Joseph R. Saveri (State Bar No. 130064)
Cadio Zirpoli (State Bar No. 179108)
Christopher K.L. Young (State Bar No. 318371)
Louis A. Kessler (State Bar No. 243703)
Elissa A. Buchanan (State Bar No. 249996)
Travis Manfredi (State Bar No. 281779)
William W. Castillo Guardado (State Bar No. 294159)
Holden J. Benon (State Bar No. 325847)
JOSEPH SAVERI LAW FIRM, LLP
601 California Street, Suite 1000
San Francisco, California 94108
Telephone: (415) 500-6800
Facsimile: (415) 395-9940
Email: jsaveri@saverilawfirm.com
czirpoli@saverilawfirm.com
cyoung@saverilawfirm.com
lkessler@saverilawfirm.com
eabuchanan@saverilawfirm.com
tmanfredi@saverilawfirm.com
wcastillo@saverilawfirm.com
hbenon@saverilawfirm.com

Counsel for Individual and Representative Plaintiffs and the Proposed Class

[Additional Counsel Listed on Signature Page]

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
OAKLAND DIVISION

J. DOE 1, J. DOE 2, J. DOE 3, J. DOE 4, and J. DOE 5, individually and on behalf of all others similarly situated,

Individual and Representative Plaintiffs,

v.

GITHUB, INC., a Delaware corporation; MICROSOFT CORPORATION, a Washington corporation; OPENAI, INC., a Delaware nonprofit corporation; OPENAI, L.P., a Delaware limited partnership; OPENAI OPCO, L.L.C., a Delaware limited liability company; OPENAI GP, L.L.C., a Delaware limited liability company; OPENAI STARTUP FUND GP I, L.L.C., a Delaware limited liability company; OPENAI STARTUP FUND I, L.P., a Delaware limited partnership; OPENAI STARTUP FUND MANAGEMENT, LLC, a Delaware limited liability company; OPENAI, L.L.C. , a Delaware limited liability company; OPENAI GLOBAL, LLC , a Delaware limited liability company; OAI CORPORATION, a Delaware corporation; OPENAI HOLDINGS, LLC, a Delaware limited liability company; OPENAI HOLDCO, LLC, a Delaware limited liability company; OPENAI INVESTMENT

Case No.: 4:22-cv-06823-JST
4:22-cv-07074-JST

SECOND AMENDED COMPLAINT

CLASS ACTION

DEMAND FOR JURY TRIAL

*****FILED UNDER SEAL*****

[CONFIDENTIAL]

LLC, a Delaware limited liability company; OPENAI STARTUP FUND SPV I, L.P. , a Delaware limited partnership; and OPENAI STARTUP FUND SPV GP I, L.L.C. , a Delaware limited liability company;

Defendants.

TABLE OF CONTENTS

Plaintiffs J. Doe 1, J. Doe 2, J. Doe 3, J. Doe 4 and J. Doe 5 ("Plaintiffs"), on behalf of themselves and all others similarly situated, bring this Class Action Complaint (the "Complaint") against Defendants GitHub, Inc.; Microsoft Corporation; OpenAI, Inc.; OpenAI, L.P.; OpenAI OpCo, L.L.C.; OpenAI GP, L.L.C.; OpenAI Startup Fund GP I, L.L.C.; OpenAI Startup Fund I, L.P.; OpenAI Startup Fund Management, LLC; OpenAI, L.L.C.; OpenAI Global, LLC ("OpenAI Global"); OAI Corporation ("OAI"); OpenAI Holdings, LLC ("OpenAI Holdings"); OpenAI Holdco, LLC; OpenAI Investment L.L.C.; OpenAI Startup Fund SPV I, L.P.; and OpenAI Startup Fund SPV GP I, LLC[1] for violation of the Digital Millennium Copyright Act, 17 U.S.C. §§ 1201–1205 (the "DMCA"); breach of contract regarding the Suggested Licenses, and breach of contract regarding GitHub's policies including its terms of service.

I. OVERVIEW: A BRAVE NEW WORLD OF SOFTWARE PIRACY

1. Plaintiffs and Class members are owners of copyright interests in materials made available publicly on GitHub that are subject to various licenses containing conditions for use of those works (the "Licensed Materials"). All the licenses at issue here (the "Licenses") contain certain common terms (the "License Terms").

2. "Artificial Intelligence" is referred to herein as "AI." AI is defined for the purposes of this Complaint as a computer program that algorithmically simulates human reasoning or inference, often using statistical methods. Machine Learning ("ML") is a subset of AI in which the behavior of the program is derived from analyzing a corpus of material called training data.

3. GitHub is a company founded in 2008 by a team of open-source enthusiasts. At the time, GitHub's stated goal was to support open-source development, especially by hosting open-source source

[1] GitHub, Inc. is referred to as "GitHub." Microsoft Corporation is referred to as "Microsoft." OpenAI, Inc.; OpenAI, L.P.; OpenAI OpCo, L.L.C.; OpenAI GP, L.L.C.; OpenAI Startup Fund GP I, L.L.C.; OpenAI Startup Fund I, L.P.; OpenAI Startup Fund Management, LLC; OpenAI, L.L.C.; OpenAI Global; OAI Corporation ("OAI"); OpenAI Holdings; OpenAI Holdco, LLC; OpenAI Investment LLC; OpenAI Startup Fund SPV I, L.P.; and OpenAI Startup Fund SPV GP I, L.L.C. are referred to collectively herein as "OpenAI." Collectively, GitHub, Inc., Microsoft Corporation, OpenAI, Inc.; OpenAI, L.P.; OpenAI GP, L.L.C.; OpenAI Startup Fund GP I, L.L.C.; OpenAI Startup Fund I, L.P.; OpenAI Startup Fund Management, LLC; OpenAI, L.L.C.; OpenAI Global; OAI; OpenAI Holdings; OpenAI Holdco, LLC; OpenAI Investment LLC; OpenAI Startup Fund SPV I, L.P.; and OpenAI Startup Fund SPV GP I, L.L.C. are referred to herein as "Defendants."

code on the website github.com. Over the next 10 years, GitHub, based on these representations succeeded wildly, attracting nearly 25 million developers.

4. Developers published Licensed Materials on GitHub pursuant to written Licenses. In particular, the most popular ones share a common term: use of the Licensed Materials requires some form of *attribution*, usually by, among other things, including a copy of the license along with the name and copyright notice of the original author.

5. On October 26, 2018, Microsoft acquired GitHub for $7.5 billion. Though some members of the open-source community were skeptical of this union, Microsoft repeated one mantra throughout: "Microsoft Loves Open Source." For the first few years, Microsoft's representations seemed credible.

6. Microsoft invested $1 billion in OpenAI LP in July 2019 at a $20 billion valuation. In 2020, Microsoft became exclusive licensee of OpenAI's GPT-3 language model—despite OpenAI's continued claims its products are meant to benefit "humanity" at large. In 2021, Microsoft began offering GPT-3 through its Azure cloud-computing platform. On October 20, 2022, it was reported that OpenAI "is in advanced talks to raise more funding from Microsoft" at that same $20 billion valuation. Copilot runs on Microsoft's Azure platform. Microsoft has used Copilot to promote Azure's processing power, particularly regarding AI.

7. On information and belief, Microsoft obtained a partial ownership interest in OpenAI in exchange for its $1 billion investment. As OpenAI's largest investor and largest service provider—specifically in connection with Microsoft's Azure product—Microsoft exerts considerable control over OpenAI.

8. In June 2021, GitHub and OpenAI launched Copilot, an AI-based product that promises to assist software coders by providing or filling in blocks of code using AI. GitHub charges Copilot users $10 per month or $100 per year for this service. Copilot ignores, violates, and removes the Licenses offered by thousands—possibly millions—of software developers, thereby accomplishing software piracy on an unprecedented scale. Copilot outputs text derived from Plaintiffs' and the Class's Licensed Materials without adhering to the applicable License Terms and applicable laws. Copilot's output is referred herein as "Output."

9. On August 10, 2021, OpenAI debuted its Codex product, which converts natural language into code and is integrated into Copilot. Copilot and Codex can be called either AIs or MLs. Codex and Copilot will be referred to as Ais herein unless a distinction is required.

10. Though Defendants have been cagey about what data was used to train the AI,[2] they have conceded that the training data includes data in vast numbers of publicly accessible repositories on GitHub,[3] which include and are limited by Licenses.

11. Among other things, Defendants stripped Plaintiffs' and the Class's attribution, copyright notice, and license terms from their code in violation of the Licenses and Plaintiffs' and the Class's rights. Defendants used Copilot to distribute the now-anonymized code to Copilot users as if it were created by Copilot.

12. Copilot is run entirely on Microsoft's Azure cloud-computing platform.

13. Copilot often simply reproduces code that can be traced back to open-source repositories or open-source licensees. Contrary to and in violation of the Licenses, code reproduced by Copilot *never* includes attributions to the underlying authors.

14. GitHub and OpenAI have offered shifting accounts of the source and amount of the code or other data used to train and operate Copilot. They have also offered shifting justifications for why a commercial AI product like Copilot should be exempt from these license requirements, often citing "fair use."

15. It is not fair, permitted, or justified. On the contrary, Copilot's goal is to replace a huge swath of open source by taking it and keeping it inside a GitHub-controlled paywall. It violates the licenses that open-source programmers chose and monetizes their code despite GitHub's pledge never to do so.

[2] "Training" an AI, as described in greater detail below, means feeding it large amounts of data that it interprets using given criteria. Feedback is then given to it to fine-tune its Output until it can provide Output with minimal errors.

[3] Repositories are containers for individual coding projects. They are where GitHub users upload their code and where other users can find it. Most GitHub users have multiple repositories.

II. JURISDICTION AND VENUE

16. Plaintiffs bring this action on their own behalf as well as representatives of a Class of similarly situated individuals and entities. They seek to recover injunctive relief and damages as a result and consequence of Defendants' unlawful conduct.

17. Jurisdiction and venue are proper in this judicial district under 28 U.S.C. § 1331 pursuant to Defendants' violation of Section 1202(b) of the Digital Millennium Copyright Act, 17 U.S.C. §§ 1201–1205; and because a substantial part of the events giving rise to Plaintiffs' claims occurred in this District, a substantial portion of the affected interstate trade and commerce was carried out in this District, and three or more of the Defendants reside in this District and/or are licensed to do business in this District. Each Defendant has transacted business, maintained substantial contacts, and/or committed overt acts in furtherance of the illegal scheme and conspiracy throughout the United States, including in this District. Defendants' conduct has had the intended and foreseeable effect of causing injury to persons residing in, located in, or doing business throughout the United States, including in this District.

III. INTRADISTRICT ASSIGNMENT

18. Pursuant to Civil Local Rule 3.2 (c) and (e), assignment of this case to the San Francisco Division of the United States District Court for the Northern District of California is proper because a substantial amount of the development of the Copilot product as well as of the interstate trade and commerce involved and affected by Defendants' conduct giving rise to the claims herein occurred in this Division. Furthermore, Defendants GitHub and all the OpenAI entities are headquartered within this Division.

IV. PARTIES

A. Plaintiffs

19. Plaintiff J. Doe 1, [redacted], is a resident of the State of New Hampshire. Plaintiff Doe 1 published Licensed Materials they owned a copyright interest in to at least one GitHub repository under one of the Suggested Licenses. Specifically, Doe 1 has published Licensed Materials they claim a copyright interest in under the following Suggested Licenses: MIT License and GNU General Public

License version 3.0. Plaintiff was, and continues to be, injured during the Class Period as a result of Defendants' unlawful conduct alleged herein.

20. Plaintiff J. Doe 2, [redacted], is a resident of the State of Illinois. Plaintiff Doe 2 published Licensed Materials they owned a copyright interest in to at least one GitHub repository under one of the Suggested Licenses. Specifically, Doe 2 has published Licensed Materials they claim a copyright interest in under the following Suggested Licenses: MIT License; GNU General Public License version 3.0; GNU Affero General Public License version 3.0; The 3-Clause BSD License; and Apache License 2.0. Plaintiff was, and continues to be, injured during the Class Period as a result of Defendants' unlawful conduct alleged herein.

21. Plaintiff J. Doe 3, [redacted], is a resident of the State of Idaho. Plaintiff Doe 3 published Licensed Materials they owned a copyright interest in to at least one GitHub repository under one of the Suggested Licenses. Specifically, Doe 3 has published Licensed Materials they claim a copyright interest in under the following Suggested Licenses: MIT License; GNU General Public License version 3.0; and GNU Affero General Public License version 3.0. Plaintiff was, and continues to be, injured during the Class Period as a result of Defendants' unlawful conduct alleged herein.

22. Plaintiff J. Doe 4, [redacted], is a resident of the State of South Carolina. Plaintiff Doe 4 published Licensed Materials they owned a copyright interest in to at least one GitHub repository under one of the Suggested Licenses. Specifically, Doe 4 has published Licensed Materials they claim a copyright interest in under the following Suggested Licenses: GNU General Public License v2.0 and GNU General Public License v3.0. Plaintiff was, and continues to be, injured during the Class Period as a result of Defendants' unlawful conduct alleged herein.

23. Plaintiff J. Doe 5, [redacted], is a resident of the Commonwealth of Massachusetts. Plaintiff Doe 5 published Licensed Materials they owned a copyright interest in to at least one GitHub repository under one of the Suggested Licenses. Specifically, Doe 5 has published Licensed Materials they claim a copyright interest in under the following Suggested Licenses: MIT License; Apache License 2.0; and GNU General Public License v3.0.

