**UNITED STATES DISTRICT COURT**
**SOUTHERN DISTRICT OF NEW YORK**

| | |
|---|---|
| IN RE:<br><br>OPENAI, INC.<br><br>COPYRIGHT INFRINGEMENT LITIGATION<br><br>This Document Relates To:<br><br>1:23-cv-11195-SHS-OTW<br>1:24-cv-04872-SHS-OTW<br>1:24-cv-03285-SHS-OTW<br>1:24-cv-01515-SHS-OTW<br>1:25-cv-04315-SHS-OTW | 1:25-md-03143 (SHS) (OTW)<br><br>**DECLARATION OF DEFENDANTS' EXPERT NICK FEAMSTER IN SUPPORT OF MOTION FOR SUMMARY JUDGMENT**<br><br>REDACTED — PUBLIC VERSION |

I, Nick Feamster, Ph.D, declare under penalty of perjury, pursuant to 28 U.S.C. § 1746, as follows:

1. I submit this Declaration in support of OpenAI Defendants' (collectively, "OpenAI") Motion for Summary Judgment.

**A. Professional Background and Qualifications**

2. I am the Neubauer Professor of Computer Science and the Founding Director of the Center for Data and Computing (now the Data Science Institute) at the University of Chicago, where I have served as a full professor since July 2019. I previously served as a computer science professor at Princeton University and the Georgia Institute of Technology.

3. My work includes extensive research on web technologies, Internet protocols, and how web applications behave under different network conditions. I have also researched how web crawlers and bots operate and how websites implement access restrictions. This research includes

understanding the relationship between client-server architecture, HTTP protocols, and user agent strings.

4. I also have extensive practical experience with the technical and operational aspects of web crawling. I was an employee and lead technical engineer at LookSmart, an early search engine company, where I designed and wrote LookSmart's first web crawler and indexer in 1997. In this role, I designed crawler architecture to discover and retrieve web content efficiently, implemented `robots.txt` parsing and compliance processes designed to respect website owner preferences, and handled HTTP protocol interactions, user agent identification, and server responses. Through this experience I obtained a first-hand understanding of engineering decisions involved in operating web crawlers and fundamental principles of web crawling, including transparent crawler identification and respecting `robots.txt`.

5. I received my Ph.D. in Computer Science from the Massachusetts Institute of Technology (MIT) in 2005. My doctoral research developed new methods to analyze and predict the behavior of large-scale Internet routing systems, focusing on ensuring that networks route traffic correctly and predictably. I also earned an undergraduate degree in Electrical Engineering and Computer Science and a Master's degree in Computer Science from MIT in 2000 and 2001 respectively.

**B. Assignment**

6. I have been engaged on behalf of OpenAI to provide an expert opinion regarding technical issues and standard practices related to web crawling, the technical operation and widespread acceptance of the Robots Exclusion Protocol (REP) via `robots.txt` files, and website operator norms related to other Internet technologies relevant to this case. I was also asked to provide opinions regarding whether OpenAI's Browse functionality operates in a manner consistent with these widely accepted practices, protocols, and norms.

7. I submitted an opening expert report in this matter on April 24, 2026; a rebuttal report on May 29, 2026; an addendum and errata to my rebuttal report on June 15, 2026; a reply report on June 26, 2026; and a supplemental response to Dr. Tom Goldstein's supplemental report on July 21, 2026. I was deposed in this case on July 23, 2026. The statements in my reports are accurate and true to the best of my knowledge, information, and belief.

8. This declaration provides a concise summary of the opinions set forth in my reports related to the technical operation and development of OpenAI's Browse functionality, as well as the REP. These opinions are based on my knowledge, skill, experience, training, and education, as well as my review and analysis of documents and data as disclosed in my reports.

**C. Overview of ChatGPT's Browse Functionality**

**i. How ChatGPT's Browse Functionality Works**

9. Retrieval Augmented Generation or "RAG" is an AI-based capability that combines a large language model (LLM) with external information retrieval tools. A RAG-enabled LLM can retrieve relevant information from sources external to the LLM's training data, such as the Internet, using automated retrieval techniques.

10. OpenAI's "Browse" feature and its later "ChatGPT Search" feature allow the models underlying ChatGPT to search the web and retrieve information before generating an answer to a user query.[1]

11. To understand how OpenAI's Browse functionality operates, I analyzed 20 million ChatGPT conversation records provided to me by OpenAI. These records contain conversation transcripts along with their associated metadata, including fields reflecting operational events not

[1] OpenAI's web-browsing service has been known at various times as "Browse with Bing," "Browse," or "ChatGPT Search." For clarity, I refer to OpenAI's web-browsing service as Browse throughout my declaration.

visible to the user, such as whether certain system tools were involved, the type of message generated, and other structured indicators associated with each message, which provide meaningful insight into how Browse behaves in practice.

