EXHIBIT 2

GitHub Copilot investigation

Maybe you don't mind if GitHub Copilot used your open-source code without asking.

But how will you feel if Copilot erases your open-source community?

Hello. This is Matthew Butterick. I'm a writer, designer, programmer, and lawyer. I've written two books on typography—Practical Typography and Typography for Lawyers—and designed the fonts in the MB Type library, including Equity, Concourse, and Triplicate.

As a programmer, I've been professionally involved with open-source software since 1998, including two years at Red Hat. More recently I've been a contributor to Racket. I wrote the Lisp-advocacy essay [illegible] source software, including Pollen, which I use to publish my online books, and even AI software that I use in my work.

On November 3, 2022, we filed our initial complaint challenging GitHub Copilot. Please follow the progress of the case at githubcopilotlitigation.com.

In June 2022, I wrote about the legal problems with GitHub Copilot, in particular its mishandling of open-source licenses. Recently, I took the next step: I reactivated my California bar membership to team up with the amazingly excellent class-action litigators Joseph Saveri, Cadio Zirpoli, and Travis Manfredi at the Joseph Saveri Law Firm on a new project—

We're investigating a potential lawsuit against GitHub Copilot for violating its legal duties to open-source authors and end users.

We want to hear from you. Click here to help with the investigation.

Or read on.

This web page is informational. General principles of law are discussed. But neither Matthew Butterick nor anyone at the Joseph Saveri Law Firm is *your* lawyer, and **nothing here is offered as legal advice**. References to copyright pertain to

US law. This page will be updated as new information becomes available.

What is GitHub Copilot?

GitHub Copilot is a product released by Microsoft in June 2022 after a yearlong technical preview. Copilot is a plugin for Visual Studio and other IDEs that produces what Microsoft calls "suggestions" based on what you type into the editor.

> *It's a marketing stunt. It's a gag. But it's also a massive license-violation framework.*
> *—Jamie Zawinski*

What makes Copilot different from traditional autocomplete? Copilot is powered by Codex, an AI system created by OpenAI and licensed to Microsoft. (Though Microsoft has also been called "the unofficial owner of OpenAI".) Copilot offers suggestions based on text prompts typed by the user. Copilot can be used for small suggestions—say, to the end of a line—but Microsoft has emphasized Copilot's ability to suggest larger blocks of code, like the entire body of a function. (I demonstrated Copilot in an earlier piece called This copilot is stupid and wants to kill me.)

But how was Codex, the underlying AI system, trained? According to OpenAI, Codex was trained on "tens of millions of public repositories" including code on GitHub. Microsoft itself has vaguely described the training material as "billions of lines of public code". But Copilot researcher Eddie Aftandilian con-

firmed in a recent podcast (@ 36:40) that Copilot is "train[ed] on public repos on GitHub".

What's wrong with Copilot?

What we know about Copilot raises legal questions relating to both the **training** of the system and the **use** of the system.

On the training of the system

The vast majority of open-source software packages are released under licenses that grant users certain rights and impose certain obligations (e.g., preserving accurate attribution of the source code). These licenses are made possible legally by software authors asserting their copyright in their code.

Thus, those who wish to use open-source software have a choice. They must either:

1. comply with the obligations imposed by the license, or
2. use the code subject to a license exception—e.g., fair use under copyright law.

Microsoft and OpenAI have conceded that Copilot & Codex are trained on open-source software in public repos on GitHub. So which choice did they make?

> *[W]e've all just had a wakeup call that Microsoft is not the awesome, friendly, totally-ethical corporation that we've been told they are …*
> *—Ryan Fleury*

If Microsoft and OpenAI chose to use these repos subject to their respective open-source licenses, Microsoft and OpenAI would've needed to publish a lot of attributions, because this is a minimal requirement of pretty much every open-source license. Yet no attributions are apparent.

Therefore, Microsoft and OpenAI must be relying on a fair-use argument. In fact we know this is so, because former GitHub CEO Nat Friedman claimed during the Copilot technical preview that "training [machine-learning] systems on public data is fair use".

Well—is it? The answer isn't a matter of opinion; it's a matter of law. Naturally, Microsoft, OpenAI, and other researchers have been promoting the fair-use argument. Nat Friedman further asserted that there is "jurisprudence" on fair use that is "broadly relied upon by the machine[-]learning community". But Software Freedom Conservancy disagreed, and pressed Microsoft for evidence to support its position. According to SFC director Bradley Kuhn—

> *[W]e inquired privately with Friedman and other Microsoft and GitHub representatives in June 2021, asking for solid legal references for GitHub's public legal positions …* ***They provided none.***

Why couldn't Microsoft produce any legal authority for its position? Because SFC is correct: there isn't any. Though some courts have considered related issues, there is no US case squarely resolving the fair-use ramifications of AI training.