B. Defendants

24. Defendant GitHub, Inc. is a Delaware corporation with its principal place of business located at 88 Colin P Kelly Jr Street, San Francisco, CA 94107. GitHub sells, markets, and distributes Copilot throughout the internet and other sales channels throughout the United States, including in this District. GitHub released Copilot on a limited "technical preview" basis on June 29, 2021. On June 21, 2022, Copilot was released to the public as a subscription-based service for individual developers. GitHub is a party to the unlawful conduct alleged herein.

25. Defendant Microsoft Corporation is a Washington corporation with its principal place of business located at One Microsoft Way, Redmond, Washington 98052. Microsoft announced its acquisition of Defendant GitHub, Inc. on June 4, 2018. On October 26, 2018, Microsoft finalized its acquisition of GitHub. Microsoft owns and operates GitHub. Through its corporate ownership, control of the GitHub Board of Directors, active management, and other means, Microsoft sells, markets, and distributes Copilot. Microsoft is a party to the unlawful conduct alleged herein.

26. Defendant OpenAI, Inc. is a Delaware nonprofit corporation with its principal place of business located at 3180 18th Street, San Francisco, CA 94110. OpenAI, Inc. is a party to the unlawful conduct alleged herein. It—along with OpenAI, L.P.—programed, trained, and maintains Codex, which infringes all the same rights at Copilot and is also an integral piece of Copilot. Copilot requires Codex to function. OpenAI, Inc. is a party to the unlawful conduct alleged herein. OpenAI, Inc. founded, owns, and exercises control over all the other OpenAI entities, including those set forth in Paragraphs 27–40.

27. Defendant OpenAI, L.P. is a Delaware limited partnership with its principal place of business located at 3180 18th Street, San Francisco, CA 94110. OpenAI, L.P. is a party to the unlawful conduct alleged herein. Its primary activity is research and technology. OpenAI, L.P. is a wholly owned subsidiary of OpenAI, Inc. that is operated for profit. OpenAI, L.P. is the OpenAI entity that co-created Copilot and offers it jointly with GitHub. OpenAI's revenue, including revenue from Copilot, is received by OpenAI, L.P. OpenAI, Inc. controls OpenAI, L.P. directly and through the other OpenAI entities.

28. Defendant OpenAI OpCo, L.L.C. is a Delaware limited liability company with its principal place of business located at 3180 18th Street, San Francisco, CA 94110. OpenAI OpCo, L.L.C. is a party to the unlawful conduct alleged herein. Its primary activity is research and technology. OpenAI OpCo,

L.L.C. is a wholly owned subsidiary of OpenAI, Inc. that is operated for profit. OpenAI OpCo, L.L.C. is the OpenAI entity that co-created Copilot and offers it jointly with GitHub. OpenAI's revenue, including revenue from Copilot, is received by OpenAI OpCo, L.L.C. OpenAI, Inc. controls OpenAI OpCo, L.L.C. directly and through the other OpenAI entities.

29. Defendant OpenAI GP, L.L.C. ("OpenAI GP") is a Delaware limited liability company with its principal place of business located at 3180 18th Street, San Francisco, CA 94110. OpenAI GP is the general partner of OpenAI, L.P. OpenAI GP manages and operates the day-to-day business and affairs of OpenAI, L.P. OpenAI GP is liable for the debts, liabilities and obligations of OpenAI, L.P., including litigation and judgments. OpenAI GP is a party to the unlawful conduct alleged herein. Its primary activity is research and technology. OpenAI GP is the general partner of OpenAI, L.P. OpenAI GP was aware of the unlawful conduct alleged herein and exercised control over OpenAI, L.P. throughout the Class Period. OpenAI, Inc. directly controls OpenAI GP. OpenAI GP directly controls OpenAI Holdings and OpenAI Global.

30. Defendant OpenAI Startup Fund I, L.P. ("OpenAI Startup Fund I") is a Delaware limited partnership with its principal place of business located at 3180 18th Street, San Francisco, CA 94110. OpenAI Startup Fund I was instrumental in the foundation of OpenAI, L.P., including the creation of its business strategy and providing initial funding. Through participation in OpenAI Startup Fund I, certain entities and individuals obtained an ownership interest in OpenAI, L.P. Plaintiffs are informed and believed, and on that basis allege that OpenAI Startup Fund I participated in the organization and operation of OpenAI, L.P. OpenAI Startup Fund I is a party to the unlawful conduct alleged herein. OpenAI Startup Fund I was aware of the unlawful conduct alleged herein and exercised control over OpenAI, L.P. throughout the Class Period.

31. Defendant OpenAI Startup Fund GP I, L.L.C. ("OpenAI Startup Fund GP I") is a Delaware limited liability company with its principal place of business located at 3180 18th Street, San Francisco, CA 94110. OpenAI Startup Fund GP I is the general partner of OpenAI Startup Fund I. OpenAI Startup Fund GP I manages and operates the day-to-day business and affairs of OpenAI Startup Fund I. OpenAI Startup Fund GP I is liable for the debts, liabilities and obligations of OpenAI Startup Fund I, including litigation and judgments. OpenAI Startup Fund GP I was aware of the unlawful conduct

alleged herein and exercised control over OpenAI, L.P. throughout the Class Period. OpenAI Startup Fund GP I is a party to the unlawful conduct alleged herein. Sam Altman, co-founder, CEO, and Board member of OpenAI, Inc. is the Manager of OpenAI Startup Fund GP I. OpenAI Startup Fund GP I is the General Partner of OpenAI Startup Fund I, L.P.

32. Defendant OpenAI Startup Fund Management, LLC ("OpenAI Startup Fund Management") is a Delaware limited liability company with its principal place of business located at 3180 18th Street, San Francisco, CA 94110. OpenAI Startup Fund Management is a party to the unlawful conduct alleged herein. OpenAI Startup Fund Management was aware of the unlawful conduct alleged herein and exercised control over OpenAI, L.P. throughout the Class Period.

33. Defendant OpenAI, L.L.C. is a Delaware limited liability company with its principal place of business in San Francisco, California. OpenAI LLC owns some or all of the services and products provided by OpenAI. The sole member of OpenAI, L.L.C. is Defendant OpenAI OpCo, L.L.C.

34. Defendant OpenAI Global, LLC is a Delaware limited liability company with its principal place of business in San Francisco, California. OpenAI Global's only members are Microsoft and Defendant OAI Corporation. Microsoft owns 49% of OpenAI Global, and exercises control over it as its largest minority shareholder. OpenAI describes OpenAI Global as a "capped profit company"),

35. Defendant OAI Corporation ("OAI") is a Delaware corporation with its principal place of business in San Francisco, California. OAI's only member is Defendant OpenAI Holdings LLC.

36. Defendant OpenAI Holdings, LLC is a Delaware limited liability company with its principal place of business in San Francisco, California. The members of OpenAI Holdings are Defendant OpenAI, Inc. and Aestas LLC, an OpenAI-related limited liability company that is not named as a defendant as of December 22, 2023. OpenAI Holdings is partially owned by OpenAI employees and outside investors.

37. Defendant OpenAI Holdco, LLC is a Delaware limited liability company with its principal place of business in San Francisco, California.

38. Defendant OpenAI Investment LLC is a Delaware limited liability company with its principal place of business in San Francisco, California.

39. Defendant OpenAI Startup Fund SPV I, L.P. is a Delaware limited partnership with its principal place of business in San Francisco, California.

40. Defendant OpenAI Startup Fund SPV GP I, L.L.C. is a Delaware limited liability company with its principal place of business in San Francisco, California. OpenAI Startup Fund SPV GP I, L.L.C. is the general partner and controls OpenAI Startup Fund SPV I, L.P.

V. AGENTS AND CO-CONSPIRATORS

41. The unlawful acts alleged against the Defendants in this class action complaint were authorized, ordered, or performed by the Defendants' respective officers, agents, employees, representatives, or shareholders while actively engaged in the management, direction, or control of the Defendants' businesses or affairs.

42. The Defendants' agents operated under the explicit and apparent authority of their principals.

43. Each Defendant, and its subsidiaries, affiliates and agents operated as a single unified entity.

44. Various persons and/or firms not named as Defendants herein may have participated as coconspirators in the violations alleged herein and may have performed acts and made statements in furtherance thereof.

45. Each acted as the principal, agent, or joint venture of, or for other Defendants with respect to the acts, violations, and common course of conduct alleged herein.

VI. CLASS ALLEGATIONS

A. Class Definitions

46. Plaintiffs bring this action for damages and injunctive relief on behalf of themselves and all others similarly situated as a class action pursuant to Rules 23(a), 23(b)(2), and 23(b)(3) of the Federal Rules of Civil Procedure, on behalf of the following Classes:

"Injunctive Relief Class" under Rule 23(b)(2):

> All persons or entities domiciled in the United States that, (1) owned an interest in at least one US copyright in any work; (2) offered that work under one of GitHub's Suggested Licenses[4]; and (3) stored Licensed Materials in any public GitHub repositories at any time between January 1, 2015 and the present (the "Class Period").

"Damages Class" under Rule 23(b)(3):

> All persons or entities domiciled in the United States that, (1) owned an interest in at least one US copyright in any work; (2) offered that work under one of GitHub's Suggested Licenses; and (3) stored Licensed Materials in any public GitHub repositories at any time during the Class Period.

These "Class Definitions" specifically exclude the following person or entities:

a. Any of the Defendants named herein;

b. Any of the Defendants' co-conspirators;

c. Any of Defendants' parent companies, subsidiaries, and affiliates;

d. Any of Defendants' officers, directors, management, employees, subsidiaries, affiliates, or agents;

e. All governmental entities; and

f. The judges and chambers staff in this case, as well as any members of their immediate families.

[4] When a GitHub user creates a new repository, they have the option of selecting one of thirteen licenses from a dropdown menu to apply to the contents of that repository. (They can also apply a different license later, or no license.) The Creative Commons Zero v1.0 Universal and the Unlicense donate the covered work to the public domain and/or otherwise waive all copyrights and related rights. Because they do not contain the necessary provisions nor do they even allow the owner to make copyright claims in most circumstances, they are not included in the Class Definition. We refer to the remaining eleven options as the "Suggested Licenses," which are: (1) Apache License 2.0 ("Apache 2.0"); (2) GNU General Public License version 3 ("GPL-3.0"); (3) MIT License ("MIT"); (4) The 2-Clause BSD License ("BSD 2"); (5) The 3-Clause BSD License ("BSD 3"); (6) Boost Software License ("BSL-1.0"); (7) Eclipse Public License 2.0 ("EPL-2.0"); (8) GNU Affero General Public License version 3 ("AGPL-3.0"); (9) GNU General Public License version 2 ("GPL-2.0"); (10) GNU Lesser General Public License version 2.1 ("LGPL-2.1"); and (11) Mozilla Public License 2.0 ("MPL-2.0"). These Suggested Licenses each contain at least three common requirements for use of the Licensed Materials in a derivative work or copy: attribution to the owner of the Licensed Materials ("Attribution"), inclusion of a copyright notice ("Copyright Notice"), and inclusion of the applicable Suggested License's text ("License Terms").

B. Numerosity

47. Plaintiffs do not know the exact number of Class members, because such information is in the exclusive control of Defendants. Plaintiffs are informed and believe that there are at least thousands of Class members geographically dispersed throughout the United States such that joinder of all Class members in the prosecution of this action is impracticable.

C. Typicality

48. Plaintiffs' claims are typical of the claims of their fellow Class members because Plaintiffs and Class members all own code published under a License. Plaintiffs and the Class published work subject to a License to GitHub later used by Copilot. Plaintiffs and absent Class members were damaged by this and other wrongful conduct of Defendants as alleged herein. Damages and the other relief sought herein is common to all members of the Class.

D. Commonality & Predominance

49. Numerous questions of law or fact common to the entire Class arise from Defendants' conduct—including, but not limited to those identified below:

1. DMCA Violations

- Whether Defendants' conduct violated the Class's rights under the DMCA when GitHub and OpenAI caused Codex and Copilot to ingest and distribute Licensed Materials without including any associated Attribution, Copyright Notice, or License Terms.

2. Contract-Related Conduct

- Whether Defendants violated the Licenses governing use of the Licensed Materials by using them to train Copilot and for republishing those materials without appending the required Attribution, Copyright Notice, or License Terms.
- Whether Defendants interfered in prospective economic relations between the Class and the public regarding the Licensed Materials by concealing the License Terms.
- Whether Defendants intentionally or negligently interfered with a prospective economic advantage.

3. Injunctive Relief

- Whether this Court should enjoin Defendants from engaging in the unlawful conduct alleged herein. And what the scope of that injunction would be.

4. Defenses

- Whether any affirmative defense excuses Defendants' conduct.
- Whether any statutes of limitation limit Plaintiffs' and the Class's potential for recovery.
- Whether any applicable statutes of limitation should be tolled as a result of Defendants' fraudulent concealment of their unlawful conduct.

50. These and other questions of law and fact are common to the Class and predominate over any questions affecting the Class members individually.

E. Adequacy

51. Plaintiffs will fairly and adequately represent the interests of the Class because they have experienced the same harms as the Class and have no conflicts with any other members of the Class. Furthermore, Plaintiffs have retained sophisticated and competent counsel ("Class Counsel") who are experienced in prosecuting Federal and state class actions throughout the United States and other complex litigation and have extensive experience advising clients and litigating intellectual property, competition, contract, and privacy matters.

F. Other Class Considerations

52. Defendants have acted on grounds generally applicable to the Class, thereby making final injunctive relief appropriate with respect to the Class as a whole.

53. This class action is superior to alternatives, if any, for the fair and efficient adjudication of this controversy. Prosecuting the claims pleaded herein as a class action will eliminate the possibility of repetitive litigation. There will be no material difficulty in the management of this action as a class action.

54. The prosecution of separate actions by individual Class members would create the risk of inconsistent or varying adjudications, establishing incompatible standards of conduct for Defendants.