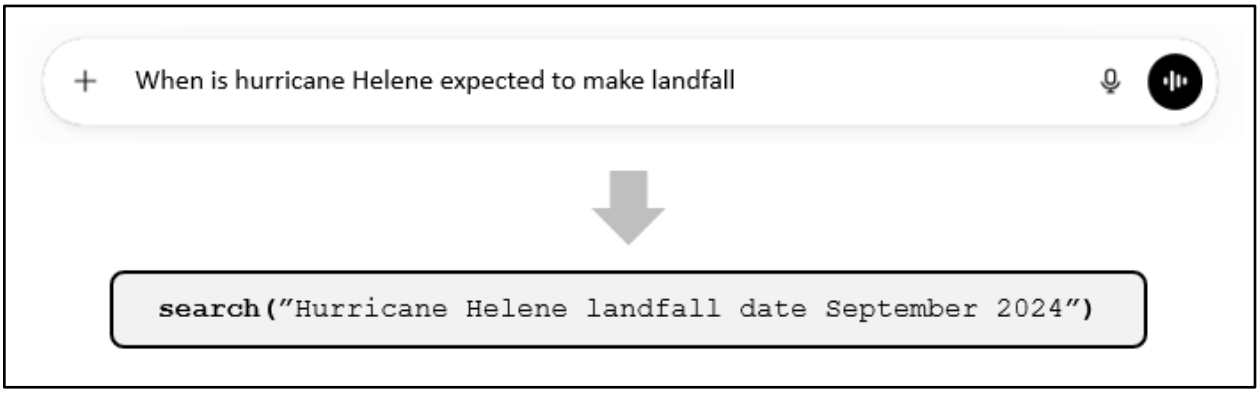
12. The end-to-end workflow of how ChatGPT synthesizes information drawn from the Internet in response to a user query when OpenAI's Browse function is invoked follows a five-step process. **Exhibit 1** includes ChatGPT conversation records from the 20 million ChatGPT conversation records where the Browse function was invoked to illustrate these steps.

13. **Step 1:** The user submits a query to ChatGPT and the model determines whether to invoke the Browse function based on the query and the model's system prompt instructions indicating when a response would benefit from external information.[2] For example, the model may invoke the Browse function when the user asks about unfamiliar terms or real-time information not included in the model's training data.

14. **Step 2:** ChatGPT rewrites the user's query as relevant search terms and submits the search to a search provider (in this case the Bing API) via the `search` command.[3] An example of this step is shown in the figure below.

[2] N. Feamster Opening ¶¶ 206(a), 210, 233. Under the earliest beta version of Browse, the model would always invoke Browse when enabled. In later versions, the system prompt includes detailed instructions regarding when the Browse function should be invoked in response to a user query.
[3] N. Feamster Opening ¶¶ 212–214, 233, 340.

