Magisterarbeit im Masterstudiengang
Europäisches Recht und Rechtsvergleich, Deutsches Recht und Rechtspraxis
Kooperationsmaster mit King's College London
Humboldt-Universität zu Berlin
Juristische Fakultät

Free and Open Source Software Licensing Requirements and Copyright Infringement Involving Artificial Intelligence Technologies

Gutachter:
Prof. Dr. Herbert Zech und Dr. Lucas Lasota

Linda Novobilská

Berlin, 2. Juni 2023

Acknowledgement

I would like to express my sincere gratitude to my supervisors Prof. Dr. Herbert Zech and Dr. Lucas Lasota for their guidance and helpful feedback throughout the writing and publishing process. Their expertise and suggestions helped me to complete this research and write this thesis.

Table of Contents

A. Bibliography

Alford, Anthony, OpenAI Announces 12 Billion Parameter Code-Generation AI Codex, https://www.infoq.com/news/2021/08/openai-codex/, 31 August 2021, accessed 6 May 2023.

Band, Jonathan, The Google Library Project: Both Sides of the Story, in Ann Arbor, MI: MPublishing, University of Michigan Library, 2006, vol. I.

Barke, Shraddha; James, Michael B; Polikarpova, Nadia' Grounded Copilot: How programmers interact with code-generating models, 2022.

Rossi, Francesca; Mitchell, Margaret; Jernite, Yacine; Ilić, Suzana and McDuff, Daniel, BigScience RAIL License, https://bigscience.huggingface.co/blog/the-bigscience-rail-license, accessed 3 April 2023.

Butterick, Matthew, GitHub Copilot litigation, https://web.archive.org/web/20221103204107/https:/githubcopilotlitigation.com/pdf/1-0-github_complaint.pdf, 3 November 2022, accessed 3 March 2023.

Chittock, Sarah, Getty Images taking UK action against Stability AI for copyright infringement in AI training, Lexis Nexis Legal News, 24 January 2023.

Choksi, Madiha Zahrah; Goedicke, David, Whose Text Is It Anyway? Exploring BigCode, Intellectual Property, and Ethics, 2023, Second Workshop on Intelligent and Interactive Writing Assistants co-located with the ACM CHI Conference on Human Factors in Computing Systems. pp.1-3.

Christian, Jon, CNET's AI Journalist Appears to Have Committed Extensive Plagiarism, Futurism, https://futurism.com/cnet-ai-plagiarism, January 2023, accessed 7 April 2023.

Curinga, Matthew; Wentworth, Peter; Elkner, Jeffrey; Downey, Allen B; Meyers, Chris; Think Javascript, https://matt.curinga.com/think-js/#solving-problems-with-for-loops, accessed 7 May 2023.

de Castilho, Eckart R; Gurevych, Iryna; Dore, Giulia; Margoni, Thomas; Labropoulou, Penny, A Legal Perspective on Training Models for Natural Language Processing, 2018, in Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), p. 1267-1274.

Dohmke, Thomas, 100 million developers and counting, The GitHub Blog, https://github.blog/2023-01-25-100-million-developers-and-counting/, 25 January 2023, accessed 23 March 2023.

De Laat, Paul, Copyright or Copyleft? An Analysis of Property Regimes for Software Development, Research Policy, 2005, Vol. 34/10, p. 1511-1532.

Denicola, Robert, Ex Machina: Copyright Protection for Computer-Generated Works, Rutgers University Law Review, 2016, p. 251-287.

Eechoud Mireille et al., Harmonizing European Copyright Law – The Challenges of Better Lawmaking, Kluwer Law International, 2011, vol. 19.

Eechoud, Mireille M; Hugenholtz, P. Bernt; van Gompel, Stef; Guibault, L.; Helberger, Natali, Harmonizing European Copyright Law: The Challenges of Better Lawmaking, Information Law Series 19, in Amsterdam Law School Research Paper No. 2012-07, 2012.

Enabling Easier Collaboration on Open Data for AI and ML with CDLA-Permissive-2.0, https://www.linuxfoundation.org/press/press-release/enabling-easier-collaboration-on-open-data-for-ai-and-ml-with-cdla-permissive-2-0, Linux Foundation, accessed 2 April 2023.

Engler, Alex, How open-source software shapes AI policy, Brookings, https://www.brookings.edu/research/how-open-source-software-shapes-ai-policy/, 10 August 2021, accessed 3 March 2023.

Field, Hayden; McDonald, Jordan; Donnelly, Grace, Three inflection points for emerging tech in 2022, TechBrew, https://www.emergingtechbrew.com/stories/2022/12/21/three-inflection-points-for-emerging-tech-in-2022, 21 December 2022, accessed 21 April 2023.

Finley, Klint, The Problem With Putting All the World's Code in GitHub, https://web.archive.org/web/20150629152927/http://www.wired.com/2015/06/problem-putting-worlds-code-github/, archived from Wired, accessed 5 March 2023.

Fitzpatrick, Stuart, On the Nature of AI Code Copilots, https://www.fsf.org/licensing/copilot/on-the-nature-of-ai-code-copilots, Free Software Foundation, 24 February, 2022, accessed 17 March 2023.

Forrester, Justin E; Miller Barton P, An Empirical Study of the Robustness of Windows NT Applications Using Random Testing, 4th USENIX Windows Systems Symposium, 27 July 2000.

Free Software Foundation, Proprietary Insecurity, https://www.gnu.org/proprietary/proprietary.html, accessed 20 February 2023.

Gay, Joshua, The Principles of Community-Oriented GPL Enforcement, https://www.fsf.org/licensing/enforcement-principles, accessed 14 March 2023.

Géron, Aurélien, Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, O'Reilly Media Inc., 2019.

Getty Images, Getty Images Statement, https://newsroom.gettyimages.com/en/getty-images/getty-images-statement, 17 January 2023, accessed 12 April 2023.

Ginsburg, Jane, No 'Sweat?' Copyright and Other Protection of Works of Information after Feist v. Rural Telephone, Columbia Law Review, 1992, 92:338–388.

GitHub Copilot, Your AI Programmer, https://github.com/features/copilot/, accessed 2 February 2023.

GitHub, Licensing a repository, https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/licensing-a-repository, accessed 2 May 2023.

GitHub, GitHub Number of Repositories, https://github.com/search, archived from the original on 25 January 2023, accessed March 5, 2023.

GitHub, Repository search for public repositories, *https://web.archive.org/web/20201105194800/https:/github.com/search?q=is:public archived from the original on 5 November 2020*, accessed 25 March 2023.

GitHub, Terms of Service, https://docs.github.com/en/site-policy/github-terms/github-terms-of-service#a-definitions, accessed 13 May 2023.

Gleick, James, The Information - A History, a Theory, a Flood, Vintage Books, 2011.

Goldstein, Paul; Hugenholtz, Berndt P, International Copyright Principles, Law, and Practice, Oxford University Press, 2013.

Gonzalez-Barahona, Jesus M., A Brief History of Free, Open-Source Software and Its Communities, Computer, vol. 54/2, 2021, pp. 75-79.

HM Government, National AI Strategy, Presented to Parliament by the Secretary of State for Digital, Culture, Media and Sport by Command of Her Majesty, Gov.UK, September 2021.

Grimmelmann, James, Regulation by software, Yale Law Journal, Vol. 114, 2005, pp.1719-58.

Grimmelmann, James, Copyright for Literate Robots, Iowa Law Review 101, 2015, pp. 657–664.

Growcoot, Matt, Getty Images is Suing Stable Diffusion for a Staggering $1.8 Trillion, https://petapixel.com/2023/02/07/getty-images-are-suing-stable-diffusion-for-a-staggering-1-8-trillion/, 7 February 2023, accessed 5 May 2023.

Her Majesty's Intellectual Property Office "Artificial Intelligence and Intellectual Property: copyright and patents: Government response to consultation", https://www.gov.uk/government/consultations/artificial-intelligence-and-ip-copyright-and-patents/outcome/artificial-intelligence-and-intellectual-property-copyright-and-patents-government-response-to-consultation, 28 June 2022, accessed 12 March 2023.

Howard, Gavin D, My Whitepaper About GitHub Copilot, GitHub Copilot: Copyright, Fair Use, Creativity, Transformativity, and Algorithms, https://gavinhoward.com/uploads/copilot.pdf, 27 October 2021, accessed 25 February 2023.

Hugenholtz, Berndt P, Something Completely Different: Europe's Sui Generis Database Right, in Frankel S. and Gervais D. (Eds.), The Internet and the Emerging Importance of New Forms of Intellectual Property, Information Law Series, Vol. 37, Kluwer Law International, 2016, 205 – 222.
Cited as "Hugenholtz, 2016, page"

Hugenholtz, Bernt; Senftleben, Martin, Fair Use in Europe: In Search of Flexibilities 14 November 2011, Amsterdam Law School Research Paper No. 2012-39, Institute for Information Law Research Paper No. 2012-33, SSRN, 2-14.
Cited as "Hugenholtz, 2011, page"

Jaeger, Till; Metzger, Axel, Open Source Software, Rechtliche Rahmenbedingungen der Freien Software, 5. Aufl. 2020, S. 73

Jose, Jomon P, Legal Liability Issues and Regulation of Artificial Intelligence (AI), Legal Dissertation, National Law School of India University Bengaluru, 2018.

Jütte, Bernd J, The New Copyright Directive: Digital and Cross-border Teaching Exception (Article 5), https://copyrightblog.kluweriplaw.com/2019/06/21/the-new-copyright-directive-digital-and-cross-border-teaching-exception-article-5/, Kluwer Copyright Blog, 21 June 2019, accessed 27 March 2023.

Kelly, Colin P Jr, How GitHub hit $200M Revenue with 40M customers in 2023, https://getlatka.com/companies/github, GitHub, accessed 1 March 2023.

Konar, Amit, Artificial Intelligence and Soft Computing - Behavioral and Cognitive Modeling of the Human Brain, CRC Press, Boca Raton, 1999.

Kop, Mauritz, AI & Intellectual Property: Towards an Articulated Public Domain, Texas Intellectual Property Law Journal, 2020, p. 1-39.

Krempl, Stefan, Die Stimmen der Revolutionäre, https://www.heise.de/tp/features/Die-Stimmen-der-Revolutionaere-3495044.html, 15 July 1999, accessed 15 March 2023.

Kuhn, Bradley M, If Software is My Copilot Who Programmed My Software?, Software Freedom Conservancy, https://sfconservancy.org/blog/2022/feb/03/github-copilot-copyleft-gpl/, 3 February 2022, accessed 17 February 2023.

Lasota, Lucas, Free Software Licensing, https://download.fsfe.org/presentations/20221128-free-software-licensing-bmbf-lucas-lasota.pdf, FSFE, 28 November 2022, accessed 6 March 2023.

Lasota, Lucas, What is a license, https://www.sfscon.it/talks/what-is-a-license/, SFSCON, 11 November 2022, accessed 8 March 2023.

Lazarova, Ana; Margoni, Thomas; Matas, Ariadna; Pearson, Sarah; Reda, Julia; Vézina, Brigitte; Walsh, Kat; Wyber, Stephen, Creative Commons Statement on the Opt-Out Exception Regime, https://creativecommons.org/wp-content/uploads/2021/12/CC-Statement-on-the-TDM-Exception-Art-4-DSM-Final.pdf, Creative Commons, 17 December 2022, accessed 12 May 2023.

Lemley, Mark A; Casey, Brian, Fair Learning, Texas Law Review, vol. 99/4, 2021.

Levendowski, Amanda, How Copyright Law Can Fix Artificial Intelligence's Implicit Bias Problem, 93 Washington Law Review, 2018, p.592.

Lin, Yi-Hsuan; Ko, Tung-Mei; Chuang, Tyng-Ruey; Lin, Kwei-Jay, Open Source Licenses and the Creative Commons Framework: License Selection and Comparison, Journal of Information Science and Engineering, vol. 22(1) 1-17, 2006.

Liptak, Adam; Alter, Aleksandra, Challenge to Google Books is declined by Supreme Court, New York Times, https://www.nytimes.com/2016/04/19/technology/google-books-case.html, 18 April 2016, accessed 2 March 2023.

Nicholas, Katrina; Robertson, Jonas, Teen hacker says he's found way to remotely control 25 Tesla EVs around the world, https://fortune.com/2022/01/12/teen-hacker-david-colombo-took-control-25-tesla-ev/ , Fortune, 12 January 2022, accessed 23 March 2023.

Margoni, Thomas, The harmonisation of EU copyright law: The originality standard, in Mark Perry (Ed.), Global Governance of Intellectual Property in the 21st Century, 2016, pp. 85–105.
Cited as "Margoni, Global Governance of Intellectual Property in the 21st Century 2016, page".

Margoni, Thomas, Artificial Intelligence, Machine Learning and EU Copyright Law: Who owns AI?, CREATe Working Paper, vol. 2018/12, 2018.
Cited as "Margoni, CREATe Working Paper 2018, page."

Margoni, Thomas; Kretschmer, Martin, A Deeper Look into the EU Text and Data Mining Exceptions: Harmonisation, Data Ownership, and the Future of Technology, *GRUR International*, vol. 71/8, 2022, pp. 685–701.
Cited as "Margoni, GRUR International 2022, page".

Masouyé, Claude, Guide to the Berne Convention, WIPO, 1978, 76-7.

McKusick, Marshall K, Twenty years of Berkeley Unix: From AT&T-owned to freely redistributable. In: DiBona S. Ockman & M. Stone (ed.), O'Reilly and Associates, 1999. pp. 31-46.

Mezei Peter, From Leonardo to the Next Rembrandt – The Need for AI-Pessimism in the Age of Algorithms, UFITA, vol. 2/2020, pp. 390-429.

Miller Barton P; Koski, David; Lee, Cjin P; Maganty, Vivekananda; Murthy, Revi; Natarajan, Ajitkumar; Steidl, Jeff; Fuzz Revisited: A Re-examination of the Reliability of UNIX Utilities and Services, Computer Sciences Technical Report #1268, University of Wisconsin-Madison, 1995.
Cited as "Miller, Computer Sciences Technical Report University of Wisconsin-Madison 1995, page".

Miller Barton P; Zhang, Mengxiao; Heymann Elisa R, The Relevance of Classic Fuzz Testing:Have We Solved This One?, IEEE Transactions on Software Engineering, , 2021, pp.1-10.
Cited as "Miller, IEEE Transactions on Software Engineering 2021, page".

Moglen, Eben, Anarchism triumphant: Free software and the death of copyright, First Monday, vol. 4(8), 1999.

Nabi, Rebaz M; Nabi, Rebwar M; Mohammed, Rania A; Open Source Development (OSS) under Eclipse Public License (EPL), International Journal of Advanced Research, vol 3/12, 2015, 677 – 686.

New Committee Will Investigate Copyleft Implications of AI-Assisted Programming, Software Freedom Conservancy, https://sfconservancy.org/news/2022/feb/23/committee-ai-assisted-software-github-copilot/, 23 February 2022, accessed 21 May 2023.

Nilsson, Nils J, The Quest for Artificial Intelligence: A History of Ideas and Achievements, Cambridge University Press, 2010.

Nzabandora, Bertrand; Davis-White, Alex; UK: Proposed changes to copyright law to facilitate data mining, https://www.allenovery.com/en-gb/global/blogs/digital-hub/proposed-changes-to-copyright-law-to-facilitate-data-mining, Allen & Overy, 21 July 2022, accessed 13 May 2023.

Ornes, Stephen, The Unpredictable Abilities Emerging From Large AI Models, Quanta Magazine, https://www.quantamagazine.org/the-unpredictable-abilities-emerging-from-large-ai-models-20230316/, 16 March 2023, accessed 27 April 2023.

Palazzetti, Laura; Mazzi, Francesca; Artificial Intelligence and the Challenges of the Fashion Industries, https://www.iusinitinere.it/artificial-intelligence-and-the-challenges-of-the-fashion-industries-29023, Ius In Itinere, 25 June 2020, accessed 1 March 2023.