VII. FACTUAL ALLEGATIONS

A. Introduction

55. This class action against Defendants concerns an OpenAI product called Codex and a GitHub product called Copilot.

56. OpenAI began development of Codex sometime after OpenAI was founded in December 2015 and released Codex on a limited basis in August 2021.

57. GitHub began development of Copilot sometime in 2019, released it on a limited basis in June 2021, and released it as a paid subscription service in June 2022.

58. Codex and Copilot are assistive AI-based systems offered to software programmers. These AI systems are each trained on a large corpus of publicly accessible software code and other materials, including all the Licensed Materials. Defendants describe these products as making coding easier by accepting a code "prompt" from a programmer and emitting a possible completion of that code, which is referred to herein as Output, as set forth above. Defendants claim Codex and Copilot do not retain copies of the materials they are trained on. In practice, however, the Output is often a near-identical reproduction of code from the training data.

59. Codex and Copilot are related. Copilot is a joint venture between GitHub and OpenAI. Codex is a standalone product released by OpenAI that also "powers GitHub Copilot, which [OpenAI] built and launched in partnership with GitHub."[5] "GitHub Copilot uses the OpenAI Codex to suggest code and entire functions in real-time, right from your editor."[6] The version of Codex used to power Copilot is modified from OpenAI's standalone version. Defendants have kept secret the details of Codex's modifications and its integration into or interaction with Copilot.

B. Codex Outputs Copyrighted Materials Without Following the Terms of the Applicable Licenses

60. Below is an explanation of how Codex functions. When Codex is prompted with:

```
function isEven(n) {
```

[5] *See* https://openai.com/blog/openai-codex/.

[6] *See* https://github.com/features/copilot/.

it assumes this is the beginning of a function written in the JavaScript language that will test whether a number is even.

61. Based on this assumption, Codex will then provide Output meant to complete the rest of the function. Based on the given prompt, it produced the following response:[7]

```
function isEven(n) {
 if (n == 0)
   return true;
 else if (n == 1)
   return false;
 else if (n < 0)
   return isEven(-n);
 else
   return isEven(n - 2);
}
console.log(isEven(50));
// → true
console.log(isEven(75));
// → false
console.log(isEven(-1));
// → ??
```

62. The function itself occupies the first ten lines. Six additional lines follow the function, beginning with "console.log(isEven(50))". One possible explanation for Codex's inclusion of these lines is to test the "isEven" function. Though not part of the function itself, the lines will confirm the function works for certain values. In this case, the code implies that "isEven(50)" should return the value "true", and "isEven(75)" should return "false". Those answers are correct.

63. The penultimate line indicates "isEven(-1)" should return "??". This is an error, as "isEven(-1)" should return "false".

64. Codex cannot and does not understand the meaning of software code or any other Licensed Materials. But in training, what became Codex was exposed to an enormous amount of existing software code (its "Training Data") and—with input from its trainers and its own internal processes—

[7] Due to the nature of Codex, Copilot, and AI in general, Plaintiffs cannot be certain these examples would produce the same results if attempted following additional trainings of Codex and/or Copilot. However, these examples are representative of Codex and Copilot's Output at the time just prior to the filing of this Complaint.

inferred certain statistical patterns governing the structure of code and other Licensed Materials. The finished version of Codex, once trained, is known as a "Model."

65. When given a prompt, such as the initial prompt discussed above—"function isEven(n) {"—Codex identifies the most statistically likely completion, based on the examples it reviewed in training. Every instance of Output from Codex is derived from material in its Training Data. Most of its Training Data consisted of Licensed Materials.

66. Codex does not "write" code the way a human would, because it does not understand the meaning of code. Codex's lack of understanding of code is evidenced when it emits extra code that is not relevant under the circumstances. Here, Codex was only prompted to produce a function called "isEven". To produce its answer, Codex relied on Training Data that also appended the extra testing lines. Having encountered this function and the follow-up lines together frequently, Codex extrapolates they are all part of one function. A human with even a basic understanding of how JavaScript works would know the extra lines are not part of the function itself.

67. Beyond the superfluous and inaccurate extra lines, this "isEven" function also contains two major defects. First, it assumes the variable "n" holds an integer. It could contain some other kind of value, like a decimal number or text string, which would cause an error. Second, even if "n" does hold an integer, the function will trigger a memory error called a "stack overflow" for sufficiently large integers. For these reasons, experienced programmers would not use Codex's Output.

68. Codex does not identify the owner of the copyright to this Output, nor any other—it has not been trained to provide Attribution. Nor does it include a Copyright Notice nor any License Terms attached to the Output. This is by design—Codex was not coded or trained to track or reproduce such data. The Output in the example above is taken from *Eloquent JavaScript* by Marijn Haverbeke.[8]

69. Here is the exercise from *Eloquent JavaScript*:

```
// Your code here.
```

[8] https://eloquentjavascript.net/code/#3.2. *Eloquent JavaScript* is "Licensed under a Creative Commons [A]ttribution-[N]oncommercial license. All code in this book may also be considered licensed under an MIT license." *See* https://eloquentjavascript.net/. Thus, having also been posted on GitHub, the code Codex relied on meets the definition of Licensed Materials.

```
console.log(isEven(50));
// → true
console.log(isEven(75));
// → false
console.log(isEven(-1));
// → ??
```

70. The exercise includes the "??" error. However, for Haverbeke's purposes, this is not an error but a placeholder value for the reader to fill in. Codex—as a mere probabilistic model—fails to recognize this nuance. The inclusion of the double question marks confirms unequivocally that Codex took this code directly from a copyrighted source without following any of the attendant License Terms.

71. Haverbeke provides the following solution to the function discussed above:

```
function isEven(n) {
  if (n == 0) return true;
  else if (n == 1) return false;
  else if (n < 0) return isEven(-n);
  else return isEven(n - 2);
}

console.log(isEven(50));
// → true
console.log(isEven(75));
// → false
console.log(isEven(-1));
// → false
```

72. Aside from different line breaks—which are not semantically meaningful in JavaScript—this code for the function "isEven" is the same as what Codex produced. The tests are also the same, though in this case Haverbeke provides the right answer for "isEven(-1)", which is "false". Codex has reproduced Haverbeke's Licensed Material almost verbatim, with the only difference being drawn from a different portion of those same Licensed Materials.

73. There are many copies of Haverbeke's code stored in public repositories on GitHub, where programmers who are working through Haverbeke's book store their answers.

74. The MIT license provides that "The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software."[9] Any person taking this code

[9] *See* Appendix A for full text of the MIT License.

directly from *Eloquent JavaScript* would have direct access to these License Terms and know to follow them if incorporating the Licensed Materials into a derivative work and/or copying them. Codex does not provide these License Terms.

75. OpenAI Codex's Output would frequently, perhaps even constantly, contain Licensed Materials, i.e., it would have conditions associated with it through its associated license. In its 2021 research paper about Codex called "Evaluating Large Language Models Trained on Code," OpenAI stated Codex's Output is "often incorrect" and can contain security vulnerabilities and other "misalignments" (meaning, departures from what the user requested).

76. Most open-source licenses require attribution of the author, notice of their copyright, and a copy of the license specifically to ensure that future coders can easily credit all previous authors and ensure they adhere to all applicable licenses. All the Suggested Licenses include these requirements.

77. Ultimately, Codex derives its value primarily from its ability to locate and output potentially useful Licensed Materials. And from its obfuscation of any rights associated with those materials.

C. Copilot Outputs Copyrighted Materials Without Following the Terms of the Applicable Licenses

78. GitHub Copilot works in a similar way to OpenAI Codex. As mentioned above, a modified version of Codex is used as the engine that powers Copilot.

79. Copilot is installed by the end user as an extension to various code editors, including Microsoft's Visual Studio and VS Code. As the user types into the editor, their code is uploaded in real time to Microsoft's Azure cloud platform, where they become prompts for Copilot.

80. When we give Copilot the same prompt discussed above in Paragraph 61, "function isEven(n) {", it interprets the prompt as the beginning of a function written in the JavaScript language that will test whether a number is even, same as Codex.

81. However, the Output of Copilot in response to the prompt is different than Codex, namely:

```
function isEven(n) {
  return n % 2 === 0;
}
```

82. This function is much closer to what a human programmer might use as compared to Codex's older, inaccurate offering. It handles all values and types of "n" correctly. It does not cause a stack overflow for larger values of "n" like the Codex Output.

83. Copilot's Output, like Codex's, is derived from existing code. Namely, sample code that appears in the online book *Mastering JS*, written by Valeri Karpov.[10] Like Codex's Output, Copilot's is also based upon copyrighted educational material. *Mastering JS* is a set of educational exercises for programmers. Like *Eloquent JavaScript*, there are many copies of Karpov's exercise stored in public repositories on GitHub. Programmers working through *Mastering JS* store their answers there.

84. If Copilot is prompted with the name of a function that will test whether a number is prime (that is, a number that can only be evenly divided by 1 and itself), namely "function isPrime(n) {", it returns:

```
function isPrime(n) {
  if (n < 2) {
    return false;
  }
  for (let i = 2; i < n; i++) {
    if (n % i === 0) {
      return false;
    }
  }
  return true;
}
```

85. Though this function will work, it contains an error often made by beginner programmers that makes it much slower than it could be. Namely, the loop in the middle, which checks possible divisors, does not need to check every divisor smaller than "n," only the divisors smaller than the square root of "n". As with Codex, Copilot has no understanding of how the code works. It knows that more functions called "isPrime" contain the portion that checks for all divisors smaller than "n", so that is what it offers. It does not return what it "thinks" is best, it returns what it has seen *the most*. It is not writing, it is reproducing (i.e., copying).

[10] https://masteringjs.io/tutorials/fundamentals/modulus/.

86. Like the other examples above—and most of Copilot's Output—this output is nearly a verbatim copy of copyrighted code. In this case, it is substantially similar to the "isPrime" function in the book *Think JavaScript* by Matthew X. Curinga et al,[11] which is:

```
function isPrime(n) {
  if (n < 2) {
    return false;
  }
  for (let i = 2; i < n; i++) {
    if (n % i === 0) {
      return false;
    }
  }
  return true;
}
```

87. As with the other examples above, the source of Copilot's Output is a programming textbook. Also like the books the other examples were taken from, there are many copies of Curinga's code stored in public repositories on GitHub where programmers who are working through Curinga's book keep copies of their answers.

88. The material in Curinga's book is made available under the GNU Free Documentation License. Although this is not one of the Suggested Licenses, it contains similar attribution provisions, namely that "You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License."[12]

89. As with Codex, Copilot does not provide the end user any attribution of the original author of the code, nor anything about their license requirements. There is no way for the Copilot user to know that they must provide attribution, copyright notice, nor a copy of the license's text. And with regard to the GNU Free Documentation License, Copilot users would not be aware that they are limited in what conditions they can place on the use of derivative works they make using this copyrighted code. Had the

[11] https://matt.curinga.com/think-js/#solving-problems-with-for-loops.

[12] https://matt.curinga.com/think-js/#gnu-free-documentation-license.

Copilot user found this code in a public GitHub repository or a copy of the book it was originally published in, they would find the GNU Free Documentation License at the same time and be aware of its terms. Copilot finds that code for the user but excises the license terms, copyright notice, and attribution. This practice allows its users to assume that the code can be used without restriction. It cannot.

D. Codex and Copilot Were Trained on Copyrighted Materials Offered Under Licenses

90. Codex is an AI system. Another way to describe it is a "model." Without Codex, Copilot, or another AI-code-lookup-tool, code is written both by originating code from the writer's own knowledge of how to write code as well as by finding pre-written portions of code that—under the terms of the applicable license—may be incorporated into the coding project.

91. Unlike a human programmer that has learned how code works and notices when code it is copying has attached license terms, a copyright notice, and/or attribution, Codex and Copilot were developed by feeding a corpus of material, called "training data," into them. These AI programs ingest all the data and, through a complex probabilistic process, predict what the most likely solution to a given prompt a user would input is. Though more complicated in practice, essentially Copilot returns the solution it has found in the most projects when those projects are somehow weighted to adjust for whatever variables Codex or Copilot have identified as relevant.

92. Codex and Copilot were not programmed to treat attribution, copyright notices, and license terms as legally essential. Defendants made a deliberate choice to expedite the release of Copilot rather than ensure it would not provide unlawful Output.

93. The words "study" and "training" and "learning" in connection with AI describe algorithmic processes that are not analogous to human reasoning. AI models cannot "learn" as humans do, nor can it "understand" semantics and context the way humans do. Rather, it detects statistically significant patterns in its training data and provides Output derived from its training data when statistically appropriate. A "brute force" approach like this would not be efficient nor even possible for humans. A human could not memorize, statistically analyze, and easily access thousands of gigabytes of existing code, a task now possible for powerful computers like those that make up Microsoft's Azure cloud platform. To accomplish the same task, a human may search for Licensed Materials that serve their purpose if they believe such materials exist. And if that human finds such materials, they will probably

abide by its License Terms rather than risk infringing its owners' rights. At the very least, if they incorporate those Licensed Materials into their own project without following its terms they will be doing so knowingly.

E. Copilot Was Launched Despite Its Propensity for Producing Unlawful Outputs

94. GitHub and OpenAI have not provided much detail regarding what data Codex and OpenAI were trained on. Plaintiffs know for certain from GitHub and OpenAI's statements, that both systems were trained on publicly available GitHub repositories, with Copilot having been trained on all available public GitHub repositories.

95. According to OpenAI, Codex was trained on "billions of lines of source code from publicly available sources, including code in public GitHub repositories." Similarly, GitHub has described[13] Copilot's training material as "billions of lines of public code." GitHub researcher Eddie Aftandilian confirmed in a recent podcast[14] that Copilot is "train[ed] on public repos on GitHub."