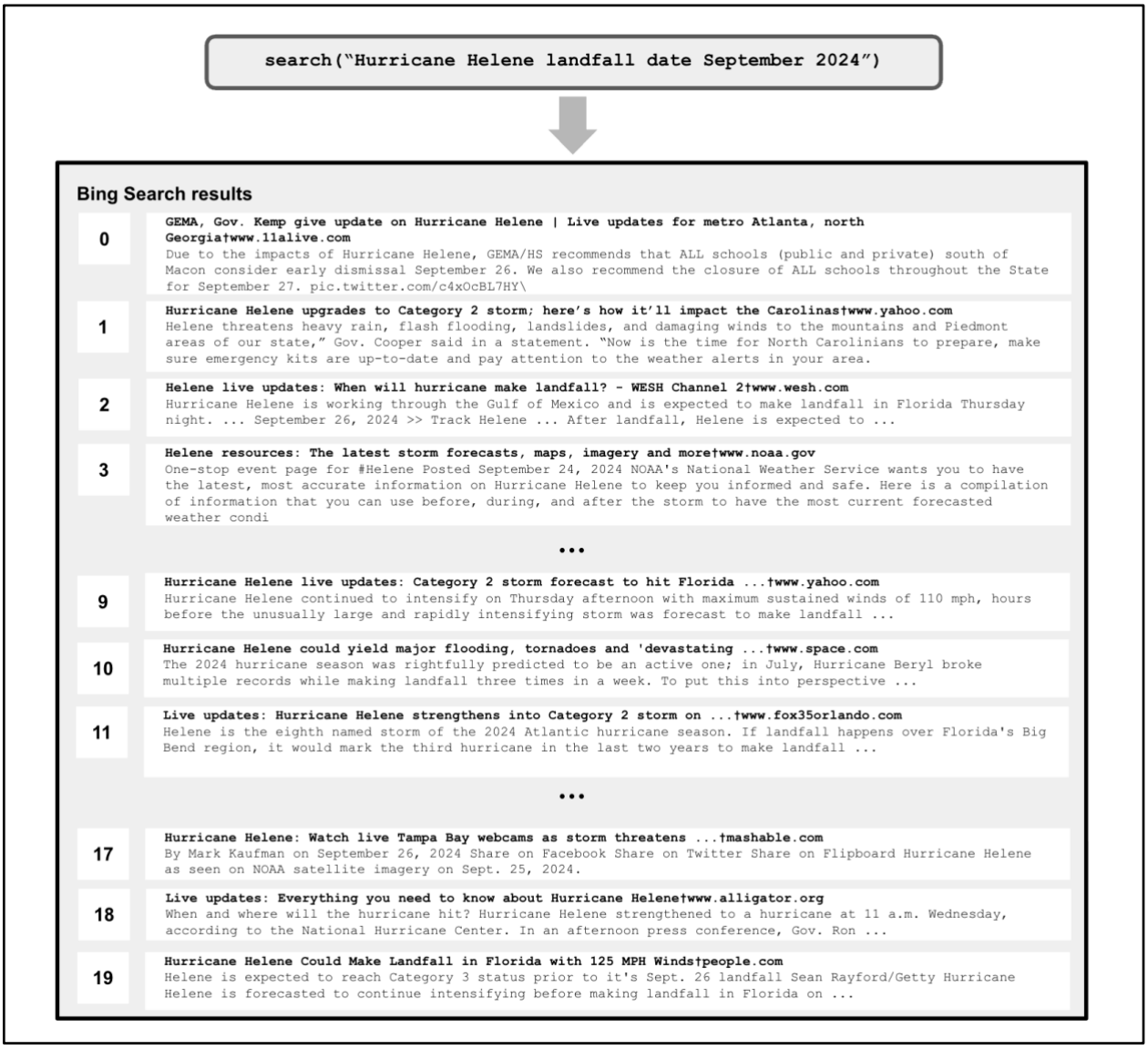
**FIGURE 1:** Step 1 & Step 2




15. **Step 3:** The search provider returns structured search results from its search index that include webpage titles, URLs, and associated text snippets.[4] These search result snippets are not visible to users and are similar to what appears on a Google Search results page.[5] An example of this step is shown in the figure below.

[4] N. Feamster Opening ¶¶ 213–214, 341.
[5] N. Feamster Opening ¶¶ 180, 190 ("Tool Message"), 337.

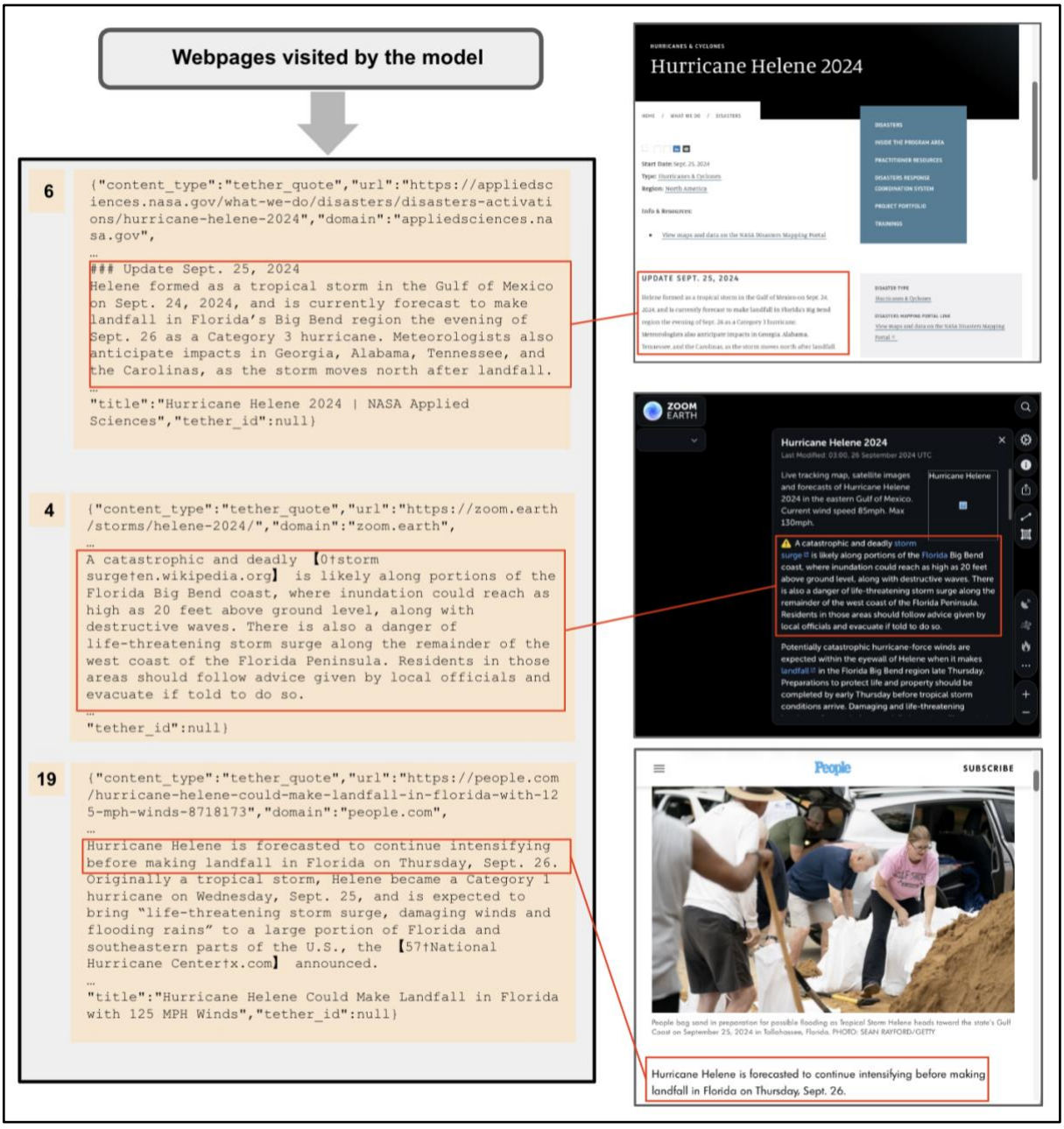
**FIGURE 2:** Step 3




16. **Step 4:** From the search results, the model selects certain webpages for retrieval. After performing a `robots.txt` check, *see infra* **Section D.v**, ChatGPT sends a request to the selected webpages for direct retrieval. The model then identifies relevant portions of text from the retrieved webpage and passes that portion of text into the model's context window.[6] An example of this step is shown in the figure below.