Peterson, Christine, How I coined the term 'open source', https://opensource.com/article/18/2/coining-term-open-source-software, Open Source.com, 1 February 2018, accessed 12 March 2023.

GNU Operating System, Philosophy of the GNU Project, https://www.gnu.org/philosophy/free-sw#n1, accessed 2 May 2023.

Radford, Alec; Narasimhan, Karthik; Salimans, Tim; Sutskever, Ilya; Improving language understanding with unsupervised learning, Technical Report, Papers With Code,2018. Cited as “Radford, Papers With Code 2018, page“.

Radford, Alec; Wu, Jeffrey; Child, Rewon; Luan, David; Amodei, Dario; Sutskever, Ilya; Language Models are Unsupervised Multitask Learners, 2019.
Cited as “Radford (2019), page”.

Ricketson, Sam; Ginsburg, Jane; International Copyright and Neighbouring Rights – The Berne Convention and Beyond, OUP, 2005, 8.05.

Seddon, Robert F J; Copilot, Copying, Commons, Community, Culture, https://www.fsf.org/licensing/copilot/copilot-copying-commons-community-culture, Free Software Foundation, 24 February 2022, accessed 11 March 2023.

Romero, Alberto, GitHub Copilot — A New Generation of AI Programmers, https://towardsdatascience.com/github-copilot-a-new-generation-of-ai-programmers-327e3c7ef3ae, Towards Data Science, 1 July 2021, accessed 17 March 2023.

Rosati, Eleonora, Copyright in the Digital Single Market: a taster, WIPO Magazine, vol 4/2021, https://www.wipo.int/wipo_magazine/en/2021/04/article_0009.html, December 2021, accessed 7 February 2023.

Rothchild, John A; Rothchild, Daniel H; Copyright Implications of the Use of Code Repositories to Train a Machine Learning Model, https://www.fsf.org/licensing/copilot/copyright-implications-of-the-use-of-code-repositories-to-train-a-machine-learning-model, Free Software Foundation, 24 February 2022, accessed 17 February 2023.

Russell, Stuart J; Norvig, Peter; Artificial intelligence: a modern approach, Pearson Education, 2009.

Salokannel, Marjut; Strowel, Alain; Final report: Study contract concerning moral rights in the context of the exploitation of works through digital technology, commissioned by the European Commission's Internal Market Directorate-General, 2000, pp. 1-252.

Sassi, Silhem B; Nesrine, Sbai; Characterizing open source software licenses texts: an insight from legal terms perspective, RIADI Laboratory, 2022.

Schatten, Jeff; Will Artificial Intelligence Kill College Writing?, https://www.chronicle.com/article/will-artificial-intelligence-kill-college-writing, The Chronicle of Higher Education, 14 September 2022, accessed 5 May 2023.

Seemann, Mark, The 80/24 Rule, https://blog.ploeh.dk/2019/11/04/the-80-24-rule/#:~:text=If%20there%27s%20any%20accepted%20industry,line%20width%2C%20it%27s%2080%20characters, 4 November 2019, accessed 19 March 2023.

Senftleben, Martin et al., Ensuring the Visibility and Accessibility of European Creative Content on the World Market – The Need for Copyright Data Improvement in the Light of New Technologies and the Opportunity Arising from Article 17 of the CDSM Directive, Journal of Intellectual Property, Information Technology and Electronic Commerce Law 67, vol. 13/1, 2022, pp. 67-86,

Sinclair, Andrew, Licence Profile: BSD, The Journal of Open Law, Technology & Society, vol. 2/1, 2010.

Software Freedom Conservancy, On the filing of the Class Action Law Suit over GitHub's Copilot, https://sfconservancy.org/news/2022/nov/04/class-action-lawsuit-filing-copilot/, 4 November 2022, accessed 1 March 2023.

Stallman Richard, Why Open Source Misses the Point of Free Software, https://www.gnu.org/philosophy/open-source-misses-the-point.en.html, GNU Operating system, accessed 17 March 2023.

Stamatoudi, Irina; Torremans, Paul (Eds.), Copyright in the New Digital Environment: The Need to Redesign Copyright. Sweet & Maxwell, 2000.

St. Laurent, Andrew M, Understanding Open Source and Free Software Licensing, O'Reilly Media Inc., 2004.

Sutrop, Margit, Challenges of Aligning Artificial Intelligence with Human Values, Acta Baltica Historiae et Philosophiae Scientiarum, vol. 8/2, 2020, pp.55-69.

Tai Li, Cheng, The History of the GPL, https://www.free-soft.org/gpl_history/, 4 July 2001, accessed 1 May 2023.

UK Department for DCMS, Office for AI, New UK initiative to shape global standards for Artificial Intelligence, https://www.gov.uk/government/news/new-uk-initiative-to-shape-global-standards-for-artificial-intelligence, 12 January 2022, accessed 21 February 2023.

United States Patent and Trademark Office, Comment Regarding Request for Comments on Intellectual Property Protection for Artificial Intelligence Innovation, https://www.uspto.gov/sites/default/files/documents/OpenAI_RFC-84-FR-58141.pdf, 2019.

Vaughan-Nichols, Stephen J, GitHub's Copilot flies into its first open source copyright lawsuit, https://www.theregister.com/2022/11/11/githubs_copilot_opinion/, The Register, 11 November 2022, accessed 22 February 2023.

Vézina, Brigitte; Hinchliff-Pearson, Sarah; Should CC-Licensed Content be Used to Train AI? It Depends, https://creativecommons.org/2021/03/04/should-cc-licensed-content-be-used-to-train-ai-it-depends/ , Creative Commons, 4 March 2021, accessed 7 April 2023.

Vincent, James, The lawsuit that could rewrite the rules of AI copyright, https://www.theverge.com/2022/11/8/23446821/microsoft-openai-github-copilot-class-action-lawsuit-ai-copyright-violation-training-data, The Verge, 8 November 2022, accessed 28 February 2023.

Wadhwani, Sumeet, Microsoft, GitHub and OpenAI Accused of Software Piracy Sued for $9B in Damages, https://www.spiceworks.com/tech/artificial-intelligence/news/github-copilot-class-action-lawsuit/, SpiceWorks, 16 December 2022, accessed 2 March 2023.

Walter, Michel; von Lewinski, Silke (Eds.), European Copyright Law A Commentary, Oxford University Press, 2010.

Weber, Steven, The Success of Open Source, in The Success of Open Source, Harvard University Press, 2005.

Wei, Jason et al., Emergent Abilities of Large Language Models, Transaction on Machine Learning Research, vol. 08/2022.

Wheeler, David A, Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!, https://dwheeler.com/oss_fs_why.html, accessed 16 March 2023.

Wiggers, Kyle, Commercial image-generating AI raises all sorts of thorny legal issues, https://techcrunch.com/2022/07/22/commercial-image-generating-ai-raises-all-sorts-of-thorny-legal-issues/ , 22 July 2022, Tech Crunch, accessed 6 April 2023.

Wiggers, Kyle, The current legal cases against generative AI are just the beginning, https://techcrunch.com/2023/01/27/the-current-legal-cases-against-generative-ai-are-just-the-beginning/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAM32E1ub5qz3LTNg7-Zu13tVt81Dg_TQguogGlR_yU2aBZqKEsXfaArxAJ1YrxK_S1KyNq8QLYX2UkwTYvbsycjpF1IqkZGUVDPpFp2OlwnupSYCNMfvILuIkDd0cqd66XPnTND6SbRWY0KTFAQduMEG7zJPVT0qj1e603xE7X-7 , Tech Crunch, 27 January 2022, accessed 3 April 2023.

Wiggers, Kyle, Image-generating AI can copy and paste from training data, raising IP concerns, https://techcrunch.com/2022/12/13/image-generating-ai-can-copy-and-paste-from-training-data-raising-ip-concerns/,13 December 2022, Tech Crunch, accessed 6 May 2023.

WIPO Secretariat, Draft Issues Paper on Intellectual Property Policy and Artificial Intelligence, WIPO/IP/AI/2/GE/20/1, 2019.

Yanisky-Ravid, Shlomit, Generating Rembrandt: Artificial Intelligence, Copyright, and Accountability in the 3A Era - The Human-Like Authors Are Already Here - A New Model, Michigan State Law Review, 2017, pp. 659-726.

B. Introduction

Artificial intelligence ("AI") has seen unprecedented developments in the last decade.[1] Its rapid growth and ability to increase efficiency, accuracy, effectivity and precision within a range of processes have allowed it to permeate substantial parts of our daily lives.[2] From AI-powered translation tools to word processing applications, navigation systems, customer service platforms and "chatbots", to industrial uses in robotics and heavy machinery, AI is met in all facets.[3] However, its great potential is also met with a range of challenges - socially, economically as well as legally. In particular, this thesis focuses on the legal challenges of AI from the perspective of intellectual property.

Intellectual property is an area of law that regulates copyright, patents, trademarks, designs and allied rights – in other words, a series of rights that protect an individual's intellectual contribution. Copyright is an exclusive legal right that is created automatically by virtue of original authorship and applies to a range of subjects, giving the author exclusive control over the use and distribution of the creation.[4]

It is necessary to distinguish between different copyright issues pertaining to AI. These can be categorised into three main groups - the copyright of the AI itself, the copyright of AI output and the copyright of AI input. This thesis places an emphasis on the latter group, which centres around the conceptual issue of AI input that the AI uses in its training stage, such as for machine learning ("ML"). This third process poses one of the biggest challenges legally because AI has reached a level where it can, and indeed needs to, process vast amounts of data for learning. In particular, this thesis considers licensed online software code in ML. This is interwoven with a range of ethical and moral questions with which policymakers grapple.[5] In particular, AI models being trained on code licensed under Free Open Source Software ("FOSS") licenses pose a significant copyright problem. And AI can be trained on code that has been published and made accessible online for free under licenses that impose legal conditions on the use of the code, such as attribution. However, the code is used by an AI in a way which does not comply with these conditions. This dilemma lies at the centre of this thesis, which considers the recent lawsuit against GitHub Copilot, an online platform that trains on FOSS code.

This thesis discusses whether AI training using copyleft software might be against the FOSS's movement to assure software freedom even after it has been modified, as well as in violation of legal norms. Secondly, the question of how different the output code must be from the input code protected by copyright is analysed. Thirdly, the thesis discusses whether the use of copyrighted works falls under any exceptions to copyright violations such as fair use or text and data mining. Fourthly, the thesis discusses possible actions that can be taken to ensure that large AIs do not violate neither FOSS principles nor the law.[6]

These questions will be discussed with reference to the following scheme. Firstly, this essay turns to discuss AI as a concept, its definitions, and an overview of natural language processing.

---

[1] Jose, p. 7.
[2] Denicola, Rutgers Uni. Law Rev. 2016, 253.
[3] Palazzetti, https://www.iusinitinere.it/artificial-intelligence-and-the-challenges-of-the-fashion-industries-29023, Artificial Intelligence and the Challenges of the Fashion Industries, accessed 1 March 2023.
[4] Directive 2009/24/EC, rec. 6.
[5] New Committee Will Investigate Copyleft Implications of AI-Assisted Programming, https://sfconservancy.org/news/2022/feb/23/committee-ai-assisted-software-github-copilot/, accessed 21 May 2023.
[6] *Ibid.*

Secondly, the copyright framework in the EU is discussed. This thesis takes a comparative approach and contrasts the EU frameworks with the US position, as a jurisdiction where much of the AI litigation can be expected to unfold. Thirdly, the essay turns to discussing the licensing of software code. Fourthly, the lawsuit of GitHub Copilot is discussed as a case study about an AI violating FOSS licensing. The section discusses the US as well as the EU legal position on Copilot to conclude that while in the EU, the training of AI on FOSS code qualifies for the data and text mining exception; in the US, where the lawsuit has been launched, the use should not fall within the "fair use" exception given the factors of the assessment and their interpretations in case law. Fifthly, a brief overview of a new example of AI copyright litigation is given to illustrate the significance of this legal dispute. Lastly, the essay discusses any potential reforms and changes to existing AI systems that could help improve the legal compliance of big AI.

C. Artificial Intelligence

I. Definition

The notions of AI are far from novel. First conceptualised by scientists such as Alan Turing and Claude Shannon around the mid-20th century,[7] AI has represented a conceptual challenge for many academics. In the past ten years, defining AI has never been more important. Yet, many academics would agree that settling on a firm definition of AI remains difficult.[8]

Academics such as Konar conceptualise AI's ability in identifying which information to utilise in solving a problem as the main component of "intelligence".[9] Konar's view is therefore very focused on the ability to filter information and to make value judgments about information.[10] Nilsson's view is that an "intelligent" machine is defined by its ability to have foresight.[11] The European Commission views a system as intelligent when it is capable of analysing its environment and of achieving certain goals to some degree autonomously.[12] Some writers require the system to be able to correct mistakes, adjust outputs and improve itself. These definitions are supported by some prominent thinkers such as Stephen Hawking or Mauritz Kop who view AI as a non-human system that possesses human-like "cognitive functions and skills such as reasoning and learning", thinking and planning strategically.[13]

As such, a prevailing theme amongst academics is the AI's ability to mimic or emulate human cognitive skills. This is significant, considering that the requirement of many copyright regulations is a human authorship element, which is further discussed in this essay.

II. The Programming of NLPs

Natural Language Processors ("NLPs") are a specific type of AI that are trained on vast amounts of natural language data to develop the ability to effectively emulate natural language. In terms of copyright, these types of AI pose some of the biggest issues due to the vast amounts of potentially copyrighted works on which they train. NLPs are to be distinguished from large

[7] Gleick p. 115-120.
[8] Yanisky-Ravid, Mich. St. L. Rev. 2017, 659 (673).
[9] Konar, p. 1.2
[10] *Ibid.*
[11] Nilsson, p.13.
[12] European Commission High-Level Expert Group on Artificial Intelligence (2019), 1. p.3
[13] Kop, Texas IP Law Journal 2020, 1, (4).

language models ("LLMs"), which are specific models often trained by natural language processing that are specifically trained for language-related tasks. Due to their size and training set, they are specifically developed to emulate and generate language.[14]

The process of developing an AI is significant for understanding the legal problem of copyright protection with regards to ML. The process of AI creation can generally be divided into three stages: 1. writing the source code, 2. training the AI and 3. output.

1. Writing AI Code

Firstly, the code is written. This code is often referred to as source code, which is the human-readable form of a programme. It is often written in high-level programming languages such as Java or C++ and then compiled into object code, or the computer-readable binary code generated by a compiler. Unlike the following stages of AI creation, the coding stage often requires human input.[15]

2. Training the AI

The second stage is where the AI is developed to achieve the particular goal for which it has been made. There are many ways in which an AI is developed.[16] Machine learning is commonly used, however, there are also other systems, such as rule-based systems, expert systems, and genetic algorithms to building AI systems.[17]

The first step of machine learning is called corpus compilation, where data (also called corpora) on which the AI will be trained is identified. These sources, the relevance of which will determine the quality of the AI, can range from dictionaries to thesauri, websites or books. The second step is pre-processing, during which the corpora are converted into a format that is readable to the computer, such as PDF or HTML. The third step is called corpus annotation, where this data is assigned labels that can be associated with the content. These labels are usually classified by categories such as grammar, morphology or syntax, following inventories of pre-categorised labelled text.[18] Fourthly, the model is trained, where the algorithm then proceeds to analyse the text, extracting statistical, grammatical or contextual patterns which are saved. Fifth, a permanent file is created containing the trained model.[19]

These stages are a general overview of the most common way to train an NLP. The processes may nevertheless vary, which is significant for the determination of copyright violations within NLP and AI in general.

