Stack Overflow Is Not Dead Yet: Crowd Answers Still Matter

Denis Helic

Graz University of Technology, Graz, Austria

Tiago Santos

Energie Steiermark, Graz, Austria

Abstract

Millions of users visit Stack Overflow regularly to ask community for answers to their programming questions. However, like many other platforms, Stack Overflow consistently struggles with low user retention and declining levels of user contributions to the platform. With the introduction of ChatGPT in November 2022, these ongoing difficulties on Stack Overflow were further magnified, as many users moved toward ChatGPT for programming help. In this paper, we build upon recent research on this phenomenon by analyzing the transformation of user-generated content on Stack Overflow during the post-ChatGPT period. Specifically, we analyze two years of Stack Overflow data and fit multiple causal regression models to estimate the effect of ChatGPT on the length and difficulty of user questions and code examples. We confirm an acceleration of decline in user contributions but find that ChatGPT had a significant positive effect on question and answer length, code length, and question difficulty on Stack Overflow across programming languages. Our results suggest that ChatGPT has effectively raised the bar for questions on Stack Overflow, as users increasingly turn to crowdsourced platforms for help with more complex and challenging problems. With our work we contribute to the ongoing discussion on the impact of tools such as ChatGPT on help-seeking in programming and, more broadly, on collaborative knowledge creation. Our results provide actionable insights for platform operators to support information management and user retention in the aftermath of ChatGPT's launch.

Keywords: Stack Overflow, Q&A platforms, ChatGPT, Help-seeking in programming, Large language models

1. Introduction

Question-and-answer (Q&A) platforms such as Stack Overflow or similar Stack Exchange instances are used by millions of users to discuss a variety of topics and problems. Specifically, Stack Overflow deals with programming and software development questions and attracts currently more than six million active users, making it the largest Web site for software developers seeking help in their daily programming tasks (Abdalkareem et al., 2017; Rahman et al., 2018). Despite its great success, Stack Overflow, like many other social platforms, suffers from typical problems related to management of online communities such as decreasing user numbers and decline in their contributions (Asaduzzaman et al., 2013). Therefore, the launch of ChatGPT—a generative artificial intelligence (AI) tool—in November 2022, has immediately raised the question about the further acceleration of this development (del Rio-Chanona et al., 2024). As such generative AI tools possess both, a notable performance in natural language processing (Chang et al., 2024), as well as a reasonable good code generating capability to solve smaller programming problems (Austin et al., 2021), our research community continues to analyze and discuss the future utility of Stack Overflow (Kabir et al., 2024).

The initial concerns about diminishing utility of Stack Overflow in the era of ChatGPT have been reinforced by several recent research studies. For example, studies by Burtch et al. and del Rio-Chanona et al. suggest that ChatGPT is associated with an accelerated downward trend in the number of questions and answers on Stack Overflow (Burtch et al., 2024; del Rio-Chanona et al., 2024). Moreover, two other works show that generative AI models can

compete and sometimes even outperform Stack Overflow answers on several tasks including resolution of compiler errors (Widjojo and Treude, 2023) or solving privacy-related problems (Delile et al., 2023). Such results suggest that ChatGPT may become a *disruptive technology* with a strong potential to sustainably change software development practices including help-seeking in programming. For example, in a study of early ChatGPT adopters, Haque et al. (Haque et al., 2022) found a strong user expectation that ChatGPT will indeed disrupt common software development processes similarly to how ChatGPT is expected to transform the current practices in other domains such as education (García-Peñalvo, 2023), healthcare (Varghese and Chapiro, 2024), or scientific publishing (Gao et al., 2023).

In this manuscript, we aim to shed light on the transformative potential of ChatGPT on Stack Overflow. In particular, we build upon previous studies on accelerated decline in contributions to Stack Overflow (Burtch et al., 2024; del Rio-Chanona et al., 2024) by asking the question about the effect of ChatGPT on the content of those contributions. Specifically, we ask whether the introduction of ChatGPT has a profound effect on the difficulty levels of the user questions post-ChatGPT. We hypothesize that the Stack Overflow users may turn to ChatGPT more frequently for answers to smaller-scale and elementary programming problems, but that the Stack Overflow community still might play a major role in finding answers to larger-scale and advanced programming questions. To answer our research question, we analyze two years of Stack Overflow data (around six millions of user posts), including all questions and answers six months before the introduction of ChatGPT as well as six months after that event. In addition, we also include the data from the same time period one year before as a control group. Then, we extract a variety of features such as question length or the length of code examples from our collected data. Moreover, using a pretrained large-scale neural embedding model for programming questions and code as well as a ground truth dataset with three difficulty levels for programming questions, we first embed all Stack Overflow questions and then construct a machine learning model to predict the difficulty level of those questions. To control for thematic categories of questions, we divide our data according to the user-created tags into question related to python, java, or web development and repeat all the calculations for each of those groups. Lastly, to estimate the effect of ChatGPT, as well as the short-term and long-terms trends in ChatGPT effect on our content features, we fit a series of causal regression models while controlling for the thematic group and general temporal trends.