Furthermore, cases that turn on fair use balance multiple factors. Even if a court ultimately rules that certain kinds of AI training are fair use—which seems possible—it may also rule out others. As of today, we have no idea where Copilot or Codex sits on that spectrum. Neither does Microsoft nor OpenAI.

On the use of the system

We can't yet say how fair use will end up being applied to AI training. But we know that finding won't affect Copilot users at all. Why? Because they're just using Copilot to emit code. So what's the copyright and licensing status of that emitted code?

Here again we find Microsoft getting handwavy. In 2021, Nat Friedman claimed that Copilot's "output belongs to the operator, just like with a compiler." But this is a mischievous analogy, because Copilot lays new traps for the unwary.

Microsoft characterizes the output of Copilot as a series of code "suggestions". Microsoft "does not claim any rights" in these suggestions. But neither does Microsoft make any guarantees about the correctness, security, or extenuating intellectual-property entanglements of the code so produced. Once you accept a Copilot suggestion, all that becomes your problem:

"You are responsible for ensuring the security and quality of your code. We recommend you take the same precautions when using code generated by GitHub Copilot that you would when using any code you didn't write yourself. These precautions include rigorous testing, IP [(= intellectual property)] scanning, and tracking for security vulnerabilities."

What entanglements might arise? Copilot users—here's one example, and another—have shown that Copilot can be induced to emit verbatim code from identifiable repositories. Just this week, Texas A&M professor Tim Davis gave numerous examples of large chunks of his code being copied verbatim by Copilot, including when he prompted Copilot with the comment `/* sparse matrix transpose in the style of Tim Davis */`.

> *Copilot leaves copyleft compliance as an exercise for the user. Users likely face growing liability that only increases as Copilot improves.*
> *—Bradley Kuhn*

Use of this code plainly creates an obligation to comply with its license. But as a side effect of Copilot's design, information about the code's origin—author, license, etc.—is stripped away. How can Copilot users comply with the license if they don't even know it exists?

Copilot's whizzy code-retrieval methods are a smokescreen intended to conceal a grubby truth: Copilot is merely a convenient alternative interface to a large corpus of open-source code. Therefore, Copilot users may incur licensing obligations to the authors of the underlying code. Against that backdrop, Nat Friedman's claim that Copilot operates "just like … a compiler" is rather dubious—compilers change the form of code, but they don't inject new intellectual-property entanglements. To be fair, Microsoft doesn't really dispute this. They just bury it in the fine print.

What does Copilot mean for open-source communities?

By offering Copilot as an alternative interface to a large body of open-source code, Microsoft is doing more than severing the legal relationship between open-source authors and users. Arguably, Microsoft is creating a new walled garden that will inhibit programmers from discovering traditional open-source communities. Or at the very least, remove any incentive to do so. Over time, this process will starve these communities. User attention and engagement will be shifted into the walled garden of Copilot and away from the open-source projects themselves—away from their source repos, their issue trackers, their mailing lists, their discussion boards. This shift in energy will be a painful, permanent loss to open source.

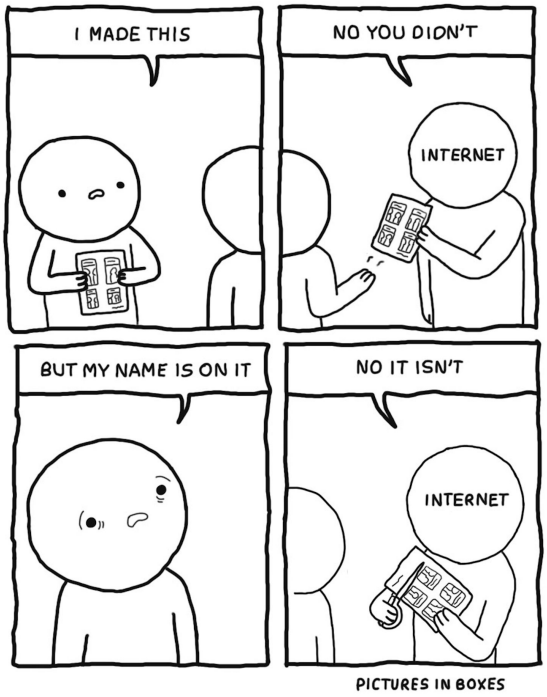
Don't take my word for it. Microsoft cloud-computing executive Scott Guthrie recently admitted that despite Microsoft CEO Satya Nadella's rosy pledge at the time of the GitHub acquisition that "GitHub will remain an open platform", Microsoft has been nudging more GitHub services—including Copilot—onto its Azure cloud platform.