3. Output and Emergent Works

The AI produces an output, which refers to the specific results created by the AI. AI output can be distinguished from a frequently used term of "emergent works" which refers to output which

[14] Wei, TMLR, vol. 08/2022, p.1.
[15] Mezei, UFITA 2020, 390 (395), p. 5.
[16] Géron p. 8.
[17] Russell p. 2.
[18] Margoni, CREATe Working Paper, 2018.
[19] de Castilho, LREC 2018, 1267, (1274), p.1.

the AI's programming did not intend based on the source code.[20] This has sparked a lot of debate about the ethics of AI from academics who anticipate the development of full AI by the end of the 21st century.[21] The vast amount of data involved in the training process of such AI raises significant questions regarding how license compliance can be ensured. The following section therefore turns to analyse the existing copyright frameworks.

D. Copyright Frameworks

Copyright protects unique expression.[22] It is an automatic and exclusive right of the creator of the creation to have the right to freely distribute and copy the work; in the case of software, it is protected as a unique literary work.[23] However, the author of the copyrighted work is allowed to enter into legal contracts to allow the work to be used or distributed (licenses), which are discussed further in the next sections.

This section discusses I. the copyright protection of the materials used in machine learning or natural language processing and II. analyses the applicability of certain economic exploitation rights regarding ML/NLP. These issues are considered from the perspectives of the international and EU perspectives, although copyright is national in scope and matter. The EU copyright framework refers to a system of uniform rules that should be applied by countries either directly (through regulations) and indirectly (through directives).

I. Copyright Protection and Requirements

Machine learning and natural language processing of AI most commonly occurs on online resources including books, texts, articles, or other sources such as code.

1. The Status of Literary Works, Software Code and Databases

Most of the training material has the potential to be protected by copyright by nature of being literary works, which are protected under art. 2 of the Berne Convention, to which many EU directives refer. Software code is protected as a computer program which is considered a literary work under art. 1(1) Computer Programs Directive.[24] Some authors, such as Margoni consider the protection of databases relevant, as some AIs train on data accessible via databases. Indeed, even Copilot takes code from GitHub, a code repository. The status of databases is regulated by art. 1 Database Directive.[25] Nevertheless, the protection extends to the organisation of the data within the database, not the data itself.[26]

2. The Requirements for Protection

Under EU law and indeed in most jurisdictions, a work must be sufficiently original to warrant copyright protection. The copyright of literary works is the most pertinent framework relating to ML of NLPs. The following sections focus on it.[27]

---

[20] Ornes, The Unpredictable Abilities Emerging From Large AI Models, https://www.quantamagazine.org/the-unpredictable-abilities-emerging-from-large-ai-models-20230316/, accessed 27 April 2023.
[21] Sutrop, *Acta Balt. Hist. et Philos. Sci*. 2020, pp.55-69.
[22] Grimmelmann, Yale Law Rev. 2005, pp. 1719-59.
[23] Directive 2009/24/EC art. 1.
[24] Directive 2009/24/EC art. 1.
[25] Directive 96/9/EC.
[26] Margoni, CREATe Working Paper, 2018, p.8.
[27] Stamatoudi (2000).

a) The Originality of Literary Works in International Law

The concept of originality is not precisely defined by The Berne Convention, which establishes international copyright standards.[28] The Berne Convention refers to "intellectual creations", which are required in art. 2(5) for the protection of literary and artistic works[29] as well as in all protected works in art. 2. As such, creations such as books, songs, photographs, paintings, drawings or sketches are all subject to this requirement. This view is supported by cases such as *Infopaq I*.[30] The level of originality of an intellectual creation has to be determined on a national level.[31] In any case, however, software code is protected as a literary work in most jurisdictions.

b) The Originality of Literary Works in EU Law

In the 1990s, the EU unified the level of originality required to be considered the author's own intellectual creation in the Computer Programs Directive.[32] In the 2010s, the EU clarified that these requirements applied horizontally, as well as vertically, meaning that the author's own intellectual creation extended to all matters in the Berne Convention and the Copyright and Information Society ("InfoSoc") Directive.[33] The EUCJ also clarified that originality occurs where free, creative decisions and personal marks are made on a work, but not when there is only one way to express an idea or some restrictions prevent the ability to make creative choices.[34]

The EUCJ also emphasised that the author's own intellectual creation is a restrictive requirement. For example, the court did not allow protection for fixtures at a sports game as there are insufficient creative choices.[35] Yet, copyright protection could extend to an eleven-words long extract (as was the case in *Infopaq*),[36] a photograph,[37] a user interface or to a programming language.[38] It appears that the court considers the qualitative nature of the works rather than the form. Even a sentence could be protected if deemed sufficiently original, which is a contrast to the UK jurisdiction.[39]

In 2019, the EU adopted the Directive on Copyright in the Digital Single Market which supplements previous directives and aims to harmonise union law applicable to copyright.

Software code (both source and object code) and other data used to train natural language models are therefore eligible for copyright protection in the EU. The sole requirements for protection are originality and the author's own unique creation, requiring intellectual contribution and personal effort (as opposed to merely technique or skill). The Berne Convention further specified that it is a matter of national legislation to regulate that the work

[28] Ricketson, OUP 2005, 8.05.
[29] Berne Convention art. 2(5).
[30] C-5/08 Infopaq International A/S v. Danske Dagblades Forening [2009] E.C.R. I-06569.
[31] Ginsburg, Columbia Law Rev. 1992, 92:338–388.
[32] Directive 2009/24/EC.
[33] Margoni, CREATe Working Paper, 2018, p. 5.
[34] C-604/10 Football Dataco v. Yahoo! [2012], ECLI:EU:C:2012:115, at [39].
[35] Margoni, CREATe Working Paper, 2018, p. 5.
[36] C-5/08 Infopaq International A/S v. Danske Dagblades Forening [2009] E.C.R. I-06569.
[37] C-145/10 Eva-Maria Painer v. Standard Verlags GmbH [2011] ECLI:EU:C:2011:798.
[38] C-406/10 SAS Institute v World Programming [2012] ECLI:EU:C:2012:259.
[39] Margoni, CREATe Working Paper, 2018, p.7.

will receive copyright protection as long as it is in a fixed medium.[40] The protection grants an economic as well as moral right to the work, enabling the owner to, *inter alia*, control the rights of distribution, sale, access and copying.[41]

II. Economic Exploitation Rights Relating to ML

The copyright frameworks establish several economic exploitation rights, notably the author's exclusive right of reproduction and adaptation (and the right to consent thereto).[42] This essay focuses on these two rights as they most closely relate to the possible rights that could be infringed by an AI training on publicly available code.

1. Right of Adaptation

The right of adaptation refers to the right to make derivative works, which means the right to change, modify or translate the original. [43]

At the international level, this is dealt with in art. 2(1) of the Berne Convention which states that some derivative works deserve autonomous protection despite being based on another copyrighted work.[44] The works must still pass the intellectual creation criterion as well as the originality thresholds.[45] Such protection even extends to works that were created without authorisation for its use (as opposed to the U.S. where this provision does not apply).[46] However, to become a secondary work, the elements that constituted an intellectual creation in the primary work must be adapted or transformed into the secondary work. It is not possible to claim that being inspired by a work makes a work a derivative. In the cases of "mere" inspiration, the work qualifies for protection as a primary work.[47] The Berne Convention recognises three types of modifications – translations, arrangements of music and adaptations or other alterations.[48] Adaptations or other alterations can encompass anything else that is not a translation or arrangement or music, such as transforming a literary work into a dramatic work.[49] However, the category does not encompass works that did not have significant new contributions. Small edits and changes do not warrant this protection.[50]

At the EU-level, the right of adaptations has not been harmonised except for in the Computer Program and Database Directives, which regulate adaptations as well as translations explicitly.[51] These provisions, together with EUCJ case law, are relevant with regards to the protection of code from being used in machine learning.[52]

---

[40] Berne Convention art. 2(2).
[41] Directive 2009/24/EC.
[42] *Ibid.* art. 4(1)(b).
[43] Margoni, CREATe Working Paper, 2018, p.12.
[44] Berne Convention art. 2.
[45] *Ibid*.
[46] US Copyright Act 1976 s.103(a).
[47] Margoni, CREATe Working Paper, 2018, p.9.
[48] Goldstein, 8.81.
[49] Masouyé, WIPO 1978, 76 (76).
[50] Goldstein, 8.81.
[51] Directive 2009/24/EC art. 4(1)(b); Directive 96/9/EC art. 5(b).
[52] Margoni, CREATe Working Paper, 2018, p.8-12.

The Computer Program Directive affirms that unauthorised translations, adaptations, or transformations of software violate copyright unless they are necessary for interoperability.[53] This protection extends to preparatory designs, source code and object code alike. Walter & von Lewinski argue that this is because all subsequent steps can be seen as adaptations of the primary versions, which are protected.[54] By nature of art. 4(b), the author is granted the exclusive right to translate, adapt, arrange and alter the programme, unless exceptions apply – in cases where it is necessary for the intended use, to study the program (art. 5) and for decompilation (art. 6).[55]

The Database Directive affirms that the database is protected to grant the author the exclusive right to translation, adaptation or arranging and alteration.[56] This relates to the database itself (the organisation and structure) but not to the data. Some have criticised that the idea of translating a database structure is conceptually flawed.[57]

Databases and software are thus two exceptions to the general position of the EU leaving the copyright protection of modifications to a work to the member states' discretion. Eechoud et al. propose that this is because harmonising modification rights without also harmonising the definition of originality could have had unpredictable outcomes.[58]

2. Right of Reproduction

Under EU law, the right of reproduction is protected as an exclusive right of the copyright holder by art. 2 of the InfoSoc Directive. It is defined as "any direct or indirect, temporary or permanent reproduction by any means and in any form, in whole or in part".[59] This also encompasses temporary reproductions such as those made in the RAM of a computer, but also when visualising websites.[60] The machine learning of an AI on vast amounts of data could therefore fall within the scope of copyright protection.

a) Art. 5(1) Exception

However, article 5(1) provides an exception to allow temporary acts or reproductions that *"are transient or incidental [and] an integral and essential part of a technological process whose sole purpose is to enable a transmission […], or a lawful use of a work or other subject-matter to be made, and which have no independent economic significance".*[61]

There are four things to note about this. Firstly, the court clarified that a human initiating or terminating the process does not interfere with the condition of a reproduction being an integral and essential part of a technological process being satisfied.[62] This is important because AI training is often initiated and terminated by a person. Secondly, the acts of temporary reproduction must pursue the goal of enabling the work to be used lawfully. Thirdly, the temporary reproduction cannot lead to a supplementary generation of profit over and above of

[53] Directive 2009/24/EC rec.15
[54] Walter, 5.1.39.
[55] Directive 2009/24/EC art. 4.
[56] Directive 96/9/EC art. 5(b).
[57] Walter, 9.5.9.
[58] Eechoud, Kluwer Law Int. 2011.
[59] Directive 2001/29/EC art. 2.
[60] *Ibid.*
[61] Directive 2001/29/EC art. 5.
[62] C-5/08 Infopaq International A/S v. Danske Dagblades Forening [2009] E.C.R. I-06569 at [29].

what is earned through the lawful use.[63] Fourthly, EUCJ case law clarified that temporary reproductions in a data capture process can also qualify for the exception of permitted temporary copies under art. 5(1). This means that, if the training of an AI is similar to the data capture process, the AI could qualify for this exception, meaning that no copyright is violated under EU law.

b) Data Capture in EUCJ Caselaw - *Infopaq I & II*

The *Infopaq* cases were crucial cornerstone decisions regarding data capture processes and copyright in the EU. Infopaq was a media monitoring business that compiled, extracted, indexed and printed newspaper articles and keywords from Danish newspapers and made them available through a data capture process. That activity was done for commercial purposes without authorisation from the newspapers.

The data capture process in *Infopaq* followed five steps and was integral to the copyright analysis:

1) The newspapers are registered manually on an electronic registration database
2) Sections of the newspapers are scanned, creating a special file for each page and uploaded to a server called the Optical Character Recognition server.
3) The server transforms the data into digitally processable data, as a part of which each letter is translated into a character code so the computer can recognise it. After this, the special file is deleted.
4) The text file is processed to give search words defined beforehand, capturing additional five words before and after the search word to enable the user to find better results.
5) The system prints a cover sheet with all the pages where the searched word was found.

By way of example from the case, a final print could take the following form:

‘4 November 2005 – *Dagbladet Arbejderen*, page 3:
TDC: 73% “a forthcoming sale of the telecommunications group TDC which is expected to be bought”’.[64]

The question was whether the data capture process qualified for the exception under art. 5(1) of the InfoSoc Directive. The court held that processes one to four did, but the fifth stage posed difficulties. The permanent nature of printing was considered to be a reproduction, and as such not covered by art. 5(1). This was held despite the printing of merely eleven words. *Infopaq* was significant, but it did not create a new test for the reproduction of copyrighted works. Indeed, academics emphasise that prints even shorter than 11 words could meet the condition of expressing the author’s intellectual creation, and even longer extracts need not meet it.[65] As such, the overall nature of the machine learning process needs to be assessed to evaluate the copyright position. This discussion will ensue in the later sections of this thesis.

[63] *Ibid.*; C-302/10 Infopaq II ECLI:EU:C:2012:16 [2012].
[64] C-5/08 Infopaq International A/S v. Danske Dagblades Forening [2009] E.C.R. I-06569.
[65] Margoni, CREATe Working Paper, 2018, p.19.

III. Text and Data Mining Exception

The machine learning of an AI could also fall under the Text and data mining ("TDM") exception, which would negate any copyright violation liabilities. This essay therefore turns to give an overview of the TDM provisions.

TDM refers to copying vast amounts of data to extract and analyse the patterns found therein. It is also a term often used in connection with AIs and LLMs such as Copilot, which use vast amounts of data, extract patterns and analyse the results.[66] TDM is defined in the Directive on Copyright in the Digital Single Market ("DCDSM") as "any automated analytical technique aimed at analysing text and data in digital form in order to generate information which includes but is not limited to patterns, trends and correlations".[67]

TDM is an exception to copyright liability and is a notable doctrine in EU law. Its effect is comparable to the US fair use doctrine in the sense that it excuses a copyright violation where there is an acceptable reason to use the data. In the EU, it is narrower, however, as it prescribes situations where the use is acceptable, namely in non-commercial instances. The US approach evaluates the situation more holistically. The EU exception is anchored in articles 3 and 4 of the DCDSM.[68]

- Article 3(1) allows "reproductions and extractions made by research organisations and cultural heritage institutions in order to carry out, for the purposes of scientific research, text and data mining of works or other subject matter to which they have lawful access."[69]
- Article 4(1) allows "reproductions and extractions of lawfully accessible works and other subject matter for the purposes of text and data mining," as long as "the use has not been expressly reserved by their rightsholders in an appropriate manner, such as machine-readable means in the case of content made publicly available online" (Art. 4(3)).[70]

In other words, the article 3 exception is applicable in cases of data mining for scientific or academic (non-commercial) purposes, while article 4 permits data mining for any purpose but includes a right to opt out contractually.

These provisions for TDM exceptions have been criticised by some academics such as Margoni and Kretschmer. Margoni views the regulation of a major developing area of TDM through a mere exception to a rule as inadequate.[71] Moreover, businesses having to obtain permission under art. 3 is seen as too burdensome by Kretschmer, who believes that the opt-out (rather than opt-in) mechanism in art. 4 only partly but not sufficiently "recalibrates" art. 3.[72] Others argue that this causes an unreasonable disadvantage to the EU AI sector, not only because of the costs associated with having to negotiate licensing over vast amounts of data.[73] The quality and type of data for AI training will deteriorate as companies would train on public, older,

---

[66] HM IPO, Artificial Intelligence and Intellectual Property, https://www.gov.uk/government/consultations/artificial-intelligence-and-ip-copyright-and-patents/outcome/artificial-intelligence-and-intellectual-property-copyright-and-patents-government-response-to-consultation, accessed 2 May 2023.
[67] *Ibid.* art. 2.
[68] Directive 2019/790 art. 3 and 4.
[69] Directive 2019/790 art. 3(1).
[70] *Ibid.* 2019/790 art. 4(3).
[71] Margoni, GRUR International 2022, pp. 685–701.
[72] *Ibid*.
[73] Senftleben, JIPITEC 2022, p.5.

biased or less accurate data, leading to poorer AI quality development, potentially promoting algorithmic discrimination.[74] Kretschmer further argues that the EU is herewith protecting mere facts and data, which do not pass the (arguably low) EU originality threshold. Importantly, the act fails to clarify how to opt in after having opted out.[75]

E. The Licensing of Software

The author of the copyrighted work can enter into legal contracts to allow the work to be used or distributed.[76] These contracts are licenses, which can be either exclusive or non-exclusive.[77] The nature of licenses used is significant in the context of NLP and ML copyright analysis, as not complying with a valid license results in a breach of copyright.