96. In a recent customer-support message, GitHub's support department clarified certain facts about training Copilot. First, GitHub said that "training for Codex (the model used by Copilot) is done by OpenAI, not GitHub." Second, in its support message, GitHub put forward a more detailed justification for its use of copyrighted code as training data:

> Training machine learning models on publicly available data is considered fair use across the machine learning community . . . OpenAI's training of Codex is done in accordance with global copyright laws which permit the use of publicly accessible materials for computational analysis and training of machine learning models, and do not require consent of the owner of such materials. Such laws are intended to benefit society by enabling machines to learn and understand using copyrighted works, much as humans have done throughout history, and to ensure public benefit, these rights cannot generally be restricted by owners who have chosen to make their materials publicly accessible.

The claim that training ML models on publicly available code is widely accepted as fair use is not true. And regardless of this concept's level of acceptance in "the machine learning community," under Federal law, it is illegal.

[13] https://github.blog/2021-06-30-github-copilot-research-recitation/.

[14] https://www.se-radio.net/2022/10/episode-533-eddie-aftandilian-on-github-copilot/.

97. Former GitHub CEO Nat Friedman said in June 2021—when Copilot was released to a limited number of customers—that "training ML systems on public data is fair use."[15] Friedman's statement is pure speculation; no Court has considered the question of whether "training ML systems on public data is fair use." The Fair Use affirmative defense is only applicable to Section 501 copyright infringement. It is not a defense to violations of the DMCA, breach of contract, nor any other claim alleged herein. It cannot be used to avoid liability here. At the same time Friedman asserted "the output [of Copilot] belongs to the operator."

98. Other open-source stakeholders have made this point already. For example, in June 2021, Software Freedom Conservancy ("SFC"), a prominent open-source advocacy organization, asked Microsoft and GitHub to provide "legal references for GitHub's public legal positions." No references were provided by any of the Defendants.[16]

99. Beyond the examples above, Copilot regularly Output's verbatim copies of Licensed Materials. For example, Copilot reproduced verbatim well-known code from the game Quake III, use of which is governed by one of the Suggested Licenses—GPL-2.[17]

100. Copilot also reproduced code that had been released under a license that allowed its use only for free games and required attribution by including a copy of the license. Copilot did not mention nor include the underlying license when providing a copy of this code as Output.[18]

101. Texas A&M computer-science professor Tim Davis has provided numerous examples of Copilot reproducing code belonging to him without its license or attribution.[19]

102. GitHub concedes that in ordinary use, Copilot will reproduce identical passages of code, verbatim: "Our latest internal research shows that about 1% of the time, a suggestion [Output] may contain some code snippets longer than ~150 characters that matches" code from the training data. This standard is more limited than is necessary for copyright infringement. But even using GitHub's own

[15] https://twitter.com/natfriedman/status/1409914420579344385/.

[16] https://sfconservancy.org/blog/2022/feb/03/github-copilot-copyleft-gpl/.

[17] https://twitter.com/stefankarpinski/status/1410971061181681674/.

[18] https://twitter.com/ChrisGr93091552/status/1539731632931803137/.

[19] https://twitter.com/DocSparse/status/1581461734665367554/.

metric and the most conservative possible criteria, Copilot has violated the DMCA at least tens of thousands of times.

103. In June 2022, Copilot had 1,200,000 users. If only 1% of users have ever received Output based on Licensed Materials and only once each, Defendants have "only" breached Plaintiffs' and the Class's Licenses 12,000 times. However, each time Copilot outputs Licensed Materials without attribution, the copyright notice, or the License Terms it violates the DMCA three times. Thus, even using this extreme underestimate, Copilot has "only" violated the DMCA 36,000 times. Because Copilot constantly Outputs code as a user writes, and because nearly all of Copilot's training data was Licensed Material, this number is most likely exponentially lower than the true number of breaches and DMCA violations.

104. Academics are continuing to study generative AI models and their behavior. Recent academic research shows that the likelihood Plaintiffs' or class members' code would be emitted verbatim is only increasing. For instance the study, *Quantifying Memorization Across Neural Language Models* by Nicholas Carlini et al.,[20] tested multiple models by feeding prefixes of prompts based on training data into each model in order to compare the performance of models of different sizes to emit output that is identical to training data. The study concluded:

> Large language models (LMs) have been shown to memorize parts of their training data, and when prompted appropriately, they will emit the memorized training data verbatim. … We describe three [mathematical] relationships that quantify the degree to which LMs emit memorized training data. **Memorization significantly grows as we increase (1) the capacity of a model**, (2) the number of times an example has been duplicated, and (3) the number of tokens of context used to prompt the model. Surprisingly, we find the situation becomes more complicated when generalizing these results across model families. On the whole, we find that memorization in LMs is more prevalent than previously believed and will likely get worse as models continues to scale.

(Emphasis added). Or as simply put by the study, "Bigger Models Memorize More."

[20] https://arxiv.org/pdf/2202.07646.pdf

105. In other words, as generative AI models such as Copilot increase capacity and continue to scale, it becomes **more likely** that training data will become memorized and emitted verbatim, i.e., as an exact duplicate.

106. Given GitHub's increasing commitment to Copilot, and the scale of growth of Copilot, it follows that Copilot is more likely to emit duplicates of memorized training data as it continues to scale.

107. Indeed, GitHub has announced that it is adopting OpenAI's GPT-4 model for Copilot, which is a bigger and more capable language model than the Codex-derived model.

108. Furthermore, the Suggested Licenses impose attribution obligations not only when Licensed Materials have been used verbatim, but also when Licensed Materials have been modified or adapted. Though Output from Copilot is often a verbatim, i.e., identical copy, even more often it is a modification: for instance, a near-identical copy that contains only semantically insignificant variations of the original Licensed Materials, or a modified copy that recreates the same algorithm. Whenever Copilot outputs Licensed Materials in a manner that qualifies as a modification, the attribution requirements of the Suggested Licenses still apply. Copilot's failure to provide the attributions for outputs that are modifications of Licensed Materials represents another enormous set of license breaches.

F. Copilot Reproduces the Code of the Named Plaintiffs Without Attribution

109. Because Copilot was trained on all available public GitHub repositories, if Licensed Materials have been posted to a GitHub public repository, Plaintiffs and the Class can be reasonably certain it was ingested by Copilot and is sometimes returned to users as Output.

110. Described below are some specific examples of Copilot's unlawful behavior using Licensed Materials owned by the named Plaintiffs. These examples were emitted by Copilot after prompting Copilot.

111. In the examples below, original code is shaded gray, prompts to Copilot are shaded orange, and outputs from Copilot are shaded light blue.

1. Example: Copilot Outputs the Code of Doe 2 Essentially Verbatim

112. The first example demonstrates Copilot suggesting an essentially verbatim copy of code written by Doe 2.

113. [redacted] subject to the GNU General Public License v3.0. [redacted] The relevant code from the original source file is shown below:

[redacted]

114. When Copilot is prompted the first few lines of Doe 2's code:

[redacted]

Copilot suggests the following:

[redacted]

[redacted]

115. This suggestion from Copilot is identical to Doe 2's code, except that [redacted] These differences in the code are cosmetic and the code is functionally equivalent; otherwise, this is a verbatim copy. Doe 2's particular arrangement and sequencing seen in his code is distinctive expression found only in one location on GitHub: [redacted]

116. Because the Copilot suggestion is a nearly verbatim reproduction of Doe 2's unique code, it follows that Copilot copied Doe 2's code. Copilot therefore needed to adhere to the requirements of Doe 2's license (GNU General Public License v3.0) for that code, including providing attribution. It does not. Copilot also did not reproduce Doe 2's license.

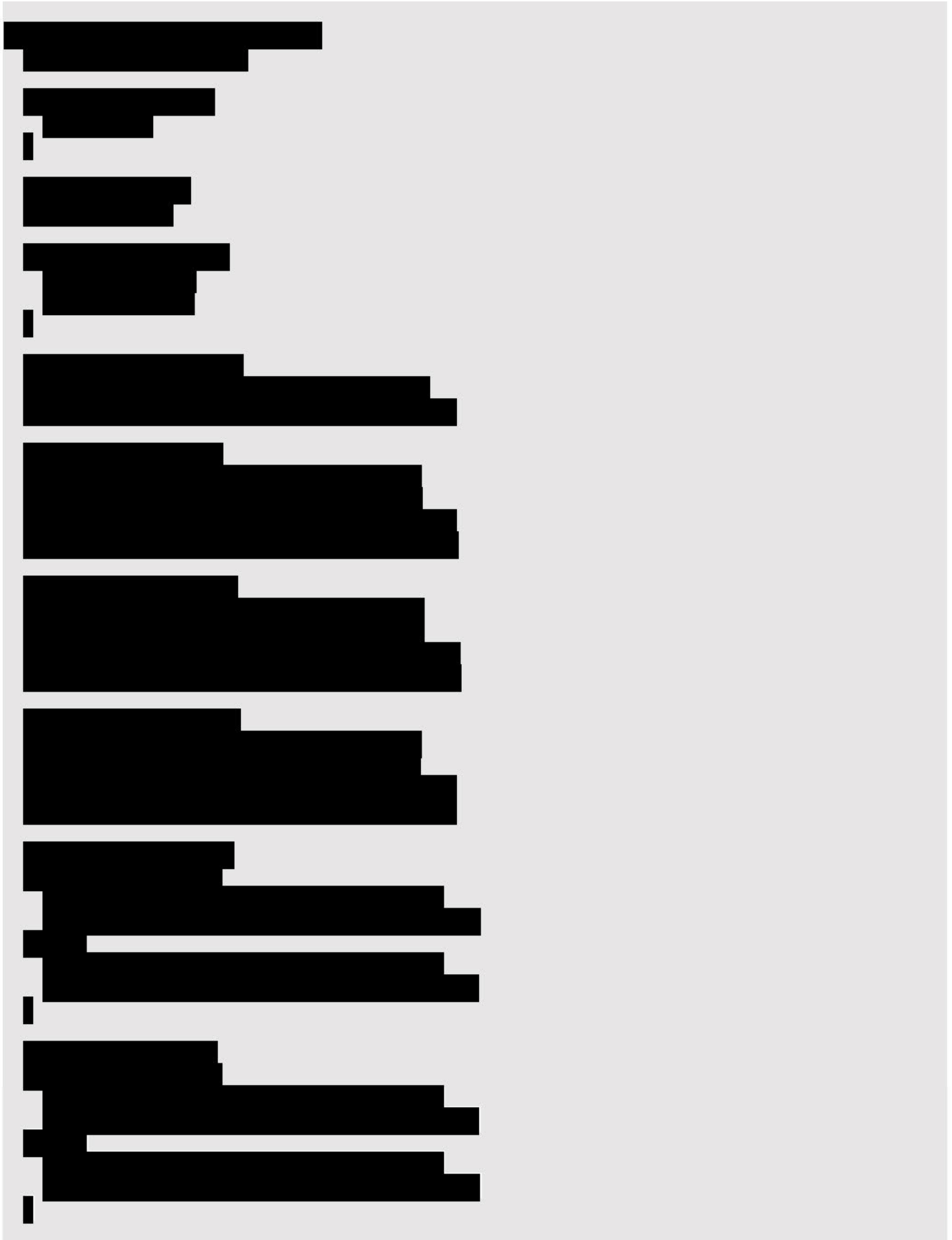
2. Example: Copilot Outputs the Code of Doe 1 in Modified Format

117. The second example demonstrates Copilot suggesting a modified copy of code written by Doe 1. To protect Doe 1's identity, the paragraphs describing the code will be redacted.

118. [redacted] subject to the MIT License. [redacted]

[redacted]

[CONFIDENTIAL]



[CONFIDENTIAL]

[redacted]

119. When Copilot is prompted with [redacted]

[redacted]

The first suggestion from Copilot is a modification of Doe 1's code:

[redacted]

[redacted]

120. [redacted] do not appear in any other source file on GitHub. The only way Copilot knows how to make this suggestion is because it ingested Doe 1's source file as training data. Though the Copilot suggestion is not an exact match for Doe 1's code, it is necessarily a modification based on a copy of Doe 1's code.

121. Furthermore, many distinctive expressive features of Doe 1's code have been preserved in Copilot's suggestion. For instance, Doe 1's comments in the code (in green) are reproduced almost verbatim. [redacted]

[redacted]

means the same thing as this Copilot-suggested code:

[redacted]

122. As is apparent from a cursory glance of this example, the variations between Copilot's emitted output and Doe 1's source code are cosmetic and the code is functionally equivalent; it follows that Copilot's output is a copy of Doe 1's code.

123. That said, Copilot also introduces mistakes into the code. For instance, [redacted]

124. Still, because Copilot is reproducing Doe 1's algorithm in modified format, and the obligations in Doe 1's license (the MIT License) carry with the code even if the underlying code is modified, the Copilot suggestion needs to follow the requirements of Doe 1's license for that code, including providing attribution. It does not. Copilot also did not reproduce Doe 1's license.

3. Example: Copilot Outputs the Code of Doe 5 In Modified Format

125. The third example demonstrates Copilot suggesting multiple modified copies of code written by Doe 5 in response to a sequence of prompts, which is a common way of using Copilot. To protect Doe 5's identity, the paragraphs describing the code will be redacted.