[6] N. Feamster Opening ¶¶ 216–222, 338, 342.

**FIGURE 3:** Step 4

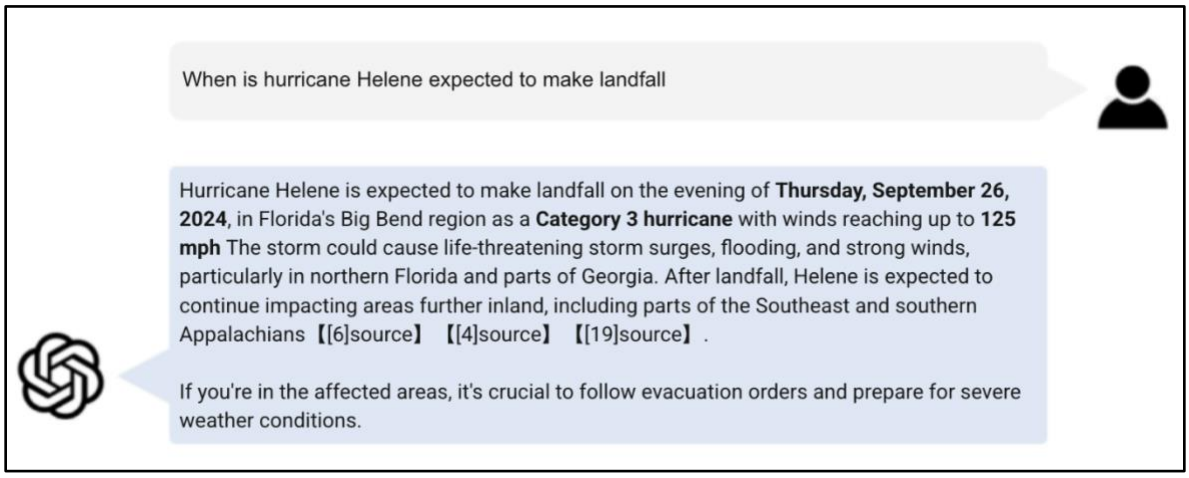



17. **Step 5:** The model then synthesizes the relevant portions of text from the retrieved webpages and generates a response that it displays to the user.[7] ChatGPT also includes citations

[7] N. Feamster Opening ¶¶ 339, 343.

and links to the underlying referenced webpages as part of the response. An example of what a user sees in their conversation with ChatGPT is shown in the figure below.

**FIGURE 4:** Step 5




18. OpenAI designed the Browse function to synthesize and triangulate information from multiple sources. This multi-source synthesis approach progressively strengthened as the tool evolved. OpenAI made structural changes requiring the system to aggregate information across multiple sources, including system prompts instructing the model to access [redacted] in **Step 4**.[8]

19. OpenAI also implemented multiple technical safeguards, including post-training and system instructions, to prevent Browse from reproducing verbatim web content in responses to users and to ensure that the model does not return long summaries of single online sources to users.[9] The OpenAI source code I reviewed also shows that OpenAI constrains verbatim outputs by implementing rules that the model "[redacted]

[8] N. Feamster Opening ¶¶ 218, 238.
[9] N. Feamster Opening ¶¶ 238-239.

[redacted]" and "[redacted]."[10]

20. Based on my analysis of the above steps, I concluded that the purpose of OpenAI's Browse function is to derive and triangulate factual information from a variety of sources to generate a user response. The resulting synthesized ChatGPT output that is visible to the user is different in kind from the limited excerpts of underlying web content that are part of the Browse process.

**ii. Search Result Snippets Returned by Bing**

21. The search result snippets returned by Bing in **Step 3** *supra* are returned from Bing's search index via the Bing Web Search API.[11]

22. The Bing Web Search API allows third-party application developers to help their users find information by providing access to Bing's search index, enabling them to "harness[] Bing's ability to comb billions of webpages, images, videos, and news with a single API call."[12] Microsoft uses its own crawler `Bingbot` to compile Bing's search index.

23. The search result snippets returned to ChatGPT by Bing are similar to what appears on a Google Search or Bing Search results page.[13]

24. I analyzed the search result snippets returned by Bing in the 20 million ChatGPT conversation records produced by OpenAI. Based on my review of these records and my conversation with OpenAI engineer Philip Pronin, I confirmed that the snippets in the conversation records are the exact snippets shown to the model in **Step 3**.[14] I also found that the average snippet

[10] N. Feamster Rebuttal ¶¶ 197–200; N. Feamster Reply ¶ 126.
[11] N. Feamster Opening ¶ 337.
[12] *See* Microsoft, *What is the Bing Web Search API*, MicrosoftDocs, https://github.com/MicrosoftDocs/bing-docs/blob/main/bing-docs/bing-web-search/overview.md.
[13] N. Feamster Opening ¶ 337.
[14] N. Feamster Opening ¶ 337; N. Feamster Supplement ¶ 7.

length for search results including a News Plaintiff's domain was [redacted] characters, which corresponds to an average of [redacted] words per snippet.[15]

25. As explained in **Section D.vi.b** *infra*, I also confirmed that in each instance where a News Plaintiff's domain was included in Bing-returned search results, the associated website permitted access to `Bingbot` at the time the snippet was returned.[16] The snippets included in the search results used by OpenAI thus originated from webpages that allowed Bing to crawl the page, index the contents of their websites, and provide search snippets through Bing's search services. *See supra* **Section C.ii**.