A license is a legal agreement between the owner of the copyrighted work and a second party that seeks to use the copyrighted work.[78] As software and openly available code online became more prevalent, two licensing models emerged to protect source code: open or free software licensing and closed, proprietary licensing.[79]

I. FOSS Licensing

The Free Open Source Software Licensing is premised on the principle that code is a non-literary form of composition and was intended for sharing, not least because it is argued that private software produces bad software. As such, source code should be openly available for anyone in the public to use, share, modify or distribute.[80] Sassi and Nesrine define an open source license as a set of legal terms that express "1) permissions which are the actions allowed on the software granted by the licensor to the licensee, and 2) obligations which are the conditions that have to be fulfilled when undertaking one of more authorised actions".[81]

The key features of all FOSS licenses are that they are free and accessible to anyone. However, the price is not intended to be the main part of the appeal of these licenses.[82] Instead, Peterson, who coined the term "open source" states that open source software focuses on the accessibility and quality of software.[83]

There are two prominent organisations and strands of thought that have developed almost parallel to each other, both in support of the notion that software licenses should not be exclusive: the Free Software Foundation and the Open Source Initiative. The following section gives a brief overview of the historical development of the movements to understand the ideological differences.

---

[74] Levendowski, Wash. L. Rev 2018, p. 592.
[75] Margoni, GRUR International 2022, p. 685–701.
[76] Directive 2009/24/EC art. 4.
[77] Sassi, RIADI Laboratory 2022, p. 2.
[78] *Ibid.*
[79] *Ibid*.
[80] Moglen, First Monday 1999, vol. 4(8).
[81] Sassi, RIADI Laboratory 2022, p. 2.
[82] Peterson, How I coined the term 'open source', https://opensource.com/article/18/2/coining-term-open-source-software, accessed 12 March 2023.
[83] *Ibid.*

1. The History of FOSS

The beginning of the software industry can arguably be traced to IBM first distinguishing between the hardware and software of a computer in 1969.[84] The natural form which software took was proprietary software, as companies entered the industry for profit.

The cornerstone of the FOSS movement began in 1972, when two scientists Ritchie and Thompson at AT&T Bell Labs developed the first operating system, Unix, distributed with a license that allowed academic use to universities, where the code was distributed and improved.

The next stepping stone occurred in 1983 with activist Richard Stallman, who was inspired by Unix and wanted to create only free software, announcing the GNU project. With this, Stallman established the main tenets of free software, which he further consolidated by founding the Free Software Foundation ("FSF") in 1985.[85]

The establishment of the Open Source Initiative can arguably be perceived as a reaction to the FSF. There was a negative perception by some that the term "free software", which implies a culture of giving away, was anti-business. To many, it also implied a worse quality.[86] The open-source movement sought to make software more attractive to businesses and to spread to a wider community of users.[87] The price-focused label free software was according to Peterson too "distracting".[88] The Open Source Initiative was founded in 1998 by Raymond, Perens and O'Reilly which consolidated the term open source software. This community also developed new licenses which were also useable by businesses, such as permissive licenses like the Mozilla Public License (MPL).[89]

To this day, some disagreements persist between the free software and the open-source communities. For instance, Stallman emphasised on numerous occasions his fear that the open-source community, by focusing solely on the open-source nature, would push software freedom into the background. He also accused the movement of neglecting software freedom to gain greater acceptance in the software business which runs on profit, the antithesis of what software freedom is about to Stallman.[90] As such, he has intentionally tried to distance himself from the open-source community. The open-source community, by contrast, tends to view Stallman as too radical.[91]

a) The Principles of FSF

The definition of the Free Software Foundation highlights the freedoms that are anchored in the non-exclusive licenses. According to FSF, free software is about the users' freedom to run,

[84] De Laat, Research Policy 2005, 1511-1532.
[85] Tai Li, The History of the GPL, https://www.free-soft.org/gpl_history/, accessed 1 May 2023.
[86] Gonzalez-Barahona, p.75-79.
[87] Peterson, How I coined the term 'open source', https://opensource.com/article/18/2/coining-term-open-source-software, accessed 12 March 2023.
[88] *Ibid.*
[89] Gonzalez-Barahona, p.75-79.
[90] Stallman, Why Open Source Misses the Point of Free Software, https://www.gnu.org/philosophy/open-source-misses-the-point.en.html, accessed 17 March 2023.
[91] Krempl, Die Stimmen der Revolutionäre, https://www.telepolis.de/features/Die-Stimmen-der-Revolutionaere-3495044.html, accessed 1 May 2023.

copy, distribute, study, change and improve software.[92] As such, the FSF lists four freedoms, the cumulative presence of which is an indicator that software is truly free.[93]

The freedoms include the freedom to use the program for any purpose, to examine and change the code, to distribute it and to improve the program and distribute these improvements.[94]

b) The Principles of OSI

The Open Source Initiative sets out a list of 10 criteria for open source software, the full list of which is appended to this thesis. The general premise is the free redistribution of the program and the source code. Works based on the program must permit further distribution under the same licenses. It further imposes some requirements on non-discrimination and tech-neutrality.[95]

There are a few philosophical differences between the two movements. The open-source software movement takes a more pragmatic and therefore also more commercial (according to Stallman, profit-oriented) approach. The FSF, by contrast, puts a strong emphasis on the four freedoms and does not elaborate on the details of distribution as OSI does. Despite these differences, the two movements are in essence perceived as synonymous because they stand in opposition to proprietary and exclusive licenses. One comes across the term Free and Open Source Software (FOSS) relatively often in legal and computer science literature.

2. FOSS License Types

As a response to FOSS, various license types have emerged that enshrine various legal duties and rights for the licensor and licensee.[96] These FOSS licenses can be divided according to their level of restrictiveness. One can distinguish between the so-called permissive licenses and copyleft licenses.

a) Copyleft Licenses

Copyleft licenses are more restrictive than permissive licenses because they allow the use, distribution, re-working and generally all other processes that permissive licenses allow, but in addition, they require that the product made with the open-source code inherits the license terms of the original license.[97] In essence, code produced with copylefted code must remain copyleft and cannot be made into a proprietary license.[98]

The extent of modification that a license under copyleft allows varies. The "strict copyleft" licenses, such as the GPL license, consider any edits, however, minor, subject to copyleft. Other copyleft licenses, also called "limited copyleft", allow the original software to be combined with extensions. An example of such a license is the Mozilla Public License (MPL).[99]

---

[92] Philosophy of the GNU Project, https://www.gnu.org/philosophy/free-sw#n1, accessed 2 May 2023.
[93] *Ibid.*
[94] *Ibid.*
[95] OSD, https://opensource.org/osd/, accessed 9 March 2023.
[96] Lasota, What is a license, https://www.sfscon.it/talks/what-is-a-license/, accessed 7 March 2023.
[97] Lin, Journal of Information Science and Engineering 2006. p. 3.
[98] *Ibid.*
[99] Jaeger, p. 73.

The most popular copyleft licenses by restrictiveness are the GNU Affero General Public License (AGPL), The GNU General Public License (GPL), The Lesser General Public License (LGPL), Eclipse Public License (EPL) and The Mozilla Public License (MPL).[100]

- The GPL license allows the user to run the code for any purpose (including commercial, private and patent use). The software that uses GPL code must be distributed under the GPL.[101]
- The AGPL license is the same as the GPL except it requires that when the code is used over a network, the source code must still be included. This seeks to address a loophole with the GPL where a user does not technically distribute software when it is only shared over a network.[102]
- The LGPL is a license with the same terms as the AGPL and the GPL, except for the situation of smaller projects being accessed through larger licensed works, where there is no requirement of distribution of the larger project. Further, the smaller project does not need to be distributed under the same terms as the large project.[103]
- The EPL is a license suitable for business software as it enables as code to be combined and sub-licensed (irrespective of whether the code is EPL, non-EPL or even proprietary), provided that the non-EPL code elements are independent and separate. The EPL permits modifications, but they must be under the same terms.[104]
- The MPL is the least restrictive copyleft license as it enables modification and the use of code in proprietary software, only requiring any code under the MPL to be kept in separate files and these to be distributed with the code. The license includes patent grants and requires the retention of copyright notices.[105]

Some licenses are better suited for commercial businesses, some are more restrictive. However, they all aim at protecting the freedom of software.

b) Permissive Licenses

By contrast, permissive licenses, also called non-reciprocal or non-copyleft licenses, are less restrictive than copyleft. As such, the software licensee can use the software for free, modify and then turn it into proprietary software with a different license than that of copyleft.

Examples of these licenses are the Apache license, the Berkeley Software Distribution (BSD) license, or the MIT license.

- The Apache licenses enshrine the principle that the more users contribute to the development of the software, the more entitled they are to keep developing it further. The Apache grants an unlimited non-exclusive right to use, copy and distribute the software and distribute the modified versions thereof as long as each modified work is provided with the license text and has a document showing the modification.[106]
- The MIT licenses allow the reuse of software for both open- and closed-source software. The sole condition of this license is that the new distributed source code

[100] De Laat, Research Policy 2005, 1511-1532, p. 56.
[101] *Ibid.* p. 19.
[102] Lasota, Free Software Licensing, https://download.fsfe.org/presentations/20221128-free-software-licensing-bmbf-lucas-lasota.pdf, accessed 6 March 2023.
[103] De Laat, Research Policy 2005, 1511-1532, p. 21.
[104] Nabi, International Journal of Advanced Research 2015, 677 – 686.
[105] De Laat, Research Policy 2005, 1511-1532, p. 43.
[106] Jaeger p. 89.

contains a notice about the original copyright and license. This makes the license highly compatible and widely used.[107]

- The BSD licenses are a group of licenses that generally require license notices of copyright but allow larger and/or licensed works to be distributed under a different license and without any source code. There are three main types – the original BSD (four-clause license), BSD 2.0 (three-clause license), the simplified BSD and the Zero Clause BSD. The 2-clause license is similar to MIT while 3 and 4-claused licenses impose higher restrictions on reuse.[108]

II. Closed Proprietary Licensing

Closed proprietary licensing is premised on the principle that source code deserves copyright protection and sole authorship.[109] Code is proprietary and restricted from public access.[110] A general closed software license does not permit access to source code, distribution, or any derivative works. While software that is licensed under a proprietary license may contain parts of open source software (typically under the more permissive licenses like MIT), proprietary software cannot be included in any open-source software.[111] The user agreement of proprietary software therefore frequently contains clauses that forbid decompiling or changing of the source code, if accessible.[112] The key component of making software proprietary is reserving the right to place legal restrictions on modifications and use. Thus, the owner of proprietary software, who is usually its creator, can make the source code available to a restricted group or, for instance, make certain agreements permitting the use of the source code on an individual basis and the software could still be labelled proprietary.

F. Case Study: GitHub's and OpenAI's Copilot

I. Introduction to GitHub's Copilot

GitHub's Copilot is an AI-powered programme launched in 2021 that seeks to support code writing by providing auto-completion prompts.[113] The programme is a closed subscription model with clear commercial goals. On its website, Copilot describes itself as a programme that helps increase the efficiency of software writers and improve the quality of code.[114] It can debug code, identifying contextual patterns and automatically suggesting functions. It can generate entire blocks from comments.[115]

GitHub is an online platform for software development, used for storing and collaborating on software development projects.[116] In addition, GitHub's "Git" is a programme that allows open-source version control, so programmers can see the revision history of code.[117] GitHub is also one of the largest repositories (storage spaces for software code). As of January 2023,

---

[107] Jaeger p. 92.
[108] Sinclair, The Journal of Open Law, Technology & Society, 2010.
[109] Choksi, Conference on Human Factors in Computing Systems 2023, p. 1.
[110] Weber (2005).
[111] *Ibid.*
[112] Weber (2005).
[113] Introducing GitHub Copilot: your AI pair programmer, https://github.blog/2021-06-29-introducing-github-copilot-ai-pair-programmer/, accessed 2 Feb 2023.
[114] *Ibid.*
[115] Barke, 5.2.5.
[116] GitHub, https://github.com/, accessed 9 February 2023.
[117] *Ibid.*

GitHub reported having over 100 million developers[118] and more than 372 million repositories,[119] including at least 28 million public repositories.[120] It is the largest source code host as of November 2021.[121]

GitHub repositories can be public and private, depending on what the creator of the code chooses.[122] By default, if published on GitHub in a public repository, the code is accessible and other users can view and fork the code.[123] The creator of the code and its repository can also choose under what license the code is viewable and useable.[124]

The way the code suggestion system works is significant from a copyright perspective. For example, if a user types in the JavaScript name of a function that tests whether a number is a prime number (function isPrime(n) {), Copilot produces the following sequence of code (see Fig.1).

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

Fig.1 Example output by Copilot

The loop that checks possible divisors need not have checked for every divisor smaller than *n,* but rather only check for divisors smaller than the square root of *n*. The fact that the code produced is not the optimised way the code could be, (namely that it is much slower than it needs to be), shows that Copilot has no understanding of coding itself, but rather, it has been trained on data to extract patterns and produce output.[125]

II. The Training of Copilot

The training of Copilot refers to the machine training stage of creating an AI that has been described above in section C.II.2.

1. The Methodology

GitHub's Copilot is a large language model that uses GPT-3 (the third version of the Generative Pre-trained Transformer).[126] This model is a type of deep neural network that is trained by unsupervised learning, or being tasked with predicting the next word in a sequence of words and initially not being given the correct corresponding value.[127] The model is subsequently trained using supervised learning as a fine-tuning method to teach specific tasks.[128] This fine-

[118] Dohmke, 100 million developers and counting, https://github.blog/2023-01-25-100-million-developers-and-counting/, accessed 5 May 2023.
[119] GitHub Number of Repositories, https://github.com/search, accessed 25 January 2023.
[120] Repository search for public repositories, https://github.com/search?q=is:public, *accessed 2 February 2023.*
[121] Finley Klint, The Problem With Putting All the World's Code in GitHub, https://web.archive.org/web/20150629152927/http:/www.wired.com/2015/06/problem-putting-worlds-code-github/, accessed 5 March 2023.
[122] Licensing a repository, GitHub, https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/licensing-a-repository, accessed 2 May 2023.
[123] *Ibid.*
[124] *Ibid.*
[125] Class Action Complaint Case 3:22-cv-06823 p.19.
[126] Howard p. 1
[127] Radford (2019) p. 1.
[128] Radford, Papers With Code 2018, p.1-2.

tuning occurs by a technique called stochastic gradient descent which is used to teach the AI to generate similar output to the input.[129] This is a part of the system of techniques that are used to ensure that the AI creates new code, rather than reiterating learned and memorised code inputs.[130] It can, however, happen that Copilot reiterates the code from the training stage verbatim, as some degree of memorisation is inevitable in ML.[131]

2. Copilot's Training Set

Copilot is "trained on natural language text and source code from publicly available sources, including code in public repositories on GitHub."[132] Copilot's AI takes in vast amounts of incalculable code as data on the platform that has been released under various FOSS licenses.[133] Copilot is also trained on other public sources of code, (generally publicly available websites or repositories with code), as well as large bodies of natural language. The goal of this is to enable Copilot to learn human language and be able to respond more effectively to coding prompts.[134] As such, the AI learns the statistical probabilities within natural language, finding patterns. Repeating the learning of these patterns increases the probability of stringing together meaningful sentences. Copilot has been trained on Wikipedia articles, books and research papers, technical documentation such as developer guides, online forums and Q&A websites to gain a good command of natural language.[135] The main alleged copyright violation occurs with regard to the training on GitHub's public source repositories under various FOSS licenses, on which this thesis focuses.