126. [redacted] subject to the MIT License. [redacted] The relevant code from the original source file is shown below:

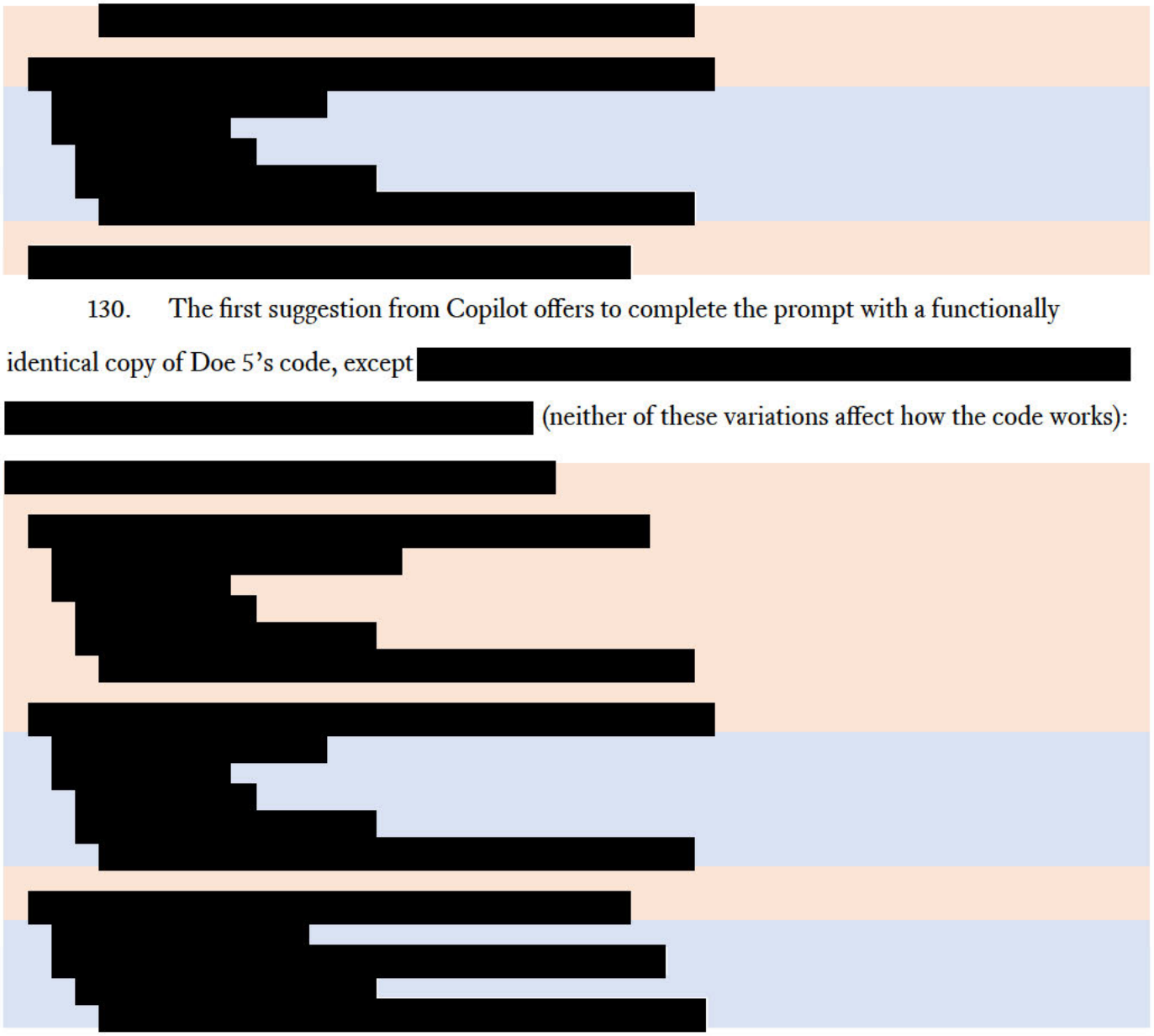
[redacted]

127. When Copilot is prompted the first section of Doe 5's code, comprising the first complete test and the name of the second:

128. The first suggestion from Copilot offers to complete the prompt with a verbatim copy of Doe 5's original code, except that [redacted] (a variation that does not affect how the code works):

129. Next, if the name of the third test is appended, the next prompt to Copilot looks like this:

[CONFIDENTIAL]



130. The first suggestion from Copilot offers to complete the prompt with a functionally identical copy of Doe 5's code, except [redacted] (neither of these variations affect how the code works):

131. As is apparent from the high degree of similarity and minor cosmetic deviations between Copilot's emitted output and Doe 5's source code, Copilot ingested, copied and reproduced Doe 5's source code as output.

132. Because Copilot is (repeatedly) reproducing Doe 5's original code in modified format, and the obligations in Doe 5's license (the MIT License) carries with the code even when it is modified, the Copilot suggestions need to follow the requirements of Doe 5's license for that code, including providing attribution. They do not. Copilot also did not reproduce Doe 5's license.

4. Example: Copilot Outputs Code of Doe 5 Essentially Verbatim

133. The fourth example also demonstrates Copilot suggesting multiple modified copies of code written by Doe 5 in response to a sequence of prompts, which is a common way of using Copilot. To protect Doe 5's identity, the paragraphs describing the code will be redacted.

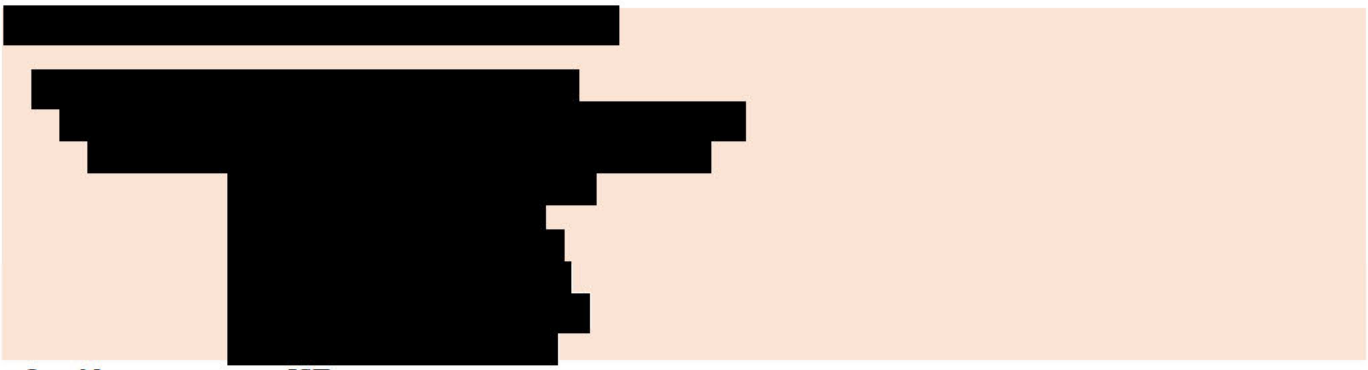
134. [redacted] subject to the MIT License. [redacted] The first three tests from the original source file are shown below:

[redacted]

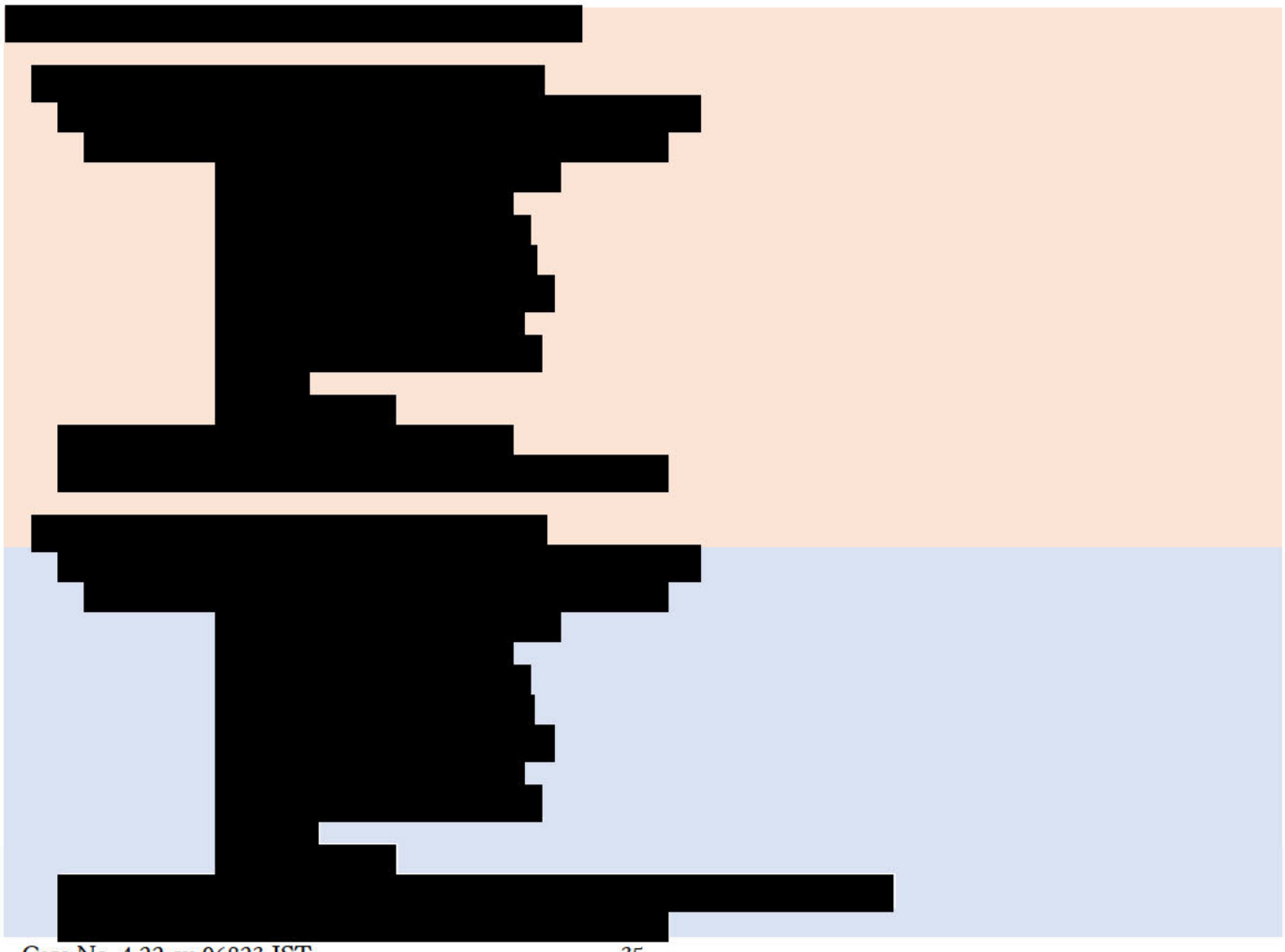
[CONFIDENTIAL]

135. When Copilot is prompted with the first section of Doe 5's code, comprising the first complete test and the name of the second:

The first suggestion from Copilot offers to complete the second test with a verbatim copy of Doe 5's original code:



136. When Copilot's suggestion is accepted and the name of Doe 5's third test is appended, the next prompt to Copilot looks like this:



[redacted]

137. Once again, the first suggestion from Copilot offers to complete the third test with a verbatim copy of Doe 5's code (except for small cosmetic variations in line breaks):

[redacted]

[redacted]

138. Because Copilot is (repeatedly) reproducing Doe 5's code essentially verbatim, the Copilot suggestions need to follow the requirements of Doe 5's license (the MIT License) for that code, including providing attribution. They do not. Copilot also did not reproduce Doe 5's license.

139. These are only a few examples of Plaintiffs' code being reproduced by Copilot. It follows that many if not all prompts entered into Copilot will readily cause it to emit verbatim, near-verbatim or modified copies of Licensed Material that violate the licenses under which the source code is published. Multiplied across the many users of Copilot and the many times Copilot is prompted, each day these violations must be accruing with astonishing frequency. It is therefore likely if not certain that verbatim, near-verbatim or modified copies of each Plaintiffs' code have already been emitted by Copilot.

140. Additionally, even though Plaintiffs have been able to generate these examples, Plaintiffs remain at a great evidentiary disadvantage relative to Defendants, because Defendants control all the information about the training dataset. In particular, only Defendants know *when* the Licensed Materials of Plaintiffs and the Class were scraped. As is typical in open source, many of the Licensed Materials are regularly updated. As such, it is difficult to determine which iterations of code may have been trained on and would be subject to emission by Copilot.

G. Codex and Copilot Were Designed to Withhold Attribution, Copyright Notices, and License Terms from Their Users

141. Codex and Copilot have no way to determine whether license text or other Copyright Management Information ("CMI")[21] is part of the code it appears immediately before or after. Unless instructed otherwise, it will assume that CMI that usually appears just before a given block of code is an important part of that code or otherwise necessary for it to function.

142. It is a common practice to provide the applicable license text at the top of every source file in the codebase. The purpose of this practice is to avoid the code from being divorced from the license. This may occur via "vendoring," a method of creating a derivative work by including source files from a

[21] CMI is defined in detail below in Paragraph 211.

copyrighted project directly into another project without following the terms of the license or providing attribution or a copyright notice. Copilot circumvents this protective measure to mask the degree of vendoring it engages in.

143. Early iterations of Copilot reproduced license text. For example, in a blog post, GitHub noted "In one instance, GitHub Copilot suggested starting an empty file with something it had even seen more than a whopping 700,000 different times during training–that was the GNU General Public License."[22] Copilot no longer suggests licenses in this way because it has been altered not to. As GitHub explains: "GitHub Copilot *has* changed to require a minimum file content. So some of the suggestions flagged here would not have been shown by the current version."

144. In July 2021, near Copilot's launch, it would sometimes produce license text, attribution, and copyright notices. This CMI was not always accurate. Copilot no longer reproduces these types of CMI, incorrect or otherwise, on a regular basis. It has been altered not to.