### iii. Third-party Tools are Not Part of OpenAI's Native Browse Function

26. I also analyzed conversation records cited by Plaintiffs' expert Dr. Goldstein in which third-party tools, such as custom GPTs and other third-party plugins, were used to conduct web searches. Based on my review of the conversation record metadata, I confirmed that these conversations did not invoke OpenAI's Browse function.[17]

27. These third-party tools were not created by OpenAI, and OpenAI's native tools associated with the Browse function were not performing the search or actively visiting the Plaintiffs' domains in these cases.

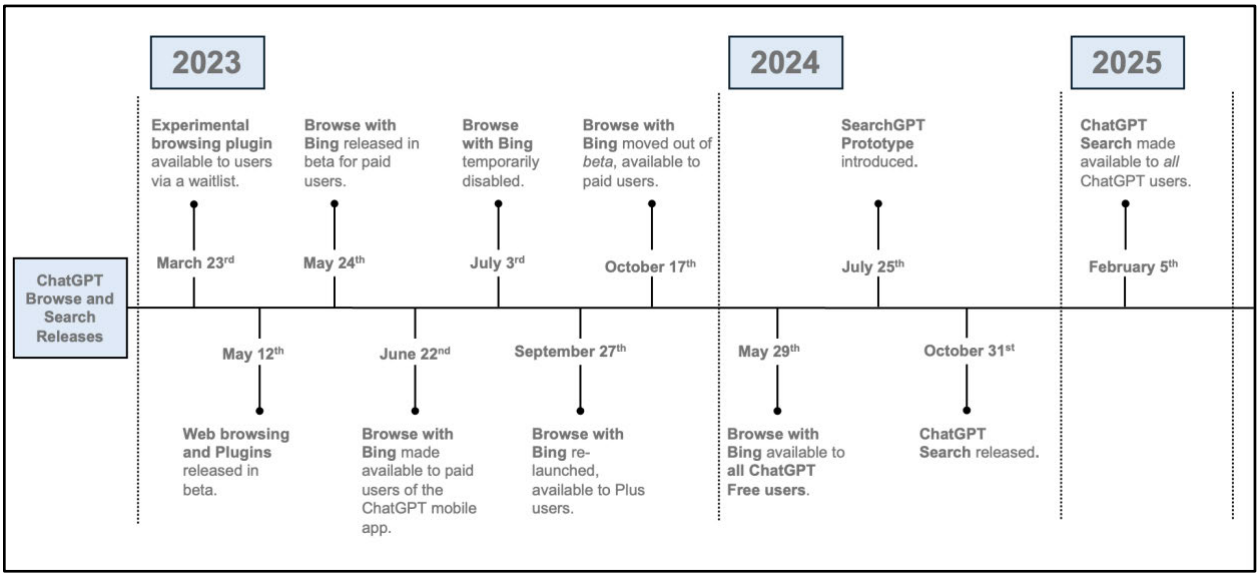
### iv. Timeline of Public Releases for ChatGPT's Browse Function

28. OpenAI's Browse function evolved through several major versions during the relevant period and has been known at various times as "Browse with Bing," "Browse," or "ChatGPT Search."

29. **Figure 5** illustrates the timeline of public releases for ChatGPT's Browse Function.

[15] N. Feamster Reply ¶¶ 117–118.
[16] N. Feamster Reply ¶71; N. Feamster Rebuttal ¶210.
[17] N. Feamster Rebuttal ¶ 237, N. Feamster Reply ¶¶ 106–107.

**Figure 5: Timeline of Public Releases for ChatGPT Browse Functionality**



## D. Overview of Web Crawling and the Robots Exclusion Protocol

### i. Web Crawling and Scraping Are Essential for Internet Functionality and Widely Practiced

30. Web crawling and scraping are well established practices fundamental to the ordinary functioning of the modern Internet. These foundational technologies power a variety of online systems, enabling them to access vast amounts of up-to-date information across the Internet. In my experience both as a researcher who has studied web technologies extensively and as someone who has built bots professionally, I can attest that web crawling and scraping are ubiquitous and essential to countless Internet services that users rely on every day.

31. Web crawling involves systematically following links across the Internet to discover and retrieve webpages, often for the purpose of building a search index. In contrast, web scraping focuses on extracting specific data from those webpages. In simplified terms, a crawler determines *which webpages to visit* by indexing them, while a scraper determines *what information to extract* from the webpages. In practice, a single system may combine both functions.

32. Web crawling and scraping are conducted by automated systems known as "bots." Bots were central to the development of the Internet, as early search and indexing services relied on bots to discover and organize online content and make the expanding Internet searchable and accessible to users. To this day, bots that crawl and scrape the Internet play a significant role across many industries and sectors, including cybersecurity and threat intelligence, e-commerce, news publishers, government agencies, and academic research institutions.

33. Scraping by news outlets is also a longstanding journalistic practice, and News Plaintiffs themselves recognize the value of web scraping and employ these techniques in their own operations. For example, The New York Times and the Chicago Tribune have publicly released web-scraping programs designed to send requests to third-party government websites and retrieve

information for journalistic purposes. I reviewed the source code of several such scrapers and confirmed that they are used to collect public data from government websites for newsroom products such as election pages, campaign finance reporting, and law enforcement data.