3. License Types

It is important to emphasise that the data on which Copilot has been trained falls under various licenses, including permissive licenses and copyleft licenses.[136]

In order to create a new repository on GitHub, a user can select one of thirteen default license types for the code.[137] The user is, in theory, free to choose a different license later or indeed no license at all, making the content closed-proprietary.[138] There are also two licenses which waive all copyright and related rights and donate the covered work to the public domain.[139] These two licenses are excluded from the lawsuit and also from the ensuing analysis because arguably no copyright violation can be committed against them.

The remaining eleven licenses are as follows:

(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");

---

[129] Rothchild, Free Software Foundation 2022, p. 2.
[130] *Ibid.*
[131] *Ibid.*
[132] Your AI Programmer, https://github.com/features/copilot/, accessed 2 February 2023.
[133] Choksi, Conference on Human Factors in Computing Systems 2023, p. 2-3
[134] Your AI Programmer, https://github.com/features/copilot/, accessed 2 February 2023.
[135] Alford, OpenAI Announces 12 Billion Parameter Code-Generation AI Codex, https://www.infoq.com/news/2021/08/openai-codex/, accessed 6 May 2023.
[136] Class Action Complaint 3:22-cv-06823 p.8.
[137] *Ibid.*
[138] *Ibid.*
[139] Class Action Complaint 3:22-cv-06823 p.8.

(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");
(11) Mozilla Public License 2.0 ("MPL-2.0").[140]

The significance of naming these licenses is that all eleven of these contain the following three requirements for the use, modification and redistribution of the copyrighted work:
- attribution to the owner of the licensed materials,
- inclusion of a copyright notice, and
- inclusion of the applicable suggested license texts.[141]

The lawsuit alleges that Copilot, by using code with these licenses in its training, but not complying with any one of these three requirements, violates copyright.

III. The Lawsuit

It has become very contentious whether the FOSS-licensed code on which Copilot was trained can be lawfully used in the manner used by Copilot without proper attribution.[142]

Thus, a group of programmers and solicitors represented by Joseph Saveri Law Firm LLP decided to sue Copilot in a class action together with suing Codex (a general-purpose programming model developed by OpenAI).[143] The contents of the lawsuit state six class allegations which contain a totality of eleven grounds for the lawsuit. The focus of this paper focuses on two out of the six main class allegations, namely copyright violations and contractual (licensing) violations because these relate to copyright violations.[144] The lawsuit does, however, also include a claim of fraud for promising to not sell and distribute licensed materials in GitHub's Privacy Statement and Terms and Conditions and a claim for Privacy Violations, Unlawful Competition as well as Conspiracy (given the cooperation of GitHub, OpenAI and Copilot).[145]

IV. General Liability

Firstly, Copilot argues that that the use of GitHub's code repositories to train the AI falls within GitHub's Terms of Service to which users of GitHub consent.[146] Secondly, GitHub alleges that the output is not a mere copy of the source code, and thus it is not a violation of copyright. Thirdly, GitHub argued that even if it were considered a copy, Copilot's use falls within the exception of fair use, which is a defence doctrine to copyright violation claims under 17 U.S.C. § 107.[147]

---

[140] *Ibid.*
[141] *Ibid.*
[142] Class Action Complaint 3:22-cv-06823 p. 17, para 64. line 23.
[143] *Ibid.*
[144] *Ibid.* p. 1.
[145] *Ibid.* p. 10.
[146] Rothchild, Free Software Foundation 2022, p. 3.
[147] USPTO, Comment Regarding Request for Comments on Intellectual Property Protection for Artificial Intelligence Innovation, p. 5

1. Alignment with GitHub's Terms of Service

Rothchild argued that GitHub's terms of service might encompass the use of code for machine learning.[148]

Upon registering with GitHub and creating a repository, users are required to accept and agree to GitHub's Terms of Service. The terms include a provision reserving GitHub's right to "store, archive, parse and display Your Content, and make incidental copies, as necessary to provide the Service".[149] The terms also define "Service" as any service provided by GitHub, which includes Copilot.[150] Further, GitHub reserved the right to copy code "to our database and make backups, show it to you and other users, and parse it into a search index or otherwise analyse it on our servers."[151]

It is questionable whether the training of Copilot can fall within the above conditions.[152] Firstly, the training of Copilot involves copying the code from the repository into a computer's RAM. Copying and showing of the code is allowed as per the terms of service. Interpreting the wording literally, therefore, some academics such as Rothchild conclude that this encompasses the extent of copying which was required for Copilot.[153]

The court might take a different stance, however. The terms of service do not refer to the extent of copying and usage that is done to train the AI. The wording refers to making "incidental copies as necessary to provide the Service". Using the word incidental suggests happening by chance and therefore that the occurrence will be limited. This is further supported by referring to necessity, which also implies minimal usage. Before the development of Copilot, the terms could have been understood to mean analysing the code to promote the basic functioning of GitHub for searching code snippets within the database. Likewise, showing the code – within the terms of service – most likely referred to displaying the repository to users on an individual basis, not a schematic attempt to read, scan and process all of the code to be shown in an application such as Copilot.

The court might find the terms of service ambiguous and tend towards finding Copilot in violation of the terms. This is admitted as a possibility by academics who suggest that the "license fails to unambiguously convey the intent of the parties".[154] If the courts do find this as an ambiguity, they might respond in one of three ways: either interpret the license narrowly, concluding that Copilot's use falls outside of this meaning, or broadly, by concluding that Copilot can be reasonably considered to be a new use within the terms of the service.

Thirdly, the court could find that Copilot generates derivative work from the source code and that the Terms of Service do not expressly allow for this use. However, as reiterated by Rothchild, there is little case law on derivative works.[155]

---

[148] Rothchild, Free Software Foundation 2022, p. 2.
[149] *Ibid*.
[150] GitHub Terms of Service, https://docs.github.com/en/site-policy/github-terms/github-terms-of-service#a-definitions, accessed 13 May 2023.
[151] *Ibid.* section D.4.
[152] Rothchild, Free Software Foundation 2022, p. 4.
[153] *Ibid.*
[154] Rothchild, Free Software Foundation 2022, p. 4.
[155] *Ibid.*

2. Extent and Probability of a Copy Occurring

GitHub concedes that in normal use, Copilot can reproduce the copied input 1% of the time, and this applies to code snippets longer than 150 characters.[156] There are several points to note about this. It has been argued by some academics, such as Rothchild, that 1% is so minimal that it does not constitute a copy.[157] However, the 1% refers to the probability of a copy occurring. The definition of a copy does not regard the probability of its occurrence. Verbatim copies of more than 150 characters occur 0.1% of the time, according to GitHub.[158] It is to be further noted that even changing a single bracket within ten lines of code is sufficient for the code to not be considered verbatim.

Secondly, this statistic is based on GitHub's own internal research and there is a strong incentive for GitHub to conceal or minimise the extent to which a copy can occur. Thirdly, the boundary of 150 characters is set by GitHub and is artificial – the boundary could well be set lower which would then produce a higher statistic for the probability of Copilot generating a copied work. Arguably, it is a generous boundary, considering that the industry standard limit for maximum line code length is 80 characters.[159]

Even when applying GitHub's conservative metric and generous boundary, we must consider that by June 2022, Copilot had 1,200,000 users and copied output has been produced 12,000 times. By way of illustration, each of these reproductions violates the US Digital Copyright Markets Act ("DMCA") three times, therefore resulting in 36,000 violations of the DMCA.[160] Each of these violations of section § 1202 of the DMCA incur statutory damages of "not less than 2500 or more than 25000" 17 U.S.C. § 1203(c)(3)(B). In the totality of Copilot's scale, this translates to $90 million or $900 million in statutory damages. Given that this number has been calculated as of June 2022 and with conservative estimates, this number is likely significantly lower than the actual value of Copilot's violations.[161]

When applying GitHub's 0.1% statistic on the copying of more than 150 lines of code verbatim, we arrive at 3600 violations of the DMCA. Each of these violations of section § 1202 of the DMCA incur statutory damages of "not less than 2500 or more than 25000" 17 U.S.C. § 1203(c)(3)(B) meaning that this translates to $9 million or $90 million in statutory damages. Likewise, given that this number has been calculated as of June 2022 and with conservative estimates, this number is still likely significantly lower than the actual value of Copilot's violations.[162]

However, some defenders of Copilot, such as Rothchild argue that it cannot amount to a copyright violation as the AI adds to the input, and a degree of transformation is established.[163] In the case of *Campbell v. Acuff-Rose Music*, the court asked whether some new use or purpose

---

[156] Your AI Programmer, https://github.com/features/copilot/, accessed 2 February 2023.
[157] Rothchild, Free Software Foundation 2022, p. 5.
[158] Romero, GitHub Copilot — A New Generation of AI Programmers, https://towardsdatascience.com/github-copilot-a-new-generation-of-ai-programmers-327e3c7ef3ae, accessed 17 March 2023.
[159] Mark Seemann, The 80/24 rule, https://blog.ploeh.dk/2019/11/04/the-80-24-rule/#:~:text=If%20there%27s%20any%20accepted%20industry,line%20width%2C%20it%27s%2080%20characters, accessed 19 March 2023.
[160] Class Action Complaint 3:22-cv-06823 p. 24.
[161] Class Action Complaint 3:22-cv-06823 p.24.
[162] *Ibid.*
[163] Rothchild, Free Software Foundation 2022, p. 3.

has been added.[164] While there may be some reason to claim that the original code was written to accomplish a different purpose than what the AI uses it for – namely to train the engine, arguably the output (which is the purpose), serves the same goal as the input – to provide software code to developers. The fact that there is evidence of the output being substantially similar or identical to the input disproves this point. Importantly, some licenses, such as the GPL, consider any extent of modification to the copyrighted work, however minor, to be subject to copyleft.[165] Given that the GPL has been proved to be present in the data training set, this establishes a license violation.

For example, the "isPrime" JavaScript function generated by Copilot contrasted with the "isPrime" function in *Think JavaScript* by Matthew X. Curinga et al, is identical[166] (see both functions produced by Copilot (fig.2) and included in the book (fig.3).[167]

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

Fig. 2 (function in book)

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

Fig. 3 (Copilot's output)

Curinga's textbook has been uploaded to GitHub's code repositories under the GNU Free Documentation License, which is a type of public license that allows copying and distribution commercially or non-commercially, as long as the 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."[168] Some defenders of Copilot argue that Copilot does not retain any copies of the inputs. However, the retention of copies is not a requirement for copyright violations.[169] Fitzpatrik argues that "it is readily apparent" that Copilot is producing verbatim or almost verbatim copied code.[170] Yet, Copilot provides no license information with its output code. This ties closely to the issues with attribution, discussed in the following sections.

3. Attribution

There is not enough information published about the specific quantities of licensed code used in the training set, nor the exact occurrence of other licenses other than GPL. However, academics such as Choksi and Kuhn report that GitHub has recorded that during training, the system encountered a copy of the GPL more than 700,000 times,[171] which serves as hard evidence that at GPL-licensed FOSS code appears in the training set.[172] Of course, as discussed above, there is plenty of evidence, including GitHub's own admission, that Copilot has been trained on code licensed under other FOSS licenses as well, such as the Apache, MIT, BSD or BSL. The uncertainty only concerns the frequency of occurrence in the training set.

[164] 510 U.S. 569 (1994).
[165] Jaeger, p. 73.
[166] Curinga, Think Javascript, https://matt.curinga.com/think-js/#solving-problems-with-for-loops, accessed 7 May 2023, at 5.12.
[167] *Ibid.*
[168] *Ibid.* at 12.8.
[169] Class Action Complaint 3:22-cv-06823 p.12 [19-21].
[170] Fitzpatrick, Free Software Foundation 2022, p.1.
[171] Kuhn, Free Software Foundation 2022, p. 2.
[172] *Ibid.*

Nevertheless, this uncertainty is not a factor that influences Copilot's liability for not attributing because it is certain that *at least some* code produced has been copied and that *no code whatsoever* includes the required licenses.[173]

The GPL licenses (as copyleft licenses) found in Copilot's output, require that any code derived from or based on or modifying the GPL-licensed software must be licensed under GPL. If someone distributes any software, the source code must be made available. Importantly, a notice that identifies the original source must be included, as well as the notice that it is licensed under the GPL.[174]

Secondly, for example, the MIT license states that "the copyright notice and this permission notice shall be included in all copies or substantial portions of the Software."[175] Anyone using this code ought to, therefore, include the copyright notice and permission notices. This is significant also because Copilot's users are prevented from seeing these requirements to the licenses and complying with the terms of using free source code.[176] In fact, almost all of the open-source licenses require the attribution of the author, notice of copyright and a copy of the license.[177] All of the licenses listed in the litigation, as shown above, have these requirements.[178]

Copilot does not provide for the code to include any attribution. In fact, Copilot has not been trained to find attribution, copyright notices and licenses as essential.[179] Moreover, Copilot does not provide the source code with its suggestions, violating the terms of the code licensed under GPL licenses, of which there were at least 700,000 sources.[180] Ultimately, Copilot is valuable to its users precisely because it is able to find and reproduce useful licensed material while obscuring any rights associated with the material.[181] As such, Copilot's liability appears to rest in the lack of attribution that has been contractually imposed as an obligation in the licenses, rather than the extent of the similarity of the works.

V. Liability under US Law

1. Copyright and/or License Violation

To establish a claim in copyright infringement, the claimants will have to, amongst other things, prove that the software falls within the subject matter and scope and that the copying was unlawful. While the former has been accepted – source code is protected by copyright,[182] the latter is heavily disputed.

[173] Class Action Complaint 3:22-cv-06823 p. 21.
[174] GPL 3.0 License, https://www.gnu.org/licenses/gpl-3.0.txt, accessed 27 March 2023. See appendix for full text of GPL-3.0
[175] MIT License, https://choosealicense.com/licenses/mit/, accessed 22 March 2023. See appendix for full text of MIT license.
[176] Class Action Complaint 3:22-cv-06823 p. 17.
[177] *Ibid.* p. 17.
[178] *Ibid.* p. 8.
[179] *Ibid.* p. 21.
[180] Kuhn, Free Software Foundation 2022, p. 2.
[181] *Ibid.* p. 17.
[182] Rothchild, Free Software Foundation 2022, p. 1-2.

Copilot will be deemed to violate copyright when following § 501 USC 17, it violated any of the rights of the copyright owner stated in §§ 106 – 122.[183] These rights confer, amongst other things, the right of the copyrighted work's author to authorise and control the distribution and reproduction of the works.[184] Such a right of the author is commonly determined by way of licenses.[185] As such, Copilot will have violated the copyright of the authors if it did not comply with the licenses, which, as illustrated above, has not been complied with, and if Copilot indeed reproduced the copyrighted work.

Every output produced by Copilot is derived from the totality of the material provided to it during training.[186] The extent to which Copilot is capable of producing a verbatim copy of the copyrighted work is unclear. Indeed, the result may vary with each search prompt. However, as illustrated in section F.IV.2., even the probability of a copy occurring to which Copilot admits, is significant. Secondly, the process of taking and saving code in the RAM for the AI to be trained on it can be seen as copying.