> *Free software is not an unqualified gift … Copilot is a bad idea as designed. It represents a flagrant disregard of FOSS licensing …*
> *—Drew DeVault*

Obviously, open-source developers—me included—don't do it for the money, because no money changes hands. But we don't do it for nothing, either. A big benefit of releasing open-source software is the people: the com-

munity of users, testers, and contributors that coalesces around our work. Our communities help us make our software better in ways we couldn't on our own. This makes the work fun and collaborative in ways it wouldn't be otherwise.

Copilot introduces what we might call a more selfish interface to open-source software: **just give me what I want!** With Copilot, open-source users never have to know who made their software. They never have to interact with a community. They never have to contribute.



Meanwhile, we open-source authors have to watch as our work is stashed in a big code library in the sky called Copilot. The user feedback & contributions we were getting? Soon, all gone. Like Neo plugged into the Matrix, or a cow on a farm, Copilot wants to convert us into nothing more than producers of a resource to be extracted. (Well, until we can be disposed of entirely.)

And for what? Even the cows get food & shelter out of the deal. Copilot contributes nothing to our individual projects. And nothing to open source broadly.

The walled garden of Copilot is antithetical—and poisonous—to open source. It's therefore also a betrayal of everything GitHub stood for before being acquired by Microsoft. If you were born before 2005, you remember that GitHub built its reputation on its goodies for open-source developers and fostering that community. Copilot, by contrast, is the Multiverse-of-Madness inversion of this idea.

"Dude, it's cool. I took SFC's advice and moved my code off GitHub." So did I. Guess what? It doesn't matter. By claiming that AI training is fair use, Microsoft is constructing a justification for training on public code *anywhere* on the internet, not just GitHub. If we take this idea to its natural endpoint, we can predict that for end users, Copilot will become not just a substitute for open-source code on GitHub, but open-source code everywhere.

On the other hand, maybe you're a fan of Copilot who thinks that AI is the future and I'm just yelling at clouds. First, the objection here is not to AI-assisted coding tools generally, but to Microsoft's specific choices with Copilot. We can easily imagine a version of Copilot that's friendlier to open-source developers—for instance, where participation is voluntary, or where coders are paid to contribute to the training corpus. Despite its professed love for open source, Microsoft chose none of these options. Second, if you find Copilot valuable, it's largely because of the quality of the underlying open-source training data. As Copilot sucks the life from open-source projects, the proximate effect will be to make Copilot ever worse—a spiraling ouroboros of garbage code.

When I first wrote about Copilot, I said "I'm not worried about its effects on open source." In the short term, I'm still not worried. But as I reflected on my own journey through open source—nearly 25 years—I realized that I was missing the bigger picture. After all, open source isn't a fixed group of people. It's an ever-growing, ever-changing collective intelligence, continually being renewed by fresh minds. We set new standards and challenges for each other, and thereby raise our expectations for what we can accomplish.

Amidst this grand alchemy, Copilot interlopes. Its goal is to arrogate the energy of open-source to itself. We needn't delve into Microsoft's very checkered history with open source to see Copilot for what it is: a parasite.

The legality of Copilot must be tested before the damage to open source becomes irreparable. That's why I'm suiting up.

Help us investigate.

I'm currently working with the Joseph Saveri Law Firm to investigate a potential lawsuit against GitHub Copilot. We'd like to talk to you if—

→ You have stored open-source code on GitHub (in a public or private repo), or if you otherwise have reason to believe your code was used to train OpenAI's Codex or Copilot.

→ You own—or represent an entity that owns—one or more copyrights, patents, or other rights in open-source code.

→ You represent a group that advocates for open-source code creators.

→ You are a current or past GitHub Copilot user.

→ You have other information about Copilot you'd like to bring to our attention.

Any information provided will be kept in the strictest confidence as provided by law.

We look forward to hearing from you. You can contact me directly at mb@buttericklaw.com or use the form on the Joseph Saveri Law Firm website to reach the investigation team.

This web page is informational. General principles of law are discussed. But neither Matthew Butterick nor anyone at the Joseph Saveri Law Firm is *your* lawyer, and **nothing here is offered as legal advice**. References to copyright pertain to US law. This page will be updated as new information becomes available.