34. Automated crawling and scraping conducted by bots are essential to maintaining the discoverability, organization, and accessibility of online information. Without these practices, the Internet would not function as a searchable, structured ecosystem of information.

**ii. The Development and Formalization of the Robots Exclusion Protocol**

35. Implemented via `robots.txt` files, the REP is the longstanding and standard approach for website owners to express preferences as to which web crawlers, scrapers, and other bots may access resources contained in their websites.

36. The REP emerged in 1994 as a protocol that would allow website owners to instruct bots on how to access their websites. Using this system, early website administrators could easily guide bots away from certain pages on their websites using a simple text file.

37. The REP was widely adopted within months of its initial proposal in 1994 and was then officially documented by the Internet Engineering Task Force (IETF) in a 1996 Internet-Draft.[18] The IETF is an international standards organization composed of computer scientists and other technical experts that publishes technical standards on widely used Internet protocols.[19]

38. The REP was used as a widespread convention for more than 30 years. In 2022, the IETF officially formalized the REP as RFC 9309.[20]

[18] Internet-Drafts are working documents which are created during the IETF's development of a specification. *See Internet Drafts*, Internet Engineering Task Force, https://www.ietf.org/participate/ids/.
[19] *See Introduction to the IETF,* Internet Engineering Task Force, https://www.ietf.org/about/introduction/. *See* Heather Flanagan (ed.), *Fifty Years of RFCs, RFC 8700*, Internet Architecture Board (Dec. 2019), https://www.rfc-editor.org/rfc/rfc8700.html.
[20] The IETF's technical standards are known as "Requests for Comments" or RFCs. *See Introduction to the IETF,* Internet Engineering Task Force, https://www.ietf.org/about/introduction/.

### iii. Implementing a Robots.txt File

39. To implement the REP, a website owner creates a `robots.txt` file, which is a publicly accessible text file hosted by the website. In the file, the website administrator can specify preferences for which pages throughout the corresponding website individual bots are allowed to access, thereby creating a set of specific guidelines for the bots to follow.

40. The bot itself is identified in the `robots.txt` file via its associated user agent string—the equivalent of a bot "name tag" used to identify the user or organization operating the bot. The website administrator uses the commands "allow" and "disallow" in the `robots.txt` file to indicate which web resources and objects on the server the bot can access. A `robots.txt` file operates on an opt-out principle; if there is no "disallow" command applicable to a user agent, the website is understood to allow that user agent to access the website by default.

### iv. News Plaintiffs' Adoption of the Robots Exclusion Protocol

41. I conducted an archive search using the Internet Archive Wayback Machine and confirmed that many of News Plaintiffs' `robots.txt` files have been maintained for over 20 years.

42. For example, I confirmed that The New York Times, New York Daily News, Chicago Tribune, Mercury News, OC Register, CNET, and ZD Net were aware of and adopted the REP in 1999.

43. I also confirmed that the Sun Sentinel, Pioneer Press, Orlando Sentinel, Mother Jones, and Baby Center were aware of and adopted the REP in 2001.

44. Other News Plaintiffs' domains including The Intercept, Lifehacker, Mashable, PC Mag, Diabetes Daily, Dice Breaker, and Everyday Health maintained `robots.txt` files before 2010.

45. CIR has maintained a `robots.txt` file for `centerforinvestigativereporting.com` since 2011 and for `revealnews.org` since 2015.

46. Appendix F of my Opening Report, which details the current and historical `robots.txt` files maintained on the news outlet websites owned and operated by the News Plaintiffs, is attached as **Exhibit 2**. As demonstrated in Appendix F, the overwhelming majority of these files have been continually updated and edited.

47. Based on News Plaintiffs' practice of actively managing the preferences expressed in their `robots.txt` files, it is reasonable to assume that if a Plaintiff's `robots.txt` file permitted one of OpenAI's widely publicized bots to access the corresponding website, that file reflected a knowing decision not to disallow access.

### v. OpenAI's Implementation of the Robots Exclusion Protocol

48. OpenAI has long recognized `robots.txt` as a mechanism for website owners to communicate their preferences regarding the scraping of their content, and OpenAI publishes the user agent identifiers `ChatGPT-User` and `OAI-SearchBot` associated with Browse to make it easy for website owners to manage access by those bots through their `robots.txt` files.

49. Responsible bot operators use descriptive user agent strings to identify themselves. When OpenAI announced the "Browse" feature in March 2023, it disclosed `ChatGPT-User` as the user-agent string for Browse and noted that website owners could disallow `ChatGPT-User` in their `robots.txt` files to block crawling.[21] Similarly, OpenAI announced `OAI-SearchBot` in connection with the 2024 rollout of the ChatGPT Search product as the bot used to automatically

[21] N. Feamster Opening ¶¶ 178, 243.

discover and retrieve content for that product.[22] The user agent identifier `OAI-SearchBot` was made public on July 25, 2024.[23, 24]

50. OpenAI designed Browse so that its bots did not fetch content from websites that indicated a contrary preference via their `robots.txt` files.[25] To understand this `robots.txt` verification process, I reviewed ChatGPT conversation records where Browse was invoked and source code produced in this case.