2. Fair Use

Under 17 U.S.C. § 107, there is a defence to a copyright infringement if the use of the copyrighted work constitutes "fair use".[187] Paragraph 107 also instructs the courts to consider the following factors when determining the meaning of fair use:

(1) "The purpose and character of the use, including whether such use is of a commercial nature or is for non-profit educational purposes;
(2) The nature of the copyrighted work;
(3) The amount and substantiality of the portion used in relation to the copyrighted work as a whole; and
(4) The effect of the use upon the potential market for or value of the copyrighted work."[188]

OpenAI argues that not allowing AIs to train on public data would cause a heavy burden on the development of AIs using machine learning.[189] Academics such as Howard agree with this notion.[190] However, as Butterick argues, ML training on public data is not always fair use, and the case law has not been settled yet.[191] The following sections discuss the criteria for fair use.

a) The Purpose and Character of Use

The court shall consider "the purpose and character of the use, including whether such use is of a commercial nature or is for non-profit educational purposes." While in the paper for the US Patents and Trademarks Office ("USPTO"), OpenAI focuses on the "transformative

[183] 17 USC § 501.
[184] 17 USC § 106.
[185] St Laurent (2004).
[186] Class Action Complaint 3:22-cv-06823 p. 15, [11].
[187] USPTO, Comment Regarding Request for Comments on Intellectual Property Protection for Artificial Intelligence Innovation, p. 5.
[188] 17 USC § 107.
[189] Howard p. 2.
[190] *Ibid.*
[191] Lemley, Texas Law Review 2021, vol. 99/4.

character" of AI,[192] they downplay the characterisation of whether the purpose is commercial. Authors such as Rothchild seem to accept OpenAI's submission on the transformative character of Copilot.[193] Indeed, in *Campbell v Acuff-Rose Music*, the Supreme Court highlighted that the purpose of this criterion is to distinguish between whether the use merely supersedes the purpose of the original code or whether it transforms it.[194] *Campbell* established that a commercial use of a copyrighted work does not automatically prevent the fair use defence. However, the case concerned a music industry lawsuit over an alleged copyright infringement over a sampled song.[195] The case at hand not only concerns a different type of "transforming" of a different type of work (that is, software code), but also is contextually different, using copyrighted work that has been shared on a basis of common solidarity among the coding community.

Arguably, more weight must be given to the commercial factor, considering the scale of Copilot's commercial endeavours and the for-profit nature of the organisation. Copilot's subscription model charges $19 per month for a business subscription and $10 for a subscription for individuals.[196] With revenue over $200 million in 2022 and over 40 million paying customers in 2023,[197] Copilot's programme is a purely commercial enterprise. This weighs into the company's disfavour in the fair use matrix of factors. Indeed, authors such as Howard support this view by arguing that commercial use is almost inherent in models such as Copilot.[198]

b) The Nature of the Copyrighted Work

Secondly, the nature of the copyrighted work is to be considered. This criterion has played a significant role in many judgments concerning the copyright of code.[199] The paper submitted by OpenAI to the USPTO argues that the nature of the copyrighted work, as a factor, does not play a major role in determining the fair use of copyright.[200] The paper presents the case of *Authors Guild v. Google, Inc*. to support this notion.[201] In the judgment, the court states that "the second factor has rarely played a significant role in the determination of a fair use dispute."[202] However, there are four arguments to dispute this statement.

Firstly, the case does not apply as precedent in all of the jurisdictions where GitHub is active.[203] Secondly, this specific remark had been stated in dictum and has no bearing legal authority. Thirdly, even if it did have authority, this remark had been made with reference to a case of *Harper & Row*, which was drawing on the distinction between factual and fictional works,

[192] USPTO, Comment Regarding Request for Comments on Intellectual Property Protection for Artificial Intelligence Innovation, p. 5.
[193] Rothchild, Free Software Foundation 2022, p.2-6
[194] 510 U.S. 569 (1994) at [579].
[195] *Ibid.*
[196] GitHub Docs, https://docs.github.com/en/billing/managing-billing-for-github-copilot/about-billing-for-github-copilot, accessed 12 February 2023.
[197] Colin, How GitHub hit $200M Revenue with 40M customers in 2023, https://getlatka.com/companies/github, accessed 1 March 2023.
[198] Howard p. 3.
[199] Rothchild, Free Software Foundation 2022, p. 4.
[200] USPTO, Comment Regarding Request for Comments on Intellectual Property Protection for Artificial Intelligence Innovation, p. 5.
[201] *Authors Guild v. Google* 804 F.3d at [220].
[202] *Ibid.*
[203] Kuhn, Free Software Foundation 2022, p. 4.

stating that an original being a factual (and not fictional) work should not imply that others may freely copy it.[204] The facts of the case at hand do not fit the cited judgment.

As Kuhn emphasises, *Authors Gild* considers "the search, not authorship of new/derived works".[205] Google copied entire books to allow users to run search results and see "snippets" of the search results. The court held that such copying was fair use because Google added value to what a user could obtain from the copies themselves and importantly, that Google's product did not constitute a substitute in the market for the original works.[206]

Some authors seek to draw an analogy to the fact pattern of GitHub, such as Howard.[207] One could argue that GitHub's Copilot could be seen as a search tool in finding code from public source code repositories. However, to argue that is a conceptual stretch, not least because as per GitHub, the code produced by Copilot is not supposed to be an identical copy of code found in the repositories. Indeed, Kuhn agrees that "the actual Copilot fact pattern is not this one".[208] Importantly, as the Google judgment emphasises, this factor is to be gauged on a case-by-case basis.[209] In this case, it is submitted that the nature of the copyrighted work is extremely significant and plays to the detriment of Copilot.

Howard, for example, interprets the "nature of the copyrighted work" from a technical perspective. He argues that given that the copyrighted work on which the AI is trained is source code, and Copilot eventually produces code that is still source code, the nature of the copyrighted work has not changed and therefore this does not play to Copilot's or any other LLM's detriment.

This paper submits that an analysis of how the code is transformed is not relevant for the purposes of the second criterion. Instead, the key component of the copyrighted work is its FOSS trait, which makes the purpose of its sharing socially beneficial and based in principles of solidarity, for which the FSF and OSI stand. Copilot, by using the code for its commercial use and creating a barrier in access (by creating a paid subscription model) to such code is violating these principles. This is especially the case given that inherent in the principle of fair use is the notion of equity. Using a public source intended for social benefits and premised on mutual solidarity and an effort to continue creating free coding sources using free code (premised in the principles of copyleft), speaks against Copilot's claim that it falls within fair use.

c) The Amount and Substantiality of the Portion Used

Arguably, the amount and substantiality of the portion of the copied work that is used in relation to the copyrighted work weighs the heaviest against Copilot. Indeed, Rothchild argues that the bigger the amount of the copyrighted work used, the less likely is the court to find its use to be fair.[210] AI machine learning as a process inherently uses the totality of the accessed sources. By OpenAI's own admission, the more data the engine consumes, the better the AI performs.[211]

[204] *Harper & Row* 471 U.S. 539, 563, 105 S.Ct. 2218 (1985).
[205] Kuhn, Free Software Foundation 2022, p. 4.
[206] *Authors Guild v. Google* 804 F.3d. (2d Cir. 2015).
[207] Howard p. 1.
[208] Kuhn, Free Software Foundation 2022, p. 4.
[209] *Authors Guild v. Google* 804 F.3d at [213].
[210] Rothchild, Free Software Foundation 2022, p. 4.
[211] USPTO, Comment Regarding Request for Comments on Intellectual Property Protection for Artificial Intelligence Innovation, p. 7.

As such, the amount and substantiality of the portion that Copilot uses in relation to the copyrighted work is total and absolute.

However, Grimmelmann argues that the verbatim copy of a whole work can still be fair use if the copy is fed into a process that does not involve processing the works expressively. [212] This argument could apply to human consumption, however, fails for source code. Arguably, a computer cannot process written work expressively as expressiveness inherently implies human processes, thoughts, creativity and/or emotion. Indeed, this view is supported by Howard who claims that should the output be used expressively, the copyrighted work's source code would lose all meaning to the machine. As such, Howard argues that this argument is void.[213]

Indeed, OpenAI does not seem to be disputing this fact in its paper for the USPTO. Instead, it focuses on the case of *Authors Guild v. Google* which stated that emphasis should rather be placed on the "amount and substantiality of what is thereby made accessible to a public for which it may serve as a competing substitute."[214] OpenAI uses this citation to argue that it is essential for AI to be able to be trained on vast amounts of data to increase the accuracy of AI as an innovative system with great potential. This argument, however, lacks in two main respects.

Firstly, while it may be true that the more data is used in machine learning, the more accurate the AI, this does not relate to the question of whether Copilot is a substitute. Copilot is a subscription-based service and imposes a relatively high fee. This is a barrier to many and, therefore, Copilot is not making a service freely available to the public. Secondly, it can be argued that Copilot is not creating a close substitute to the copyrighted work. Copilot is creating a digital tool that is interactive and can develop. The copyrighted works are static groups of data and code that require human input to be changed. Thirdly, the citation is taken out of context. The case concerned a claim that Google was violating copyright by scanning books to make them searchable on Google Books, a platform for online books. Even though the case was held to fall under fair use, it is integral to note that Google does not reproduce the content as a whole.[215] Instead, it displays small parts of the books. Copilot, by contrast, scans the totality of the work and the totality of the work can then be reflected in Copilot's output. Howard supports this statement by arguing that in theory, Copilot has no limit on how much copied output it can reproduce.[216]

Fourthly, OpenAI states that "this factor asks whether 'the quantity and value of the materials used,' are reasonable in relation to the purpose of the copying."[217] This proportionality test is derived from the case of *Campbell*.[218] However, even if the purpose of the copying is to be considered, the criterion in the case of Copilot speaks against it. The purpose of OpenAI's Copilot is primarily commercial gain. The creation of an AI that aids software developers can be seen as a beneficial side product at best – one which is also not made accessible to everyone due to its paywall.

---

[212] Grimmelmann, ILR 2015, 657–664.
[213] Howard, p. 4.
[214] *Authors Guild v. Google* 804 F.3d at [202].
[215] Band (2006).
[216] Howard p. 5.
[217] USPTO, Comment Regarding Request for Comments on Intellectual Property Protection for Artificial Intelligence Innovation, p. 6.
[218] *Campbell v. Acuff-Rose Music* 510 U.S. at [586].

d) The Effect of the Use on the Potential Market

OpenAI argues that not allowing AIs to train on public data would cause a heavy burden on the development of AIs using machine learning.[219] Academics such as Howard agree with this notion.[220] However, for the purposes of this criterion, it is integral to consider the effect that OpenAI has on the FOSS industry by taking the code and copying it for machine learning of Copilot's AI.

Howard argues that the source code generated by AI has the same market as the copyrighted FOSS source code and therefore serves as a substitute.[221] Indeed, case law seems to confirm this correlation. In *Authors Guild v. Google*, the court found Google's copying of another company's books to fall within "fair use" in part because the service provided by Google was not a direct substitute.[222] As such, the rule seems to be that where it does not create a market substitute, the court is more likely to find fair use. After all, it would not be conventionally fair to enable a party to profit from another's party work and create market competition. This is one of the problems which the law of intellectual property seeks to address.

The potentially detrimental effect on the FOSS industry is severe. Some academics argue that FOSS is a cornerstone in software development and open-source languages like R, Python or Javascript.[223] Some academics would, however, argue, that the protection of FOSS is not necessary at all given that proprietary software yields satisfactory or even better outcomes. The cornerstone of this paper is that FOSS is integral for software development, and as such a social good that requires protection.

aa) The Arguments for FOSS

FOSS distinguishes itself from proprietary software by being easily accessible, modifiable, and reviewable.[224] This makes the detection of any potential malware much easier and more transparent.[225] This transparency is a part of what makes many argue that FOSS is a more ethical form of coding than proprietary. Proprietary software namely places the owner of the code in a position of power over the user, coercing the user into accepting the software, its terms, as well as potential threats such as malware. The "hidden code" in proprietary software often conceals what the Free Software Foundation labels as "malicious functionalities," namely functionalities that are aimed at decreasing the useability of certain features, or disabling any free competition within the software industry, or even placing a certain limitation on the user's use of the software.[226] Generally, these are largely aimed at increasing profitability, but at the user's expense.[227]

The FSF has created a forum for sharing instances of companies using proprietary software creating some of these malicious functionalities, to help spread awareness of these practices.

---

[219] Howard p. 2.
[220] *Ibid.*
[221] Howard p. 5.
[222] *Authors Guild v. Google* 804 F.3d 202 at [211].
[223] Vaughan-Nichols, GitHub's Copilot flies into its first open source copyright lawsuit, https://www.theregister.com/2022/11/11/githubs_copilot_opinion/, accessed 22 February 2023.
[224] McKusick p. 31-46.
[225] *Ibid.* p. 40.
[226] Proprietary Malware, https://www.gnu.org/proprietary/proprietary.en.html, FSF, accessed 20 February 2023.
[227] *Ibid.*

On the list are companies such as Apple, Xiaomi, Whatsapp, Amazon, Zoom, Android, HP or even Volkswagen.[228]

Further, making software proprietary does not necessarily mean that the software will be more secure. In fact, several prominent security cases show otherwise.[229] Notably, a 2022 Tesla hack showed that creating new car keys, unlocking cars, starting engines, and even blocking access to owners was possible. The same hackers have reported that they were able to disable security and control 25 cars.[230]

The lack of transparency of proprietary software means that any malware, whether intentional or unintentional, gets spotted, inspected and/or corrected slower than with FOSS.[231] One might argue that the extent of the transparency of FOSS would prevent a large proportion of these malware functionalities from occurring (or being inserted) in the first place due to public accountability or public image.[232]

Indeed, studies published on the matter consistently show that open source scores higher in reliability upon being tested with a series of prompts.[233] A study done by Miller et al. measured reliability by giving both proprietary and open source programs random characters and recording the probability of them crashing or freezing up.[234] While the approach is limited in the sense that it cannot detect subtler failures, it allows the comparison of the programmes.[235] The study records that the commercial systems studied had an average fail rate of 23%, while Linux had a 9% and GNU 6% fail rate. A subsequent study by Forrester and Miller further confirmed that proprietary software was found to be less reliable than FLOSS.[236] The study found that Windows NT GUI applications could crash 21% of the time.[237] A 2020 study by Miller and Zhang comparing FreeBDS, MacOS and Linux found that Linux had the lowest fail rate (at 12%) while MacOS had a 16% fail rate.[238]

This analysis shows that not only does FOSS help prevent malicious functionalities, but it also helps eliminate security risks and produces more reliable code. There is a strong argument to be made for FOSS either co-existing alongside proprietary software, as some academics such as Rothchild argue,[239] or even as FOSS overtaking as the sole form of software, as radical FOSS advocates such as Stallman[240] or academics such as Kuhn argue.[241]

---

[228] *Ibid.*
[229] *Ibid.*
[230] Nicholas, Teen hacker says he's found way to remotely control 25 Tesla EVs around the world, https://fortune.com/2022/01/12/teen-hacker-david-colombo-took-control-25-tesla-ev/, accessed 23 March 2023.
[231] De Laat, Research Policy 2005, 1511-1532, p. 9-13.
[232] *Ibid.*
[233] Wheeler, Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!, https://dwheeler.com/oss_fs_why.html, accessed 16 March 2023.
[234] Miller, Computer Sciences Technical Report University of Wisconsin-Madison 1995, p.1.
[235] Wheeler, Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!, https://dwheeler.com/oss_fs_why.html, accessed 16 March 2023.
[236] Forrester, USENIX Windows Systems Symposium 2000, p. 8-9.
[237] *Ibid*. p. 1.
[238] Miller, IEEE Transactions on Software Engineering 2021, p. 5.
[239] Rothchild, Free Software Foundation 2022, p. 5.
[240] Stallman, Why Open Source Misses the Point of Free Software, https://www.gnu.org/philosophy/open-source-misses-the-point.en.html, accessed 17 March 2023.
[241] Kuhn, Free Software Foundation 2022, p. 1.

bb) The Impact of Copilot on FOSS

It has been argued by many FOSS advocates, including Mr Butterick, one of the leading claimants of the Copilot lawsuit, that GitHub Copilot and similar systems actively undermine the principles of FOSS,[242] exploit the principle of solidarity operating within the movement and ultimately, could destroy the movement altogether.[243] This threat might be exacerbated by the uncertain legislative landscape and the relative lack of FOSS-driven policy decisions.[244]

Laurent points to the strong sense of solidarity amongst the FOSS community.[245] Members of the community devote their free time to developing and distributing good code for the benefit of as many people as possible. Exploiting the good intentions of the community can be viewed as a moral wrong in itself. Indeed, Laurent claims that the moral principle is the most important factor enforcing the licenses.[246] Choksi points to this from a similar perspective, there is a risk of normative consequences, namely the perceived precedence of corporations' profit-making over FOSS for training and distributing code from FOSS.[247] At the scale at which Copilot operates, this is significant.