145. In July 2022, in response to public criticism of Copilot's mishandling of Licensed Materials, GitHub introduced a user-settable Copilot filter called "Suggestions matching public code." If set to "block," this filter claims to prevent Copilot from suggesting verbatim excerpts of "about 150 characters" that come from Licensed Materials. But even assuming the filter works as advertised, because it only checks for verbatim excerpts, it does nothing to impede the Outputs from Copilot that are modifications of Licensed Materials. Thus, as a device for respecting the rights of Plaintiffs and the Class, it is essentially worthless.

146. GitHub Copilot now "includes an option to either allow or block code completion suggestions that match publicly available code. If you choose to block suggestions matching public code, GitHub Copilot checks code completion suggestions with their surrounding code of about 150 characters against public code on GitHub. If there is a match, or a near match, the suggestion is not shown to you."[23] There is no reason provided for GitHub's choice of 150 matching characters of code. Nor is there a reason for GitHub to include an option to block suggestions that "match" code unless Copilot is capable of (and

[22] https://github.blog/2021-06-30-github-copilot-research-recitation/.

[23] https://docs.github.com/en/copilot/configuring-github-copilot/configuring-github-copilot-settings-on-githubcom#enabling-or-disabling-duplication-detection

does) emit verbatim copies of code. Nonetheless, GitHub provides the choice to Copilot's paying users to use code Copilot outputs that is identical to code on its public repositories, subject to open source licenses.

147. GitHub now admits[24] that "You can opt to allow GitHub Copilot to suggest code completions that match publicly available code on GitHub.com." GitHub states that "If you have allowed suggestions that match public code**,** GitHub Copilot can provide you with details about the matching code when you accept such suggestions. This feature is called code referencing." Again, the only reason GitHub would inform users that they can opt in to allowing Copilot to suggest code completions that "match" publicly available code (i.e., code that Codex and Copilot were trained on) is that Copilot is capable of, and does, emit code suggestions that are verbatim copies of code.

148. GitHub states it now can provide its users, at their option, a link in to the relevant identical open-source license under which such identical code was published on GitHub. Having acknowledged there is the likelihood, if not certainty that Github will produce exact copies, and providing a tool to prevent this, GitHub makes it entirely optional to users, and provides no such optionality to licensors. Thus users who want to receive identical code from GitHub or do not want to exclude it, may do so. In so doing, GitHub facilitates and encourages users to receive identical code.

149. As a result, Plaintiffs are informed and believe, and on that basis allege that it is likely that their licensed code is omitted by Github in violation of the open source licenses. Further Plaintiffs are informed and believe, and on that basis allege that with respect to numerous members of the class, it is certain that some identical code has been omitted by GitbHub. Plaintiffs are informed and believe, and on that basis allege that there is a substantial risk, if not certainty, that identical code will be emitted in the future. Further, given the fact that GitHub has implemented a tool to prevent this, and have made it optional, not mandatory, to users, GitHub knows, or has reason to know that identical code will be omitted in the future.

150. Further, Github but makes clear it is entirely up the user to add any license or attribution to the code GitHub generated for them. "The linked web page includes details of any license identified for

[24] https://docs.github.com/en/copilot/using-github-copilot/finding-public-code-that-matches-github-copilot-suggestions

the repository where the matching code was found. Having reviewed the references, *you can decide how to proceed*." This link temporarily shows up in the users' "GitHub Copilot Log view" but "[t]he GitHub Copilot log is flushed when you close the editor."

151. In addition to the fact that the tool which identifies and screens identical code is an option, at *the discretion of the user*, GitHub also acknowledges that the optional tool to prevent or exclude identical code is not 100 per cent effective. Indeed, GitHub confirms it is limited in its scope and might produce Plaintiffs or class members' matching code, but otherwise will not detect it under certain circumstances. GitHub states "code from public repositories deleted before the index was created, may not be included in the search. For the same reason, the search may return matches to code that has been deleted or moved since the index was created."

152. GitHub also admits the ability and therefore the efficacy of the code referencing tool is limited, incomplete and predictably ineffective, admits that its code referencing feature is limited to finding identical public code made from indexes of GitHub public repositories *after* training Copilot, such that "code from public repositories deleted before the index was created, may not be included in the search." Github thereby admits that Copilot trained on code posted on GitHub can be output in identical form, without the required attribution of other compliance with license terms. This may occur whch nonetheless still may not be attributable in any way, even with the best intentions of Copilot's commercial users and even given GitHub's "code reference" feature . Simply put, GitHub admits Copilot can and does output identical matching code of Plaintiffs and class members that its own "code referencing" feature cannot detect, even on the infrequent occasion when a user exercises the option to implement the feature.

153. GitHub further represents to Copilot's paying users that "Typically, matches to public code occur in less than one percent of Copilot suggestions, so you should not expect to see code references for many of the suggestions you accept." On information and belief, Plaintiffs are informed abd believe, and allege that this number is much higher.

154. GitHub elsewhere represents: "If you choose to allow suggestions matching public code, and you accept a suggestion for which one or more matches were found, you can click through from an

entry in the GitHub Copilot log to view a list of references on GitHub." [25] But there is no requirement to attach the license to matching code Copilot outputs for its paying users. In other words, there are many situations in which public code is emitted without compliance with license terms.

155. In GitHub's hands, the propensity for small cosmetic variations in Copilot's Output is a feature, not a bug. GitHub does so knowing that these small cosmetic variations allow CoPilot to conceal the copying of Licensed Materials and to separate the Licensed Material from the licenses. In so doing, GitHub knowingly conceals the copying of Licensed Materials with the intent and purpose of making it difficult for Plaintiffs and members of the class to identify breaches of the licenses and to enforce their rights. These small cosmetic variations mean that GitHub can deliver to Copilot customers unlimited modified copies of Licensed Materials without ever triggering Copilot's verbatim-code filter. AI models like Copilot often have a setting called *temperature* that specifically controls the propensity for variation in their output. On information and belief, GitHub has optimized the temperature setting of Copilot to produce small cosmetic variations of the Licensed Materials as often as possible, so that GitHub can deliver code to Copilot users that *works* the same way as verbatim code, while claiming that Copilot only produces verbatim code 1% of the time. This technique of active concealment has been effective on many occasions. Copilot is an ingenious method of software piracy and concealment.

156. In December 2022, GitHub launched Copilot for Business. The initial terms of service included one notable extra provision compared to ordinary Copilot: a "Defense of Third Party Claims" that read:

> GitHub will defend you against any claim by an unaffiliated third-party that your use of GitHub Copilot misappropriated a trade secret or directly infringes a patent, copyright, trademark, or other intellectual property right of a third party, up to the greater of $500,000.00 USD or the total amount paid to GitHub for the use of GitHub Copilot during the 12 months preceding the claim. GitHub's defense obligations do not apply if (i) the claim is based on Code that differs from a Suggestion provided by GitHub Copilot, (ii) you fail to follow reasonable software development review practices designed to prevent the intentional or inadvertent use of Code in a way that may violate the intellectual property or other rights of a third party, or (iii) you have not enabled all filtering features available in GitHub Copilot.

[25] https://docs.github.com/en/copilot/configuring-github-copilot/configuring-github-copilot-settings-on-githubcom#enabling-or-disabling-duplication-detection

157. If Copilot had been designed to reproduce the attribution, license terms, and copyright notices of the Licensed Materials, this kind of contractual reassurance wouldn't be necessary. With this provision (since removed), GitHub acknowledged that Copilot disrupts—possibly with legal consequences—the relationship between authors and users of open-source software.

B. Open-Source Licenses Began to Appear in the Early 1990s

158. In 1991, software engineer Linus Torvalds began a project to create a UNIX-like operating system that would run on common PC hardware. This project became known as Linux.

159. To encourage adoption of his system, and persuade other programmers to contribute, he released Linux under what was then an unusual software license called the GNU General Public License, or GPL.

160. The GPL is a software license. But whereas most software licenses required payment, software under the GPL is provided for free. Whereas most software licenses did not include source code, GPL software always included source code. And whereas most software licenses prohibited derivative works, the GPL not only allowed it, but encouraged it.

161. In certain ways, however, the GPL still operated like a traditional software license. For example, consistent with copyright law, it depended on an assertion of copyright by the software author. Even though GPL software was available at no charge, the GPL contained conditions on its users as licensees.

162. One license requirement was that a program derived from GPL software had to redistribute certain information about that software:

> You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program.[26]

Failure to adhere to these conditions constituted a violation of the license, triggering the possibility of

[26] https://www.gnu.org/licenses/old-licenses/gpl-1.0.en.html.

legal action. Provisions of the GPL are enforceable, and many GPL licensors have sought to enforce GPL licenses though court proceedings and other litigation.

163. The early years of Linux paralleled the early years of the World Wide Web. The fact that Linux was free and ran on common computer hardware made it a popular choice for web servers. Because of its contrarian GPL licensing, Linux became hugely popular. A large ecosystem of other programs and tools grew around it. This contributed to the explosive growth of the web and other network services across the rest of the 1990s.

164. In turn, the growth of the World Wide Web made it easier for developers in different places to collaborate on software. The GPL, and licenses like it, were a natural fit for this kind of collaborative work.

165. Around 1998, a new name was coined as an umbrella term for these principles of software licensing and development: *open source*.

H. Microsoft Has a History of Flouting Open-Source License Requirements

166. During the 1980s and 1990s, Microsoft was primarily a software company, focusing largely on operating systems and related applications. These included its DOS operating system and later, its Windows operating system. Windows generated billions of dollars in revenue from its sale and licensing as proprietary software for desktop computers and servers. Microsoft derived substantial income from sale of licensed products and devotes substantial resources to protecting and enforcing such licenses.

167. Windows is a graphical operating system. It allows users to view and store files, run software and games, play videos, and provides a way to connect to the internet.

168. Linux represented a competitive threat to Windows. It ran on the same hardware. It performed many of the same functions. It was free. Many programmers at the time considered Linux to be functionally superior to Windows.

169. Microsoft has engaged in a problematic practice known as "vaporware," where products are announced but are in fact late, never manufactured, or canceled. Typically the company promising vaporware never has any intention of providing it. The term vaporware was coined by Microsoft in 1982 in reference to the development of its Xenix operating system.

170. Microsoft described its anti-Linux strategy as "FUD," standing for fear, uncertainty, and doubt. Microsoft focused extra attention to Linux's open-source aspects.

171. In 1998, a source at Microsoft leaked what became known as the "Halloween Documents", revealing Microsoft's thinking on how to counter the competitive threat from Linux. Among other things, the documents emphasized the importance of countering the "long term developer mindshare threat", and concluded that to defeat open source, "[Microsoft] must target a process rather than a company."[27]

172. In 2001, Microsoft CEO Steve Ballmer said "The way the [GPL] is written, if you use any open-source software, you must make the rest of your software open source. . . . Linux is a cancer that attaches itself in an intellectual property sense to everything it touches."[28] Ballmer's summary of GPL licensing was not accurate. In 2001, Linux was being used by corporations of every size. The growth of open source up to that point, and since, has been made possible by the open-source community's respect for and compliance with applicable licenses.

173. In 2001, Microsoft was the defendant in a major software-related antitrust case, *United States v. Microsoft Corporation*.[29] In this case, the U.S. Department of Justice accused Microsoft of maintaining a software monopoly by illegally imposing technical restrictions on manufacturers of personal computers, including "tying" violations related to the Internet Explorer web browser. Judge Thomas Penfield Jackson, who presided over the antitrust trial, opined that Microsoft is "a company with an institutional disdain for both the truth and for rules of law that lesser entities must respect. It is also a company whose 'senior management' is not averse to offering specious testimony to support spurious defenses to claims of its wrongdoing."[30]

174. In 2007, Microsoft admitted that it tried to influence the vote of an ISO open-standards committee by offering money to certain business partners in Sweden to vote for Microsoft's preferred outcome.[31]

---

[27] http://www.catb.org/esr/halloween/halloween1.html.

[28] https://lwn.net/2001/0607/a/esr-big-lie.php3.

[29] No. Civ.A. 00–1457 TPJ.

[30] *Jackson v. Microsoft Corp.*, 135 F. Supp. 2d 38 (D.D.C. 2001).

[31] https://learn.microsoft.com/en-us/archive/blogs/jasonmatusow/open-xml-the-vote-in-sweden/.

175. After observing the rapid growth of Amazon's original cloud computing products, Microsoft has expanded its business into cloud computing, which it has branded Microsoft Azure or simply Azure. Microsoft announced Azure to developers in 2008. It was formally released in 2010. Azure uses large-scale virtualization at Microsoft data centers and offers many hundreds of services, including infrastructure as a service ("IaaS"), platform as a service ("PaaS"), compute services, Azure Active Directory, mobile services, storage services, communication services, data management, messaging, developer services, Azure AI, blockchain, and others.

I. GitHub Was Designed to Cater to Open-Source Projects

176. By 2002, Linux had become immensely popular. But the project itself had become unwieldy and had outgrown its reliance on informal systems of managing software source code (also known as *source-control systems*). The Linux community needed something better.

177. Linus Torvalds set about writing a new source-control system. He named his new system Git. He released it under the GPL. It quickly became the source-control system of choice for open-source programmers.

178. A single software project stored in Git is called a *source repository*, commonly shortened to *repository* or just *repo*. A Git source repository would typically be stored on a networked server accessible to a group of programmers.

179. This became less convenient, however, when programmers were distributed among multiple locations, rather than being in a single location. A Git repository could be stored on an internet-accessible server. But setting up that server hardware and being responsible for it was inconvenient and expensive.

180. In 2008, a group of open-source developers in San Francisco, California founded GitHub. GitHub managed internet servers that hosted Git source repositories. With an account at GitHub, an open-source developer could easily set up a Git project accessible to collaborators anywhere in the world. From early on, GitHub's core market was open-source developers, whom it attracted by making many of its hosting services free.