51. Before sending a request to a relevant webpage for direct webpage retrieval during the Browse workflow, OpenAI performs a `robots.txt` check.[26] If the `robots.txt` file for a webpage includes a "disallow" for the operative user agent associated with the Browse function, the user agent is configured, by default, not to request content from the webpage.[27]

### vi. OpenAI's Compliance with robots.txt directives in Conversations Where a News Plaintiff's Domain Appears

52. To analyze OpenAI's compliance with `robots.txt` directives in conversations identified by News Plaintiffs' experts as containing News Plaintiffs' content, I analyzed the relevant conversation IDs from the 20 million ChatGPT conversation records provided to me by OpenAI and the historical `robots.txt` files for the relevant domains.

---

[22] N. Feamster Opening ¶¶ 177, 246.
[23] *See* OpenAI, *Overview of OpenAI Crawlers*, OpenAI Developers, https://developers.openai.com/api/docs/bots (first documentation of the user agent string); *see* Danny Goodwin, *OpenAI starts testing SearchGPT prototype, here's what it looks like*, Search Engine Land (July 25, 2024, 2:42 PM), https://searchengineland.com/searchgpt-launches-444399 ("OpenAI has a new crawler, OAI-SearchBot."). OpenAI publicly updated its description of the `OAI-SearchBot` user agent no later than November 5, 2024. *See* OpenAI, *Overview of OpenAI Crawlers, OpenAI Developers*, https://developers.openai.com/api/docs/bots; *see also* BKA Content, *The FASTEST Way to Get Your Site Indexed in Chat GPT Search*, at 0:19-1:10 (YouTube, Nov. 5, 2024), https://youtu.be/4ALnYSt7828?si=jB1PWDV4XFFF2mLS (showing a screenshot of OpenAI's Crawlers documentation in a video dated November 5, 2024).
[24] N. Feamster Opening ¶243–244.
[25] N. Feamster Opening ¶¶ 309, 316.
[26] *See* N. Feamster Opening ¶¶ 220, 249; N. Feamster Rebuttal ¶ 224.
[27] N. Feamster Opening ¶ 316; N. Feamster Rebuttal ¶¶ 186–189.

53. In conducting my analysis, I consulted the designated URL fields in the conversation records to verify the exact URLs returned by the Bing API or fetched by OpenAI in order to limit my analysis to conversations in which the Browse function was invoked—rather than relying solely on the appearance of domain names in the text. I also distinguished between conversation records that merely show News Plaintiffs' content appearing in search results returned by the Bing API and conversation records showing that the model actually quoted Plaintiffs' content, whether from a search result snippet returned by Bing or from content obtained through a direct visit to the webpage.

54. I analyzed the 11,015 conversations News Plaintiffs' experts identified in which OpenAI's Browse function was invoked.[28] In total, I identified 228 conversations in which the Browse function was invoked where ChatGPT accessed a webpage associated with a News Plaintiff's domain. I also confirmed that in 10,638 of the identified conversations, OpenAI received a Bing search result snippet from a News Plaintiff's domain that it did not show to the user and did not even use in responding to the user. Finally, I found 149 conversations in which a News Plaintiff's snippet appeared in search results returned by Bing and was used to inform the response to the user without Browse ever clicking on or accessing the webpage. I will address each of these instances of News Plaintiffs' content appearing in the Browse conversation records in turn.

**a. Conversations where Browse Accessed a News Plaintiff's Webpage**

55. From the 228 conversations identified by News Plaintiffs' experts in which Browse accessed a webpage associated with a News Plaintiff, there are 5 conversations where a CIR or

[28] In the 20 million ChatGPT conversation records, there are 248,162 conversations which invoke OpenAI's Browse functionality. *See* Feamster Opening Report ¶ 194. The News Plaintiffs' experts allege that there are 13,972 conversations in which the Browse functionality used content from a News Plaintiff's domain. However, in only 11,015 of these 13,972 conversations does a News Plaintiff's domain actually appear in the search results from OpenAI's Browse functionality.

Intercept domain was visited, 2 conversations where one of the New York Times's or Daily News Plaintiffs' domains was visited, and 221 conversations where a Ziff Davis domain was visited.

56. Examples 1–2 in **Exhibit 1** provide illustrative examples of this behavior where Browse accessed a News Plaintiff's webpage.

57. I also reviewed OpenAI's Browse source code as part of my analysis and observed that the model checks the `robots.txt` file before clicking the webpage and does not visit the webpages it is disallowed from. I further confirmed `robots.txt` compliance by checking the relevant domain's `robots.txt` preferences for OpenAI's Browse-related bots at the time of the visit.[29]

58. Appendix D of my Opening Report attached hereto as **Exhibit 3** shows that none of the visited News Plaintiffs' domains disallowed OpenAI's user agents for Browse—`ChatGPT-User` or `OAI-SearchBot`—as of the latest visit to those domains in the 20 million ChatGPT conversation records.

59. In all of these 228 conversations, the operative user agent for the Browse function was permitted by the domain in question at the time of the visit. Therefore, I concluded that OpenAI's Browse function did not visit News Plaintiffs' webpages, except in instances where the websites' `robots.txt` files indicated that OpenAI's bots were allowed to do so.

**b. Conversations Where a News Plaintiff's Domain Only Appeared in a Bing-Returned Search Result Snippet**

60. As explained in **Section C.i** and **Step 4** *supra*, the model does not open every webpage that appears in Bing-returned search results. Many of the snippets contained in the Bing-returned search results appear only in **Step 3** *supra* and are not otherwise used in the Browse process.