Secondly, an immediate consequence is strongly disincentivising and discouraging the FOSS community's voluntary activities in writing better code.[248] This can lead to a gradual decrease in FOSS code analysis and improvement. Companies that have been using FOSS and opened their software will be incentivised to close it.[249] This, in turn, could disrupt the creation of good code that is free of malware and promote proprietary code. Bug fixes may be kept hidden, decreasing the security of code and consequently, computer systems.[250] Moreover, Copilot will enable companies to "launder code", a method that involves viewing FOSS licensed code, and then using Copilot to generate something very similar to the source code, thereby exploiting the code without FOSS license compliance.[251]

Individuals will be disincentivised to work on FOSS code voluntarily.[252] The profession of working as a software developer can be permanently affected.[253] Although one might argue that the monetary loss caused by Copilot is not as significant given that FOSS code is provided for free, the significance lies in the service that Copilot seeks to provide, which could replace a significant part of the service that software developers provide now.[254]

---

[242] Seddon, Rutgers University Law Review 2016, 251-287, p. 2.
[243] Kuhn, Free Software Foundation 2022, p. 1.; Butterick, This Copilot Is Stupid And Wants To Kill Me, https://matthewbutterick.com/chron/this-copilot-is-stupid-and-wants-to-kill-me.html, accessed 10 March 2023.
[244] Engler, How open-source software shapes AI policy, https://www.brookings.edu/research/how-open-source-software-shapes-ai-policy/, accessed 3 March 2023.
[245] St. Laurent, p. 158.
[246] *Ibid.*
[247] Choksi, Conference on Human Factors in Computing Systems 2023, p. 2-4.
[248] St. Laurent, p. 158.
[249] Howard p. 10.
[250] *Ibid.*
[251] *Ibid.*
[252] St. Laurent, p. 158.
[253] Kuhn, Free Software Foundation 2022, p. 1.
[254] *Ibid*. p. 1.

cc) The Substitutive Nature of Copilot

A large proportion of this debate hinges on the extent to which Copilot is a direct substitute for the copyrighted works. Arguments speaking for this are the fact that the same code is used to train the AI which then provides, in essence, a result based on the copyrighted work.

However, one could argue that Copilot is different because it introduces new features and unexpected outputs, including some mistakes in code. For example, in writing a JavaScript function for calculating the nth prime number, Copilot filled in the rest of the code with a function that does not exist.[255] The level of reliability could, therefore, mean that Copilot is not a perfect substitute. On the other hand, some developers would argue that no code is ever perfect and might respond well or break down depending on the rest of the code to which it is applied.[256] Indeed, the above-quoted studies showed that even FOSS has "bugs" and freeze-ups. The incidence of these bugs was generally lower, however.[257]

Thirdly, Copilot and FOSS software generally operate in the same market.[258] It is a market for software developers to use, enhance, develop and edit code. While one could argue that the paid subscription model for Copilot is a barrier and therefore distinguishes the market, the target audience for both models is software developers.[259] In fact, Rothchild argues that this factor specifically asks to consider how far the unauthorised use affects the ability of the owners of the copyrighted work to derive some sort of economic impact from their copyrighted work.[260] In this case, as argued by Howard and Butterick, Copilot can cause software developers' markets to shrink and their value to decline.[261]

e) Evaluation

Given the matrix of factors and the case law specifying how much emphasis ought to be placed on each criterion within the fair use analysis, the outcome of the court's decision is uncertain. Nevertheless, it is submitted that the commercial purpose of Copilot, the nature of the copyrighted work, and the large proportion of the copyrighted work used all weigh against Copilot. Additionally, the potentially very detrimental impact that Copilot might have on the FOSS movement, software development and the software market as a whole, weighs very heavily against Copilot's cause.

Crucially, as Kuhn highlights, the fair use doctrine is an affirmative defence to a copyright violation claim.[262] The defendant bears the burden to prove that their actions are not a copyright violation, which is an additional procedural factor that could mean that the scales are tilted against Copilot.

[255] Kuhn, Free Software Foundation 2022, p. 1.
[256] Forrester, USENIX Windows Systems Symposium 2000, p.7-9.
[257] *Ibid.*
[258] Howard p. 5.
[259] Rothchild, Free Software Foundation 2022, p. 5.
[260] *Ibid.*
[261] Kuhn, Free Software Foundation 2022, p. 1.
[262] Kuhn, Free Software Foundation 2022, p. 4.

VI. Liability under EU Law

With reference to the EU protection framework discussed above in section D., the specific application to Copilot will be discussed in the following sections. Specifically, the transformations and uses of data could be protected by the right of adaptation and the right of reproduction, however, the TDM exception might apply.

1. Right of Adaptation

As stated above, the author of the code deposited on GitHub has the exclusive right to control and authorise any adaptations or modifications.[263] An adaptation is the creation of a derivative work, which, as argued by academics such as Margoni, could apply in the situation of AI processing corpora of data for ML.[264] Namely, the process of inputting code and data into the AI and the AI producing an output could be considered an adaptation of the original text. This application is, however, relatively less applicable to the current situation than the right of reproduction.[265] Moreover, some exceptions apply in cases where it is necessary for the intended use, to study the program (art. 5) and for decompilation (art. 6).[266] In this case, the exception under article 5(1) is most applicable as it allows unauthorised adaptations "where they are necessary for the use of the computer program by the lawful acquirer in accordance with its intended purpose, including for error correction."[267]

However, this is very limited in scope. Firstly, commentators emphasise that the scope of this exception relates to educational purposes, the definition of which is outlined in Recital 20.[268] Even if Copilot were conceived to hold an educational purpose, for-profit teaching platforms cannot be considered within this exception. Moreover, the article provides that the exception applies "in the absence of specific contractual provisions".[269] As such, by including a license to the code which requires attribution, the user of the code must follow the contractual provision.

2. Right of Reproduction

The exclusive right of the author to authorise reproductions might apply to Copilot. The right is defined as "any direct or indirect, temporary or permanent reproduction by any means and in any form, in whole or in part".[270] This also encompasses temporary reproductions such as those made in the RAM of a computer.[271] This means, that in the process of machine learning, when the AI makes a copy and stores it, the application of this right is triggered. Secondly, this right can be triggered with regards to the final stage where the output is displayed and produces content that is a copy of the original text.

---

[263] Directive 2009/24/EC art. 4(b).
[264] Margoni, CREATe Working Paper 2018.
[265] Margoni, CREATe Working Paper 2018.
[266] Directive 2009/24/EC art. 5 and 6.
[267] *Ibid.* art. 5(1).
[268] Jütte, The New Copyright Directive: Digital and Cross-border Teaching Exception (Article 5), https://copyrightblog.kluweriplaw.com/2019/06/21/the-new-copyright-directive-digital-and-cross-border-teaching-exception-article-5/, accessed 27 March 2023.
[269] Directive 2001/29/EC art. 5(1).
[270] *Ibid.* art. 2.
[271] *Ibid.*

The comparison between the process of training Copilot and the data capture process from *Infopaq* will determine whether the reproduction right can be considered violated.

It is questionable whether the training of an AI is comparable to the data capture process that has been analysed in *Infopaq*. The first step of training an AI is compiling a corpus of data, a process that could be analogous to obtaining publications from an electronic database in *Infopaq*. Secondly, the creation of a special file from the publications and uploading them on the OCR server is essentially comparable to the process of pre-training where text is converted into an NLP tool-readable format (plaintext). Thirdly, the translation of a human-readable text into a computer-readable one is very similar to the process of NLP where words are annotated and enriched to be readable by the AI. The process of searching for possible matching words and displaying five words before and after the searched word is very similar to the ML process as well because an ML analyses patterns in a probabilistic way the most possible matches to words. The fifth stage comprises in both cases of the production of an output that displays the matches to the given inputs. The main difference lies in the fact that an ML might not always produce an output that constitutes an identical copy.

As such, the ML process of NLP is very similar to the data capture process and it is possible that a court might evaluate the legal position of programmes such as Copilot or other NLPs in a similar manner, holding that it does not fall within the exception of art. 5(1) which allows temporary acts or reproductions that are transient and necessary for non-commercial purposes,[272] meaning that the right to reproduction under art. 2 of the InfoSoc Directive is violated. Ultimately, it could also be argued that the exception under does not apply because Copilot, as established, uses the code for commercial purposes.

3. TDM Exceptions

Despite this criticism, it is essential to consider whether the exceptions under art. 3 and 4 can be considered to apply to NLP cases such as Copilot. Given that machine learning falls under the definition and scope of TDM because it involved the analysis of large amounts of data for detecting patterns, the exact legal position will depend on two factors:

- Whether the materials are lawfully accessible – either by permission, contract or license; and
- whether the author of the material has opted out of the right to use the materials for TDM following art. 4(3).

In the case of AI such as Copilot, lawful access will be regulated via licenses. The licenses on which Copilot has been trained are public, accessible, and free. The position is, therefore, contingent on the opt-out rights solely, the exercise of which would mean that the use might violate copyright.

The licenses do not include an express opt-out right. However, one might consider whether the opting out can ensue impliedly. By way of analogy, the Creative Commons "CC" organisation has issued a report confirming that CC licenses cannot be construed as containing an opt-out provision for art. 4(3).[273] Reasoning by analogy, we may extrapolate this principle for FOSS

[272] Directive 2001/29/EC art. 5.
[273] Lazarova, Creative Commons Statement on the Opt-Out Exception Regime, https://creativecommons.org/wp-content/uploads/2021/12/CC-Statement-on-the-TDM-Exception-Art-4-DSM-Final.pdf, accessed 12 March 2023.

licenses as well, especially given that these licenses are intended to not restrict their use. Indeed, art. 4 of the directive specifies that the opting out shall be done by "machine-readable means" when it concerns content online. Recital 18 further confirms that "it should only be considered appropriate to reserve those rights by the use of machine-readable means, including metadata and terms and conditions of a website or a service. […] In other cases, it can be appropriate to reserve the rights by other means, such as contractual agreements or a unilateral declaration".[274] In theory, therefore, a software developer using GitHub or publishing FOSS can opt-out from machine learning freely. In the current Copilot case, however, it cannot be imputed to the licenses that the authors intended to opt-out. There is no wording in the licenses to suggest it. Moreover, the case of CC which can be viewed as an analogy to FOSS, speaks against such an interpretation, as does the general purpose of the licenses. As such, AIs that use publicly available sources, including Copilot, are generally eligible for the TDM exception.

G. Other Cases of AI Violating Copyright

With the rapid growth of the AI industry, lawsuits such as Copilot are only likely to become more frequent. Particularly, generative AI seems to be on the rise. Through November 2022, it has managed to secure over $1.3 billion in funding through venture capital alone, which was a 15% yearly increase.[275] It is no surprise that as a new technology, it leads to legal uncertainties and consequently, litigation.

A fascinating example is the case of Stable Diffusion, an AI image generator, which has had two lawsuits filed against it for allegedly violating copyright by feeding images to the AI engine for ML.[276] One of the claimants is Getty Images, a company producing stock images with a "Getty images" watermark. It is alleging violations of the copyright of millions of images and claiming over $1.8 trillion.[277] As Chittock states, it is a landmark case as the first case in the UK on the copyright protection of data used in ML.[278] Stable Diffusion takes open source data from many stock photo websites. Getty Images claims that Stable Diffusion unlawfully copied and processed millions of images protected by copyright without having a license.

A sample study of 12 million images conducted by Waxy found that Diffusion has copied the works from Getty images approximately 1.88% of the time. This is a significant figure considering that Stable Diffusion created over 170 million outputs as of October 2022.[279] Moreover, there is evidence of the copying as the AI has, on many occasions, reproduced the Getty Images watermark in its output.[280] As with Copilot, it appears that the issue is more about the lack of attribution rather than the extent of similarity of the works.[281]

---

[274] Directive 2019/790 art. 4.
[275] Field, Three inflection points for emerging tech in 2022, https://www.emergingtechbrew.com/stories/2022/12/21/three-inflection-points-for-emerging-tech-in-2022, accessed 21 April 2023.
[276] *Ibid.*
[277] Case 1:23-cv-00135-UNA.
[278] Chittock, Getty Images taking UK action against Stability AI for copyright infringement in AI training, Lexis Nexis Legal News, January 2023, accessed 7 May 2023.
[279] Wiggers, Image-generating AI can copy and paste from training data, raising IP concerns, https://techcrunch.com/2022/12/13/image-generating-ai-can-copy-and-paste-from-training-data-raising-ip-concerns/, accessed 6 May 2023.
[280] Case 1:23-cv-00135-UNA.
[281] Getty Images Statement, https://newsroom.gettyimages.com/en/getty-images/getty-images-statement, accessed 12 April 2023.

However, as with Copilot, the outcome of the litigation is unclear because the courts are navigating unchartered territories. Academics speculate that the outcome of the case will largely hinge on the court's interpretation of the fair use doctrine, which is anticipated to be answered in the Copilot litigation later this year.[282] Hulbert has stated that he believes that the court is unlikely to deem this use as "fair use" given its commercial nature, however, the outcome of the litigation is highly questioned.[283] In any case, it will have a big impact on creative industries.[284]

H. Outlooks

Academics do not agree on whether AI requires immediate regulation, or whether its growth will be more incremental.[285] Kuhn argues that AI is more slow-moving than society believes, and the problem is not imminent nor irreversible. Despite this, Kuhn believes that legislation needs to respond immediately and deliberately.[286] Academics leaning towards copyleft activism such as Kuhn advocate for a comprehensive rewriting of copyright and intellectual property frameworks to protect the interests of the FOSS community.[287]

Arguably, copyleft developed in response to intellectual property law's stagnant character and the political non-viability of rewriting copyright.[288] Any legislative reform underlies strong lobbying forces, in this case, the particularly strong proprietary software industry.[289]

Due to various pressures, the normative uncertainties, the factual uncertainties surrounding the machine learning of AI and the absence of any comparable legal precedent make the outcome of the GitHub class-action lawsuit uncertain. In any case, there are a few ways in which Copilot and similar LLMs training on public code could adapt to be compliant, from least intrusive – such as training attribution, to more intrusive - such as new licensing models, to most intrusive – a regulatory ban on such LLMs.

I. Training Attribution

One proposition is to train Copilot to keep the source information attached to each section of code. The output would then provide the option to inspect the code and its license. This would also enable further beneficial features on the AI, such as the option for users to generate code with specifying a license type that was used on the data. The user would still have the responsibility to verify the terms of the license and adhere to them, following the principle of the chain of custody. This is a proposition also made by Butterick in his blog post critiquing Copilot.[290] However, at this moment, this proposition is purely speculative and theoretical. The

[282] Growcoot, Getty Images is Suing Stable Diffusion for a Staggering $1.8 Trillion, https://petapixel.com/2023/02/07/getty-images-are-suing-stable-diffusion-for-a-staggering-1-8-trillion/, accessed 5 May 2023.
[283] Wiggers, Commercial image-generating AI raises all sorts of thorny legal issues, https://techcrunch.com/2022/07/22/commercial-image-generating-ai-raises-all-sorts-of-thorny-legal-issues/, accessed 6 April 2023.
[284] Chittock, Getty Images taking UK action against Stability AI for copyright infringement in AI training, Lexis Nexis Legal News, January 2023, accessed 7 May 2023.
[285] Kuhn, Free Software Foundation 2022, p. 2.
[286] *Ibid.*
[287] *Ibid.*
[288] *Ibid.*
[289] *Ibid.* p. 2.
[290] Butterick, This Copilot Is Stupid And Wants To Kill Me, https://matthewbutterick.com/chron/this-copilot-is-stupid-and-wants-to-kill-me.html, accessed 10 March 2023.

reliability of the material is promoted if it can be verified – which is not possible in AI which does not keep track of all of the sources and terms of the code used. This also works against Copilot in the sense that currently, many users see Copilot as an "IP rights black hole" that cannot be trusted and deters users from leveraging the benefits of Copilot.