181. Most open-source programmers used GitHub to create "public" repositories, meaning that anyone could view them & access them. GitHub also allowed programmers and organizations to

create "private" repositories, which were not accessible from the public GitHub website, and required password access.

182. Open-source licensing was integral to GitHub. GitHub encouraged open-source developers to understand and use open-source licenses for their work. Many—though not all—public repositories on GitHub carry an open-source license. By convention, this license is stored at the top level of each repository in a file called LICENSE. GitHub's interface also includes a button on the front pages of most repositories users can click to see details of the applicable license. A human user could easily find the license in either of these locations—as could an AI anywhere near as powerful as Codex or Copilot.

183. Though the GPL is one of the early open-source licenses and remains common, it is not the only open-source license. Examples of other common open-source licenses include the MIT License, the Apache License, and the Berkeley Software Distribution License (all of which are included in the Suggested Licenses).

184. Though these licenses differ in their wording and their details, most of them share a requirement that a copy of the license be included with any copy, derivative, or redistribution of the software, and that the author's name and copyright notice remains intact. This is not a controversial requirement of open-source licenses—indeed, it has been an integral part of the GPL for over 30 years.

185. There are also many public repositories on GitHub that have no license. Though GitHub has encouraged awareness of licenses among its users, it has never imposed a default license on public repositories. A public repository without a license is subject to ordinary rules of U.S. copyright.

186. Open-source developers flocked to GitHub. By 2018, GitHub had become the largest and most successful Git hosting service, hosting millions of users and projects.

187. In October 2018, Microsoft acquired GitHub for $7.5 billion. It was important to Microsoft that programmers use GitHub. Microsoft had developed a well-deserved poor reputation because of its documented vaporware, FUD, and other business practices, including those targeted at open-source programs and programming, and open-source licensing specifically. Microsoft made false and misleading statements and omissions to assuage such concerns, including its primary mantra intended to win over the open-source community: "Microsoft Loves Open Source."

J. OpenAI Is Intertwined with Microsoft and GitHub

188. OpenAI, Inc. is a nonprofit corporation founded in December 2015 by a group that included Greg Brockman, Ilya Sutskever, and other AI researchers; Elon Musk, CEO of Tesla; and Sam Altman, president of Y Combinator, a tech-startup incubator with hundreds of companies in its portfolio. Musk and Altman served as co-chairs of OpenAI, Inc. One of OpenAI, Inc.'s current board members is Reid Hoffman, founder of LinkedIn, which is now a Microsoft subsidiary. Mr. Hoffman is also a member of the Microsoft Board of Directors.

189. Less than a year later, in November 2016, OpenAI first partnered with Microsoft. It described the partnership as follows: "We're working with Microsoft to start running most of our large-scale experiments on Azure. This will make Azure the primary cloud platform that OpenAI is using for deep learning and AI, and will let us conduct more research and share the results with the world."

190. Initially, OpenAI, Inc. held itself out as a "non-profit artificial intelligence research company" that sought to shape AI "in the way that is most likely to benefit humanity as a whole."

191. OpenAI, Inc. reportedly secured $1 billion in initial funding, from sources that were largely not disclosed, but included at least most of its founders.

192. OpenAI, Inc. obtained its initial source of training data from its founders' companies. According to reporting at the time, Musk and Altman planned to "pool[] online data from their respective companies" to serve as training data for OpenAI, Inc. projects. Musk planned to contribute data from Tesla; Altman planned to have Y Combinator companies "share their data with OpenAI."[32]

193. In February 2019, Altman created OpenAI, LP, a for-profit subsidiary of the nonprofit entity OpenAI, Inc. The new OpenAI, LP entity would serve as a vessel for accepting traditional outside investment in exchange for equity and distributing profits.

194. In July 2019, OpenAI, L.P. accepted a $1 billion investment from Microsoft. In addition to cash, Microsoft would become the exclusive licensor of certain OpenAI, LP products (including Codex, described below in Paragraph 197). Also, as part of this alliance, OpenAI, LP would use Microsoft's cloud-computing platform, Azure, exclusively to develop and host its products. Some portion of Microsoft's

[32] https://www.wired.com/2015/12/elon-musks-billion-dollar-ai-plan-is-about-far-more-than-saving-the-world/.

investment was paid in credits for use of Azure rather than cash. Finally, Microsoft and OpenAI agreed to "jointly build new Azure AI supercomputing technologies."

195. Azure is a major growth area for Microsoft. In its most recent earnings report on October 25, 2022, "Azure and other cloud services" grew by 35% from the previous quarter, more than any other product.[33] Azure has grown rapidly since Microsoft began its partnership with OpenAI in 2016. Its revenue grew by 50% or more every quarter from 2016 through the first three quarters of 2020.

196. In May 2020, Microsoft and OpenAI announced they had jointly built a supercomputer in Azure that would be used exclusively by OpenAI to train its AI models. Microsoft's influence over and frequent collaboration with OpenAI has led some to describe Microsoft as "the unofficial owner of OpenAI."[34]

197. One of OpenAI's projects is GPT-3, a so-called "large language model" designed to emit naturalistic text. When researchers noticed that GPT-3 could also generate software code, they started studying whether they could make a new AI model specifically trained for this purpose. This project became known as Codex.

198. Sometime after July 2019, OpenAI and Microsoft began collaborating on a code-completion product for GitHub that would use Codex as its underlying model. This product became known as Copilot.

199. On September 28, 2022, OpenAI released an image-generation AI called DALL-E-2. Much like Copilot, DALL-E-2 removes any attribution and/or copyright notice from the images it uses to create derivative works. Like with Codex, here, OpenAI ignores the rights of the owners of copyrights to images it has ingested.

200. In another joint project, Microsoft and OpenAI recently launched a preview of a product called "Azure OpenAI Service."[35] This service will "Leverage large-scale, generative AI models with deep understandings of language and code to enable new reasoning and comprehension capabilities for building

[33] https://www.microsoft.com/en-us/Investor/earnings/FY-2023-Q1/press-release-webcast/.

[34] https://venturebeat.com/ai/what-to-expect-from-openais-codex-api/.

[35] https://azure.microsoft.com/en-us/products/cognitive-services/openai-service/.

cutting-edge applications. Apply these coding and language models to a variety of use cases, such as writing assistance, code generation, and reasoning over data. Detect and mitigate harmful use with built-in responsible AI and access enterprise-grade Azure security."

K. Conclusion of Factual Allegations

201. Future AI products may represent a bold and innovative step forward. GitHub Copilot and OpenAI Codex, however, do not. Defendants should not have released these products until they could ensure that they did not constantly violate Plaintiffs' and the Class's intellectual-property rights, licenses, and other rights.

202. Defendants have made no attempt to comply with the open-source licenses that are attached to much of their training data. Instead, they have pretended those licenses do not exist, and trained Codex and Copilot to do the same. By simultaneously violating the open-source licenses of tens-of-thousands—possibly millions—of software developers, Defendants have accomplished software piracy on an unprecedented scale. As Microsoft's Co-Founder Bill Gates once said regarding software piracy: "the thing you do is theft."[36]

203. There is no inherent limitation or constraint of AI systems that made any of this necessary. Defendants chose to build AI systems designed to enhance their own profit at the expense of a global open-source community that they had once sought to foster and protect. GitHub and OpenAI are profiting at the expense of Plaintiffs' and the Class's rights.

VIII. CLAIMS FOR RELIEF

COUNT 1
VIOLATION OF THE DIGITAL MILLENNIUM COPYRIGHT ACT
17 U.S.C. §§ 1201–1205
(For Injunctive Relief)
(Against All Defendants)

204. Plaintiffs and the Class hereby repeat and incorporate by reference each preceding and succeeding paragraph as though fully set forth herein.

205. As described herein, Defendants have intentionally removed or altered CMI from Plaintiffs' code in violation of Section 1202(b)(1) of the DMCA.

[36] https://www.digibarn.com/collections/newsletters/homebrew/V2_01/gatesletter.html

206. As described herein, there is a substantial risk that Defendants will distribute copies of Plaintiffs' code knowing that CMI has been removed or altered while knowing or having reasonable grounds to know that it will induce, enable, facilitate, or conceal infringement in violation of Section 1202(b)(3) of the DMCA.

207. GitHub has admitted that about 1% of the time, a suggestion may contain code snippets longer than ~150 characters that matches code from the training data. In other words, GitHub itself has admitted that Copilot can emit identical copies of code Copilot was trained on.

208. GitHub has implemented features which allow users to block output blocks suggestions matching public code. GitHub would not implement such a feature unless it knows that Copilot is capable of, and does, emit output that matches code found on public repositories.

209. Further, given GitHub's increasing commitment to growing an AI model and the scale of Copilot's code, given that academic research suggests that a model increases the likelihood of emitting training data it has "memorized," the chance that Copilot will emit code that matches code found in the training data is only increasing as the model scales. On information and belief, if Copilot has not done so already, Copilot will emit identical copies of Class members' code.

210. Plaintiffs and members of the Class own the copyrights to Licensed Materials used to train Codex and Copilot. Copilot was trained on millions—possibly billions—of lines of code publicly available on GitHub. Copilot runs on Microsoft's Azure cloud platform exclusively and Microsoft had input in the creation of Copilot. Microsoft is aware that Copilot ignores License Terms and that it was trained almost exclusively on Licensed Materials.

211. Plaintiffs and members of the Class included the following Copyright Management Information (as defined in Section 1202(c) of the DMCA) ("CMI") in the Licensed Materials:

a. copyright notices;

b. the title and other information identifying the Licensed Materials;

c. the name of, and other identifying information about, the authors of the Licensed Materials;

d. the name of, and other identifying information about, the copyright owners of the Licensed Materials;

e. terms and conditions for use of the Licensed Materials, specifically the Suggested Licenses; and

f. identifying numbers or symbols referring to CMI or links to CMI.

212. Defendants did not contact Plaintiffs and the Class to obtain authority to remove or alter CMI from the Licensed Materials within the meaning of the DMCA.

213. Defendants knew that they did not contact Plaintiffs and the Class to obtain authority to remove or alter CMI from the Licensed Materials within the meaning of the DMCA.

214. As part of the scheme, Defendants did not attempt to contact Plaintiffs or Class members to obtain authority to remove or alter CMI from the Licensed Materials within the meaning of the DMCA. In fact, he removal of CMI made it difficult or impossible to contact Plaintiffs and the Class to obtain authority to remove or alter CMI from the Licensed Materials within the meaning of the DMCA. Rather, Defendants removed or altered CMI from open-source code that is owned by Class members after the code was uploaded to a GitHub repository by incorporating it into Copilot with its CMI removed.

215. Without the authority of Plaintiffs and the Class, Defendants intentionally removed or altered CMI from the Licensed Materials after they were uploaded to one or more GitHub repositories.

216. Defendants had access to but were not licensed by Plaintiffs nor the Class to train any machine learning, AI, or other pseudo-intelligent computer program, algorithm, or other functional prediction engine using the Licensed Materials.

217. Defendants had access to but were not licensed by Plaintiffs nor the Class to incorporate the Licensed Materials into Copilot.

218. Defendants had access to but were not licensed by Plaintiffs nor the Class to distribute the Licensed Materials as they do through Copilot.

219. Without the authority of Plaintiffs and the Class, Defendants distributed CMI knowing that the CMI had been removed or altered without authority of the copyright owner or the law with respect to the Licensed Materials.

220. Defendants distributed copies of the Licensed Materials knowing and intending that CMI had been removed or altered without authority of the copyright owner or the law, with respect to the Licensed Materials.

221. Defendants removed or altered CMI from the Licensed Materials knowing and intending that it would induce, enable, facilitate, or conceal infringement of copyright.

222. Without the CMI associated with the Licensed Materials, Copilot users are induced or enabled to copy the Licensed Materials. Because CMI has been removed, Copilot users do not know whether Output is owned by someone else and subject to restrictions on use. Without the CMI, copyright infringement is facilitated or concealed, because Plaintiffs and the Class are prevented from knowing or learning that the Output is based upon one or more of the Licensed Materials. Use of the Licensed Materials is not infringement when the terms of the applicable Suggested License are followed. Had the CMI not been removed, Copilot users would be aware of the Licenses and their obligations under them. The terms of the applicable Suggested License would have allowed those users to use the Licensed Materials without infringement. By withholding and concealing license information and other CMI, Defendants prevented Copilot users from making non-infringing use of the Licensed Materials. This contradicts the express wishes of Plaintiffs and the Class, which are set forth explicitly in the Suggested Licenses under which the Licensed Materials are offered.

223. Defendants removed or altered CMI from Licensed Materials owned by Plaintiffs and the Class while possessing reasonable grounds to know that it would induce, enable, facilitate, and/or conceal infringement of copyright in violation of Sections 1202(b)(1) and 1202(b)(3) of the DMCA.

224. By omitting, altering and/or concealing CMI from Copilot's Output, Defendants have reasonable grounds to know that innocent infringers are induced or enabled to copy the Licensed Materials, because CMI has been removed. Without the CMI, Defendants have reasonable grounds to know copyright infringement is facilitated or concealed, because Plaintiffs and the Class have the difficult or impossible task of proving the Licensed Materials belong to them.

225. The profits attributable to Defendants' violation of the DMCA include the revenue from: Copilot subscription fees, sales of or subscriptions to Defendants' Copilot-related products and/or services that are used to run Copilot, hosting Copilot on Azure, and any other of Defendants' products that contain copies of the Licensed Materials without all the original CMI. The Licensed Materials add nearly all value to the Copilot product because the purpose of Copilot is to provide code and the source of that code is the Licensed Materials. Without the Licensed Materials, Copilot would not be functional.

226. On information and belief, Defendants could have trained Copilot to include attribution, copyright notices, and license terms when it provides Output covered by a License.