[29] N. Feamster Rebuttal ¶ 224.

61. Based on my analysis of the 11,015 conversations News Plaintiffs' experts identified in which OpenAI's Browse function was invoked, I identified 10,638 conversations where the only appearance of a News Plaintiff's domain was a Bing-returned snippet that appeared in search results, but was not used in any subsequent step of the Browse process. This represents over 96% of the instances in which News Plaintiffs claim that Browse made an infringing copy of their work. There are 415 conversations where one of the New York Times' or Daily News Plaintiffs' domains appears only in these search results, 130 conversations where a CIR or Intercept domain appears only in these search results, and 10,470 conversations where a Ziff Davis domain appears only in these search results.

62. Examples 3–4 in **Exhibit 1** provide illustrations of this behavior where a News Plaintiff's domain appears in the Bing-returned search results at **Step 3** but does not appear elsewhere in the conversation record. Again, these Bing-returned search results are never visible to the user, and ChatGPT does not rely on them to formulate the output to the user.

63. These Bing-returned search results have already been filtered in accordance with the `robots.txt` preferences applicable to Bing. Bing's crawlers respect `robots.txt` preferences set by website operators.[30]

64. `Bingbot`, the Microsoft crawler associated with the Bing search index used by the Bing Search API,[31] is the user agent associated with scraping content for search snippets returned to ChatGPT by the Bing API.

[30] N. Feamster Opening ¶ 247.

[31] *See* Microsoft Bing, How to create a robots.txt file, Bing Webmaster Tools, https://www.bing.com/webmasters/help/how-to-create-a-robots-txt-file-cb7c31ec; *see* Microsoft Bing, APIs for Developers Series - *Bing Search APIs*, Microsoft Bing Blogs (Nov. 29, 2016), https://blogs.bing.com/search-quality-insights/November-2016/apis-for-developers-series-bing-search-apis.

65. A website that permits crawling by `Bingbot` allows Bing to use webpage content for its search function, including the Bing Search API, which encompasses search results and related snippets.[32]

66. I analyzed the `robots.txt` files for News Plaintiffs' domains and confirmed that, for every Bing search snippet containing News Plaintiffs' content, the associated website permitted access to `Bingbot` at the time the snippet was returned.

67. Therefore, I concluded that in every identified conversation where a News Plaintiff's domain appears in Bing-returned search results but nowhere else in the Browse workflow, the Plaintiff's `robots.txt` preferences were respected.

**c. Conversations Where Only Search Result Snippets from News Plaintiffs' Domains Were Used**

68. Out of the 149 conversations in which a News Plaintiff's snippet appeared in search results returned by Bing and was used in the Browse workflow without accessing the webpage, 11 conversations related to a New York Times or Daily News domain, 3 conversations related to a CIR or Intercept domain, and 135 conversations related to a Ziff Davis domain.

69. In each of those 149 conversations, the associated website permitted access to `Bingbot` at the time the snippet was returned. Therefore, the snippets relied on by ChatGPT originated from content that the associated website had already authorized Bing to collect and distribute in public search results and via its API.[33]

[32] *See* Microsoft Bing, Bing Webmaster Guidelines, Bing Webmaster Tools, https://www.bing.com/webmasters/help/webmaster-guidelines-30fba23a (explaining that allowing Bingbot in a webpage's robots.txt file means that it can appear in Bing search, Copilot experiences, or grounding API results).
[33] N. Feamster Reply ¶ 71.

### vii. Server-Side Blocking as a Mechanism to Block Crawlers

70. Separate from `robots.txt`, website operators have other technical controls available to manage whether bots can access their content. Among the most effective is server-side blocking, also known as "hard blocking," which denies users and entities (such as bots) access to website resources or content based on specific access control policies.

71. Website operators can configure their server or content-delivery network to reject automated requests based on the request's source IP address or user-agent string. A website that employs this "hard block" against a particular source IP address or user-agent string will return an error, such as "Forbidden," when the blocked source IP address or user-agent string sends a request to the website.

72. Responsible bot operators frequently publish the IP addresses they use to enable website operators to verify where bot traffic is coming from and make it easy for administrators to implement IP address-based filtering if they choose to block traffic from specific entities.

73. OpenAI publishes the IP addresses it uses for its bots to enable website operators to identify incoming network traffic from OpenAI bots, and by extension, to allow website owners to implement blocks based on OpenAI's IP addresses.[34]

74. In preparing my Rebuttal Report, I reviewed News Plaintiffs' expert Ms. Stricchiola's Opening Report, where she stated that the New York Times has employed such a "hard" block against OpenAI bots since [redacted][35] According to her analysis, this block has been [redacted] in preventing OpenAI from accessing New York Times webpage content. Ziff Davis implemented

[34] N. Feamster Opening ¶ 311.
[35] *See* J. Stricchiola Opening Rep. (New York Times, Daily News) ¶275.

similar blocks in [redacted].[36] Although CIR, The Intercept, and the Daily News Plaintiffs could have easily applied this kind of access restriction against OpenAI bots, they elected not to do so.

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct, and that I have personal knowledge of these facts and am competent to testify thereto if called upon to do so.

Dated: September 4, 2026
Paris, France

By: _____
Nick Feamster

[36] *See* J. Stricchiola Opening Rep. (Ziff Davis) ¶158.