Howard agrees that training attribution is essential. As an example, in the *Authors Guild v. Google* case, Google Books keeping the source information and displaying the name, licenses, titles and other important details about the book ensured proper attribution and led to a finding of no copyright violation.[291] The user can therefore obtain information about how to legally obtain copies of the work. Thus, it has been proposed that Copilot does two things: 1) track the licenses of each copyrighted work (including source code copied from a copy) and 2) whenever Copilot produces an output, it must determine the source of the code that influenced the output.

However, there are large technical shortcomings to this method, leading many software programmers to strongly question whether this can be done at all. Firstly, AI operates in a probabilistic way – the input does not directly reflect in the output – so knowing and determining where the output comes from might be extremely difficult, if not impossible, to achieve. Secondly, thousands of sources can go into the production of one snippet of output. This also severely complicates the ability of the AI to determine which source it originated from. An alternate but more realistic model is one where the AI develops a new separate ability to detect the source which it most resembles (but not necessarily the one which it sourced from) and attribute it to this one. This might, however, undermine the most basic principles of copyright – namely that the credit goes to the source of the output which it has copied.

I. New Licensing Models

A more restrictive, but not all-encompassing solution would be the creation of a new open-source license that does not allow the code to be used for AI training. There are a few problems with this.

Firstly, such a restriction on use might go against the ethos of FOSS. One of the goals of FOSS is to promote the development of software for the benefit of society. Preventing and restricting the use of code for some purposes, whether commercial or not, might thwart innovation. As Kuhn argues, AI progress cannot be abruptly stopped.[292] Secondly, it might be too all-encompassing. Not all AI systems will be unethical, especially as second and third AI generations develop. Open source authors would be contradicting themselves – technological progress lies at the core of open source, yet this measure would seek to hinder it.[293] Thirdly, it is inconsistent to hold AI systems to a different standard than human users of FOSS code. The scale at which AI machine learning operates (as opposed to a human software developer using source code) might, however, justify a different approach to licensing. While nobody can guarantee the behaviour of a non-deterministic system, non-determinism is not a sufficient defence to misbehaviour – however novel or innovative a technology is.[294] As such, an AI ought to be prevented from abusing FOSS and justifiably held to a different standard than human users.

[291] Authors Guild v. Google 804 F.3d 202 (2d Cir. 2015).
[292] Kuhn, Free Software Foundation 2022, p. 2.
[293] *Ibid*. p. 2-7.
[294] Moglen, First Monday 1999, p. 2.

There have been recent developments regarding licenses tailored for machine learning. Linux had developed the CDLA license which seeks to create a short and simple permissive license for the sharing and usage of open data, particularly for machine learning of AI.[295] The license seeks to address the issue of how data may be transformed in AI and machine learning, for which traditional license models such as the creative commons licenses are not equipped.[296] As such, the license permits the right to use, share, and modify the data and use any output generated through computational analysis. The only restriction is the requirement to "make available the text of this agreement with the shared Data" together with a disclaimer of warranties and liability.[297] Linux compares the general nature of the license to the permissive MIT or BSD-2 licenses but specific data and more limited liabilities.[298]

The CDLA license was considered by many as overly permissive. Partly as a reaction to this, BigScience, an open science initiative, developed the RAIL license.[299] The license is a good illustration of the software community's effort to prevent the abuse and undermining of FOSS principles. It is developed specifically for the data used to train LLMs, imposing "behavioural-use terms on the use of the model."[300] This is a novel development, as previous license types have generally covered software but not the data to train AI. It is not an open source license as it restricts the use of the model in some ways. However, it does allow reuse, redistribution, adaptation or commercialisation.[301] The goal is to enable AI researchers concerned about the misuse of their AI models but who would still like to share their work to advance software development. The nature of the license is therefore relatively permissive save for the use-based restrictions.

These licenses are a testament to the dynamic world of licensing and software development and the effort to maintain FOSS freedom and prevent commercial corporate abuse and could illuminate a path forward for many of the legal issues presented by machine learning of NLP AIs.

II. New Regulation

Some experts, such as Burt, founder of an AI law firm, believe that employing generative AI as extensively as is being done, without undertaking legislative measures to address any risks, is problematic. This puts pressure on businesses to predict and anticipate the legal situation.[302]

---

[295] The Linux Foundation, Enabling Easier Collaboration on Open Data for AI and ML with CDLA-Permissive-2.0, https://www.linuxfoundation.org/press/press-release/enabling-easier-collaboration-on-open-data-for-ai-and-ml-with-cdla-permissive-2-0, accessed 2 April 2023.
[296] Vézina, Should CC-Licensed Content be Used to Train AI? It Depends., https://creativecommons.org/2021/03/04/should-cc-licensed-content-be-used-to-train-ai-it-depends/, accessed 7 April 2023.
[297] The Linux Foundation, Enabling Easier Collaboration on Open Data for AI and ML with CDLA-Permissive-2.0, https://www.linuxfoundation.org/press/press-release/enabling-easier-collaboration-on-open-data-for-ai-and-ml-with-cdla-permissive-2-0, accessed 2 April 2023.
[298] *Ibid.*
[299] BigScience RAIL License, https://bigscience.huggingface.co/blog/the-bigscience-rail-license, accessed 3 April 2023.
[300] BigScience RAIL License, https://bigscience.huggingface.co/blog/the-bigscience-rail-license, accessed 3 April 2023.
[301] *Ibid.*
[302] Wiggers, The current legal cases against generative AI are just the beginning, https://techcrunch.com/2023/01/27/the-current-legal-cases-against-generative-ai-are-just-the-beginning/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAM32E1ub5qz3LTNg7-

Meeker also urges legislative measures as he views the legal uncertainty as a possible reason why the industry and AI businesses could be crippled, and limit Innovation.[303]

Indeed, countries worldwide have registered a significant effort to regulate AI. The proposed EU AI Act is a prominent example of the effort, but it focuses on classifying the risk of AI and regulating the development thereof according to the level of risk. While it seeks to address data quality, transparency and liability,[304] the act does not address the protection of data for the purposes of ML of AI.[305] A part of the problem with regulating ML is the difficulty of formulating specific requirements for the AI. It is unclear, whether training attribution is possible, and if yes, then the extent of attribution required is debatable. However, some academics propose that the mere matter of increasing transparency with regards to ML inputs would be desirable. For instance, Kuhn talks of the auditability of AI output on systems such as Copilot, whereby regulation would impose certain disclosure obligations on the company with regard to the range of possible sources for their machine learning input.[306] One could view this as too lenient as it does not address attribution, but merely specifies a range of possible sources from which the AI has potentially violated copyrights. A more restrictive approach may, however, encounter more technological barriers. An outright ban on ML using FOSS code might be too restrictive. Secondly, the passing of such acts will likely be met with resistance from the lobbying forces of Big Tech.[307]

Interestingly, some countries have signalled an interest in adopting a permissive approach to publicly available works. The UK proposal to change existing legislation to allow TDM for any purpose, if implemented (which is uncertain), would be one of the most lenient in the world.[308] In the US, there seems to be a more careful approach to this – and some experts, such as Torres, do not expect any change soon.[309]

Importantly, the EU may wish to revisit the existing TDM exception regulation, considering broader policy factors, pragmatic reasons and doctrinal issues. Academics such as Kretschmer have questioned the appropriateness of regulating a major developing area of AI machine learning through a mere exception to a provision in article 4.[310] There have been further criticisms raised about the inconsistency of the protection that is afforded by allowing TDM for any purpose at all, arguing that the EU is herewith protecting mere facts and data, which do

---

Zu13tVt81Dg_TQguogGlR_yU2aBZqKEsXfaArxAJ1YrxK_S1KyNq8QLYX2UkwTYvbsycjpF1IqkZGUVDP pFp2OlwnupSYCNMfvILuIkDd0cqd66XPnTND6SbRWY0KTFAQduMEG7zJPVT0qj1e603xE7X-7, accessed 3 April 2023.

[303] *Ibid.*

[304] European Commission Document SEC(2021)167.

[305] *Ibid.*

[306] Kuhn, Free Software Foundation 2022, p. 1-7.

[307] Butterick, This Copilot Is Stupid And Wants To Kill Me, https://matthewbutterick.com/chron/this-copilot-is-stupid-and-wants-to-kill-me.html, accessed 10 March 2023.

[308] HM IPO, Artificial Intelligence and Intellectual Property, https://www.gov.uk/government/consultations/artificial-intelligence-and-ip-copyright-and-patents/outcome/artificial-intelligence-and-intellectual-property-copyright-and-patents-government-response-to-consultation, accessed 1 May 2023.

[309] Wiggers, The current legal cases against generative AI are just the beginning, https://techcrunch.com/2023/01/27/the-current-legal-cases-against-generative-ai-are-just-the-beginning/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAM32E1ub5qz3LTNg7-Zu13tVt81Dg_TQguogGlR_yU2aBZqKEsXfaArxAJ1YrxK_S1KyNq8QLYX2UkwTYvbsycjpF1IqkZGUVDP pFp2OlwnupSYCNMfvILuIkDd0cqd66XPnTND6SbRWY0KTFAQduMEG7zJPVT0qj1e603xE7X-7, accessed 3 April 2023.

[310] Margoni, GRUR International 2022, pp. 685–701.

not pass the (arguably low) EU originality threshold. Indeed, Kretschmer raises the point that article 3 is made relatively redundant by article 4. Importantly, the act ought to clarify how to opt in after having opted out.[311] One may argue that the opt-out mechanism exploits authors' inertia. The possibility of an opt-in mechanism (rather than opt-out) might therefore be worth considering to better protect authors' rights, and, as Kretschmer believed, to help "recalibrate" art. 3.[312] The policy goal of pursuing a booming EU AI industry ought not come at the cost of exploiting authors' rights.

I. Conclusion

Much discussion has been led on the copyright status of AI output, or the AI's code itself. This thesis sought to demonstrate the importance of considering the copyright status of the materials which are fed into an AI, given the unpredictability of AI output. The Copilot case serves as a good illustration of this dilemma, albeit in the US jurisdiction. The ML of NLP AIs will likely remain a contentious topic from a copyright perspective, as seen in the Getty Images lawsuit. Bringing attention to the possible exploitation of FOSS and other copyrighted works by Big Tech is therefore critical.

The copyright frameworks in the EU mandate countries to regulate and protect software under copyright as a literary work. The same protection also applies in the US. The acquis communautaire has developed a range of legal doctrines which are protected – such as the right to reproduction, the protection of databases and, arguably, a right of adaptation. The EU frameworks also contain exceptions which can be considered when discussing the protection of input into AI, namely of TDM. These can exempt an AI from copyright violation liability in cases of non-commercial uses or where the owner has not opted out of TDM. The US doctrine of fair use, unlike the EU, is not tailored specifically to data mining but rather considers a range of factors. The outcome of the litigation is very dependent on the interpretation of the court.

The FOSS licensing models are premised on allowing use and distribution, as long as proper attribution is given. This thesis sought to argue that code produced by Copilot would violate the license terms because no attribution is given, thereby fundamentally undermining the principles around which the FOSS movement is centred.

Despite this, the legal analysis showed that in the EU, the TDM exception will most likely apply to Copilot and many other LLMs. This is, however, subject to the authors' right to opt out of the use, which provides an attractive solution for developers who share their code as FOSS, to prevent exploitation by Big Tech. The thesis further argued that Copilot will not be eligible for the fair use exception in the US, given its strong commercial nature.

There are various policy, ethical and legal arguments to both protect the materials which AIs are trained on, as well as allow AIs to be trained on as much as possible. Mezei argues that in general, it is not justified to allow protection to AI merely as it is a lucrative and growing industry. [313] Indeed, this paper submits that the policymakers should look behind the money and consider the impacts on the software development industry, the copyright protection of code and ultimately, the protection of all copyrighted materials. Fundamentally, much of the regulatory debate centres around striking a fair balance between the goal of promoting AI development by not imposing onerous burdens on AI developers of having to negotiate

[311] Margoni, GRUR International 2022, p. 685–701.
[312] *Ibid*.
[313] Mezei, UFITA 2020, 390, (395). p.13.

licenses, but at the same time protecting copyrighted works to a sufficient degree so as not to discourage innovation. The normative debates are in essence a double-edged sword.

The assessment of an AI's liability on a case-by-case basis might appear as a lucrative and appeasing solution, however, the legal uncertainty associated therewith might call for a unified framework. Despite some regulatory attempts by the EU, more discussion needs to be turned to the protection of these materials from machine learning.

The AI industry is rapidly developing and is at the forefront of many governments' agendas. With investment into the industry growing, more litigation is likely to arise, with more copyright and licensing advocates coming to FOSS's rescue. It is questionable whether the advocacy of FOSS developers can safeguard and protect the community effort to promote the quality and accessibility of software. But it is essential that this effort is borne in mind when considering any AI-related copyright problem.

Appendix

I. The Principles of the Open Source Initiative

1. Free redistribution: The license must allow the unrestricted redistribution of the software. Therefore, no license or other kind of fee may be required for the use of the software.

2. Source Code: The program must be available in source code and permit redistribution both as source code and in compiled form. If a part of the product is not distributed with source code, it must be indicated that the source code can be downloaded from the Internet free of charge.
3. Works based on the program: The license must permit further development and modification of the Program. Likewise, the license must permit redistribution and distribution under the same license terms.
4. Integrity of the original code: The license may restrict the distribution of modified source code only if it permits the distribution of so-called "patch files" in conjunction with the original code so that the program can be modified before it is used. The license must explicitly permit the distribution of software created with modified source code.
5. No discrimination against individuals or groups: The license must not discriminate against any individual or group of individuals. For this reason, the OSI prohibits any open-source license from excluding anyone from the process.
6. No restrictions for specific fields of application: The license may not restrict anyone from using the program in a particular field of use. For example, it may not prohibit commercial use or use in genetic research.
7. Distribution of the license: The rights pertaining to the program must apply to everyone who has received the program. It is not allowed to link the license with another one.
8. The license must not apply to a specific product: The rights granted by the license must not depend on the program being part of a certain software distribution.
9. The license may not restrict other software: The license may not restrict other software that is distributed together with the licensed software.
10. The license must be technology neutral: No provision of the license may restrict the use of the software to any single technology or to any type of interface.[314]

II. GPL-3 License – full text

```
GNU GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <https://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The GNU General Public License is a free, copyleft license for
software and other kinds of works.

  The licenses for most software and other practical works are designed
to take away your freedom to share and change the works.  By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.  We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors.  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
```

[314] The Open Source Definition, https://opensource.org/docs/definition.php, Open Source Initiative, accessed 27 February 2023.

have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work

in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey 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; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you

received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible

for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the

option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <https://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

  If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show
w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

  You should also get your employer (if you work as a programmer) or
school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<https://www.gnu.org/licenses/>.

  The GNU General Public License does not permit incorporating your program
into proprietary programs.  If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.  But first, please read
<https://www.gnu.org/licenses/why-not-lgpl.html>.
```
[315]

III. MIT License – full text

```
MIT License

Copyright (c) [year] [fullname]

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
```

[315] GNU General Public Licence, https://www.gnu.org/licenses/gpl-3.0.txt, accessed 6 March 2023.

```
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.[316]
```

[316] MIT License, https://choosealicense.com/licenses/mit/, accessed 24 March 2023.