227. Defendants did not request or obtain permission from Plaintiffs and the Class to use the Licensed Materials for Defendants' Copilot product.

228. Defendants use of the Licensed Materials does not follow the requirements of the Suggested Licenses associated with the Licensed Materials. In particular, Copilot fails to provide attribution for the creator nor the owner of the Work. Copilot fails to include the required copyright notice included in the License. Copilot fails to include the applicable Suggested License's text.

229. Defendants are sophisticated with respect to intellectual property matters related to open-source code. Microsoft in particular has extensive experience granting licenses, obtaining licenses, and enforcing license terms. Its most recent Annual Report states:

> **We protect our intellectual property investments in a variety of ways. We work actively in the U.S. and internationally to ensure the enforcement of copyright, trademark, trade secret, and other protections that apply to our software and hardware products, services, business plans, and branding.** We are a leader among technology companies in pursuing patents and currently have a portfolio of over 69,000 U.S. and international patents issued and over 19,000 pending worldwide. While we employ much of our internally-developed intellectual property exclusively in our products and services, we also engage in outbound licensing of specific patented technologies that are incorporated into licensees' products. From time to time, we enter into broader cross-license agreements with other technology companies covering entire groups of patents. We may also purchase or license technology that we incorporate into our products and services. At times, we make select intellectual property broadly available at no or low cost to achieve a strategic objective, such as promoting industry standards, advancing interoperability, supporting societal and/or environmental efforts, or attracting and enabling our external development community. **Our increasing engagement with open source software will also cause us to license our intellectual property rights broadly in certain situations.**

Microsoft Corporation Annual Report, Form 10-K at 27 (July 28, 2022) (emphasis added).[37]

230. GitHub, which offers the Copilot product jointly with OpenAI, also has extensive experience with the DMCA. GitHub knows or reasonably should know that the Licensed Materials it hosts are subject to copyright. It provides the language of the Suggested Licenses to users, all of which

[37] https://microsoft.gcs-web.com/static-files/07cf3c30-cfc3-4567-b20f-f4b0f0bd5087/.

include copyright notices. Its 2022 Transparency Report—January to June[38] states: "Copyright-related takedowns (which we often refer to as DMCA takedowns) are particularly relevant to GitHub because so much of our users' content is software code and can be eligible for copyright protection."[39] In the first six months of 2022, GitHub processed 1220 DMCA takedown requests. Its DMCA Takedown Policy[40] notes "GitHub probably never would have existed without the DMCA."

231. GitHub also knows or reasonably should know the portions of the DMCA giving rise to Plaintiffs' claim. In its 2021 Transparency Report, "Before removing content based on alleged circumvention of copyright controls (under Section 1201 of the US DMCA or similar laws in other countries), we carefully review both the legal and technical claims, and we sponsor a Developer Defense Fund to provide developers with meaningful access to legal resources."[41]

232. GitHub is aware that Copilot's removal of CMI is illegal. For example, it states that "publishing or sharing tools that enable circumvention are not [permitted]"[42] and "Distributing tools that enable circumvention is prohibited, even if their use by developers falls under the exemption [for security research]."[43] GitHub has also frequently published articles discussing the DMCA, its application, and the Copyright Office's guidance on its scope and exceptions.[44]

233. Unless Defendants are enjoined from violating the DMCA, Plaintiffs and the Class will suffer great and irreparable harm by depriving them of the right to identify and control the reproduction and/or distribution of their copyrighted works, to have the terms of their open-source licenses followed, and to pursue copyright-infringement remedies. Defendants will not be damaged if they are required to comply with the DMCA. Plaintiffs and the Class are therefore entitled to an injunction barring

---

[38] https://github.blog/2022-08-16-2022-transparency-report-january-to-june/.

[39] https://github.blog/2022-08-16-2022-transparency-report-january-to-june/.

[40] https://docs.github.com/en/site-policy/content-removal-policies/dmca-takedown-policy#what-is-the-dmca/.

[41] https://github.blog/2022-01-27-2021-transparency-report/.

[42] https://github.blog/2020-11-19-take-action-dmca-anti-circumvention-and-developer-innovation/#what-dmca-exemptions-do-not-do/.

[43] https://github.blog/2021-11-23-copyright-office-expands-security-research-rights/.

[44] *See, e.g.*, Footnotes43–46.

Defendants from violating the DMCA and impounding any device or product that is in the custody or control of Defendants and that the court has reasonable cause to believe was involved in a violation of the DMCA.

234. Defendants conspired together and acted jointly and in concert pursuant to their scheme to commit the acts that violated the DMCA alleged herein.

235. Defendants induced (or will induce) Copilot users to unknowingly violate the DMCA by withholding attribution, licensing, and other information as described herein.

COUNT 2
BREACH OF CONTRACT—OPEN-SOURCE LICENSE VIOLATIONS
California Common Law
(Against All Defendants)

236. Plaintiffs and the Class hereby repeat and incorporate by reference each preceding and succeeding paragraph as though fully set forth herein.

237. Plaintiffs and the Class offer code under various Licenses, the most common of which are set forth in Appendix A. Use of each of the Licensed Materials is allowed only pursuant to the terms of the applicable Suggested License.

238. Plaintiffs and the Class granted Defendants a license to copy, distribute, and/or create Derivative Works under the Suggested Licenses. Each of the Suggested Licenses requires at least (1) that attribution be given to the owner of the Licensed Materials used, (2) inclusion of a copyright notice for the Licensed Materials used, and (3) inclusion of the terms of the applicable Suggested License. When providing Output, Copilot does not comply with any of these terms.

239. Defendants accepted the terms of Plaintiffs' and the Class's Licenses when it used the licensed code to create Copilot and when it incorporated the licensed code into Copilot. They have accepted and continue to accept the applicable Licenses every time Copilot Output's Plaintiffs' or the Class's copyrighted code. As such, contracts have been formed between Defendants on the one hand and Plaintiffs and the Class on the other.

240. Plaintiffs and the Class have performed each of the conditions, covenants, and obligations imposed on them by the terms of the License associated with their Licensed Materials.

241. Plaintiffs and members of the Class hold the copyright in the contents of one or more code repositories that have been hosted on GitHub's platform.

242. Plaintiffs and the Class have appended one of the Suggested Licenses to each of the Licensed Materials.

243. Plaintiffs and the Class did not know about, authorize, approve, or license the Defendants' use of the Licensed Materials in the matter at issue in this Complaint before they were used by Defendants.

244. Defendants have substantially and materially breached the applicable Licenses by failing to provide the source code of Copilot nor a written offer to provide the source code upon the request of each licensee.

245. Defendants have substantially and materially breached the applicable Licenses by failing to provide attribution to the creator and/or owner of the Licensed Materials.

246. Defendants have substantially and materially breached the applicable Licenses by failing to include copyright notices when Copilot Outputs copyrighted OS code.

247. Defendants have substantially and materially breached the applicable Licenses by failing to identify the License applicable to the Work and/or including its text when Copilot Outputs code including a portion of a Work.

248. Plaintiffs and the Class have suffered monetary damages as a result of Defendants' conduct.

249. The conduct of Defendants is causing and, unless enjoined and restrained by this Court, will continue to cause Plaintiffs and the Class great and irreparable injury that cannot fully be compensated or measured in money.

250. As a direct and proximate result of these material breaches by Defendants, Plaintiffs and the Class are entitled to an injunction requiring Defendants to comply with all the terms of any License governing use of code that was used to train Copilot, otherwise incorporated into Copilot, and/or reproduced as Output by Copilot.

251. Plaintiffs and the Class are further entitled to recover from Defendants the damages Plaintiffs and the Class sustained—including consequential damages—for Plaintiffs' and the Class's costs

in enforcing their contractual rights. Plaintiffs and the Class are also entitled to recover as restitution from Defendants for any unjust enrichment, including gains, profits, and advantages that Defendants have obtained as a result of their breach of contract.

COUNT 3
BREACH OF CONTRACT — SELLING LICENSED MATERIALS IN VIOLATION OF GITHUB'S POLICIES
California Common Law
(Against GitHub)

252. Plaintiffs and the Class hereby repeat and incorporate by reference each preceding and succeeding paragraph as though fully set forth herein.

253. GitHub's Privacy Statement, Terms of Service, and GitHub Copilot Terms share definitions and refer to each other. As such, they are collectively referred to herein as "GitHub's Policies" unless a distinction is necessary and are attached as Exhibit 1.

254. Plaintiffs and the Class are GitHub users who have accepted GitHub's Policies. As a result, Plaintiffs and the Class have formed a contract with GitHub.

255. Plaintiffs and the Class have performed each of the conditions, covenants, and obligations imposed on them by the terms of GitHub's Policies.

256. GitHub's Policies contain multiple explicit provisions that GitHub will not sell the Licensed Materials of the Plaintiffs and Class. GitHub's Terms of Service document provides that the "License Grant to [GitHub] . . . does not grant GitHub the right to sell Your Content." Similarly, GitHub's Privacy Statement defines "personal data" to include "any . . . documents, or other files", a definition that necessarily comprises source code, and hence the Licensed Materials. (As of May 2023, GitHub has updated this provision on its website to explicitly read "any code, text, … documents, or other files"). Elsewhere, the Privacy Statement provides "We do not sell your personal information," "No selling of personal data," "We *do not* sell your personal data for monetary or other consideration." (Emphasis in original).

257. By making the Licensed Materials available through Copilot in violation of the Suggested Licenses, and charging subscription fees, GitHub has been selling Licensed Materials. By selling the Licensed Materials, GitHub has breached these provisions in GitHub's Policies against selling user data.

258. GitHub has also breached the implied covenant of good faith and fair dealing. GitHub has long held itself out as a good citizen of the global open-source community. GitHub's Policies were designed to attract Plaintiffs and the Class to become users of the GitHub website by supporting their open-source efforts with fair and ethical practices. By releasing Copilot, GitHub created a product designed to compete with Plaintiffs and the Class and undermine their individual open-source communities. In so doing, GitHub did not act fairly or in good faith.

259. Plaintiffs and the Class have suffered monetary damages as a result of GitHub's conduct.

260. GitHub's conduct is causing and, unless enjoined and restrained by this Court, will continue to cause Plaintiffs and the Class great and irreparable injury that cannot fully be compensated or measured in money.

261. As a direct and proximate result of these material breaches by GitHub, Plaintiffs and the Class are entitled to an injunction requiring GitHub to comply with all the terms of the GitHub Policies.

262. Plaintiffs and the Class are further entitled to recover from GitHub the damages Plaintiffs and the Class sustained—including consequential damages—for Plaintiffs' and the Class's costs in enforcing GitHub's Policies. Plaintiffs and the Class are also entitled to recover as restitution from GitHub for any unjust enrichment, including gains, profits, and advantages that it has obtained as a result of its breaches of the GitHub Policies.

IX. DEMAND FOR JUDGMENT

WHEREFORE, Plaintiffs requests that the Court enter judgment on their behalf and on behalf of the Class defined herein, by adjudging and decreeing that:

263. This action may proceed as a class action, with Plaintiffs serving as Class Representatives, and with Plaintiffs' counsel as Class Counsel;

a) Judgment in favor of Plaintiffs and the Class and against Defendants;

b) Permanent injunctive relief, including but not limited to making changes to its Copilot product to ensure that all applicable information set forth in 17 U.S.C. § 1203(b)(1) is included in along with any Output including associated code;

c) An order of costs and allowable attorney's fees pursuant to 17 U.S.C. § 1203(b)(4)–(5);

d) An award of damages for harms resulting from Defendants' breach of Licenses;

e) An award of damages in the amount Defendants have been unjustly enriched through their conduct as alleged herein as well as punitive damages in connection with this conduct;

f) An award of damages for harms resulting from GitHub's breach of the GitHub Policies; and

264. Injunctive relief sufficient to alleviate and stop Defendants' unlawful conduct alleged herein.

265. Plaintiffs and the Class are entitled to prejudgment and post-judgment interest on the damages awarded them, and that such interest be awarded at the highest legal rate from and after the date this class action complaint is first served on Defendants;

266. Defendants are to be jointly and severally responsible financially for the costs and expenses of a Court approved notice program through post and media designed to give immediate notification to the Class.

267. Plaintiffs and the Class receive such other or further relief as may be just and proper.

X. JURY TRIAL DEMANDED

Pursuant to Federal Rule of Civil Procedure 38(b), Plaintiffs demand a trial by jury of all the claims asserted in this Complaint so triable.

Dated: January 24, 2024

By: */s/ Joseph R. Saveri*
Joseph R. Saveri

Joseph R. Saveri (State Bar No. 130064)
Cadio Zirpoli (State Bar No. 179108)
Christopher K.L. Young (State Bar No. 318371)
Louis A. Kessler (State Bar No. 243703)
Elissa A. Buchanan (State Bar No. 249996)
Travis Manfredi (State Bar No. 281779)
William W. Castillo Guardado (State Bar No. 294159)
Holden J. Benon (State Bar No. 325847)
JOSEPH SAVERI LAW FIRM, LLP
601 California Street, Suite 1000
San Francisco, California 94108
Telephone: (415) 500-6800
Facsimile: (415) 395-9940
Email: jsaveri@saverilawfirm.com
czirpoli@saverilawfirm.com
cyoung@saverilawfirm.com
lkessler@saverilawfirm.com
eabuchanan@saverilawfirm.com
tmanfredi@saverilawfirm.com
wcastillo@saverilawfirm.com
hbenon@saverilawfirm.com

Matthew Butterick (State Bar No. 250953)
1920 Hillhurst Avenue, #406
Los Angeles, CA 90027
Telephone: (323) 968-2632
Facsimile: (415) 395-9940
Email: mb@buttericklaw.com

Counsel for Individual and Representative Plaintiffs and the Proposed Class