that this factor supports fair use.

C. The use of Copilot’s output by its developer-customers is likely not infringing

A developer-customer of Copilot may make use of the code that Copilot outputs in a few different ways.

First, the developer may incorporate the code into the developer’s own project. This constitutes copying, which is one of the exclusive rights of the copyright owner. The copying does not infringe any copyright that GitHub may hold in the code, since GitHub will have licensed the developer’s use of it, either explicitly or by implication. What about the copyrights held by the owners of the code repositories? As discussed above, the code that Copilot outputs is unlikely to represent verbatim code contributed by a code repository owner – according to GitHub, the code that Copilot outputs will include verbatim snippets of the repository code only about 0.1%

of the time. So there are two possibilities. It may be that Copilot has presented the developer with code that is copied from a code repository, and the developer copies that code into its project. In that case, a fair use analysis similar to that set out above would apply, likely concluding that the use is fair. Or it may be that the outputted code is not verbatim from (or substantially similar to) any code from a code repository. Non-verbatim copying can be infringing: for example, it is possible to infringe by copying the plot of a literary work, even if none of the work is copied verbatim. However, the courts' usual approach to assessing non-verbatim copying of computer code is likely to yield the result that the copying was only of ideas, not of expression, and is therefore non-infringing. Computer Associates Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992).

Alternatively, the developer might not copy the code but only use it for inspiration – as GitHub ackknowledges, the code Copilot generates "may not always work, or even make sense." https://copilot.github.com/ (FAQs, linked from "General"). In this situation the copying would perforce amount to copying of ideas rather expression, and would not be infringing.

IV. Conclusion

Many software developers who host code on GitHub are concerned that GitHub is using their code without their knowledge to train Copilot, and that Copilot will potentially provide other developers with snippets of their code without displaying their license. To be clear, the analysis presented above does not absolve GitHub of wrongdoing, but rather argues that Copilot and its developer-customers likely do not infringe developers' copyrights. We do not address whether Copilot violates free software licenses in ways that do not infringe these copyrights, or whether Copilot violates code authors' moral rights. Copilot is still in beta testing, and GitHub is developing a system to detect when verbatim copying of the training set has occurred. However, detecting copying from such a vast training set is algorithmically quite difficult, especially when non-exact matches (e.g. the same code just with different variable names) must be considered.

One positive effect of Copilot is that developers are becoming increasingly aware of intellectual property issues and of the dangers associated with hosting their code with a third party they have no control over. We hope this discussion helps to elucidate some of the copyright issues that arise from Copilot's development.