While we confirm the previous results on the shrinking volume of user contributions to Stack Overflow, we find that ChatGPT has a differential effect on the content of those contributions. Specifically, we find a significant positive effect of ChatGPT on the question and answer length, code length, as well as the question difficulty. For example, expressed as a percentage of the standard deviation of the given quantity, we observe a 6% increase in question length and a 5% increase in answer length, six months after the ChatGPT launch. In addition, we find a strong, consistent, and positive trend of the ChatGPT effect on those quantities over the whole period, i.e., in short-term (a few weeks after the launch) as well as long-term periods (a few months after the launch). Moreover, our results are robust to disaggregation of the questions according to the user tags. For instance, we find a 21% increase in length of *python* code examples or an 11% increase in *java* question difficulty at the scale of individual standard deviations at the end of our observation period. However, we find that the effects are stronger and more stable for tags with larger numbers of questions such as *python* or *web* development. Our large-scale results suggest that Stack Overflow community experiences a sustained behavioral change in the aftermath of the ChatGPT start. We argue that due to ChatGPT, users ask the Stack Overflow crowd fewer of elementary and more of advanced programming questions. While there are less questions to Stack Overflow in total, these questions shift towards higher difficulty levels, and the crowd answers to those questions are still important—Stack Overflow is far from being dead, evolving further as a consequence of the new technological advancements such as ChatGPT.

With our work we contribute to the ongoing discussion on the effect of generative AI tools, here ChatGPT, on software development practices and help-seeking in programming. Going beyond software development, we also frame our work within the broader discussion on the consequences of modern AI on collaborative knowledge creation, as such advanced tools, capable of generating text and code, could change practices on other knowledge platforms not limited to Q&A platforms such as Wikipedia. Our results provide actionable information for platform operators that can help in information management and organization, as well as in onboarding and retaining users. Moreover, we also give initial insights into the drifts of the user content post-ChatGPT that may be helpful for training and education in programming, as well as for training new versions of such AI models.

2. Related Work

ChatGPT and software development. Technically, ChatGPT belongs to a class of large language models (LLM), a particular type of generative AI tools. LLMs comprise billions of parameters in deep neural network architectures that are trained and fine-tuned on huge textual datasets. Once trained, users interact with the LLMs by writing prompts, and LLMs generate corresponding answers in response to those prompts. For a more in-depth information and an overview of architectures, models, training, fine-tuning, and prompting strategies for a wide range of LLMs, we refer an interesting reader to the paper by Naveed et al. (Naveed et al., 2025).

Going beyond natural language processing (Chang et al., 2024), ChatGPT also demonstrated its remarkable capabilities on various tasks in diverse domains (Teubner et al., 2023) including education (García-Peñalvo, 2023), medicine (Chow et al., 2023; Heng et al., 2023), or even research (Gao et al., 2023). Moreover, in the area of software development, LLMs such as ChatGPT and Copilot have received a lot of attention from practitioners and researchers, in particular for the task of code generation (Liu et al., 2023). For example, LLMs demonstrated outstanding code generation performance on popular code completion benchmarks (Austin et al., 2021; Chen et al., 2021), even further improved by instruction fine-tuning (Luo et al., 2023). Similarly, recent research also demonstrated the ability of LLMs in locating bugs (Jesse et al., 2023) or resolving simple issues from github (Jimenez et al., 2024). While these initial results potentially suggest a range of useful applications of LLMs in programming practice, several other studies caution about potential issues related to such applications. For example, Siddiq et al. (Siddiq et al., 2022) criticize poor quality of the generated code that frequently includes too long methods or code duplicates. Regarding code security, Perry et al. (Perry et al., 2023) found that the code generated by AI is typically less secure, while the work by Sandoval et al. (Sandoval et al., 2023) suggested that code written in collaboration between programmers and LLMs is generally less secure, but the difference to the control group (programmers only) is not substantial. In another line of research, Vaithilingam et al. (Vaithilingam et al., 2022) evaluated the usability of LLMs for code generation and found that the LLM frequently provided useful starting points and saved time for searching information online, but at the same the tool usage was associated with understanding difficulties and reduced task-solving efficiency.

In this paper, we analyze another aspect of software development that may have been affected by LLMs, that of collaborative knowledge creation and sharing in software domain. Hence, we ask the question whether such advanced tools, capable of generating code and answering programming questions, have changed online community practices on large software-related Q&A platforms. For example, users may increasingly turn to ChatGPT as a starting information seeking activity (Vaithilingam et al., 2022), which may lead to a decline in knowledge sharing on the platforms as the first evidence suggests (del Rio-Chanona et al., 2024). Nevertheless, while ChatGPT may support users in simple or programming tasks easily solved with a quick lookup in vast software knowledge databases, we aim to investigate whether users still may seek programming help from the community in case of more complex coding problems.

Software knowledge platforms and ChatGPT. Currently, Stack Overflow is the most successful help-seeking online community, used by a large number of programmers to ask questions on various programming topics (Abdalkareem et al., 2017; Rahman et al., 2018). Several studies investigated the reasons for Stack Overflow's popularity suggesting quick response times (Mamykina et al., 2011), high quality answers to conceptual questions (Treude et al., 2011), potential for usable code snippets (Yang et al., 2016), or possibility to ask question from a broad range of topics and types (Allamanis and Sutton, 2013), as potential factors contributing to the Stack Overflow success. However, recently Stack Overflow experienced decreasing numbers of users, questions, and answers (del Rio-Chanona et al., 2024) and the community started analyzing the potential reasons for this development. For example, Stack Overflow community has been recognized as a community that is unwelcoming to the newcomers[1], resulting in problems of attracting new users. As the online communities need to have a healthy mix of experienced, expert, and novice users for sustainable activity levels (Santos et al., 2019a,b), this caused a set of measures by the Stack Overflow operators to promote more welcoming culture including, for instance, a new user badge (Santos et al., 2020). Also, toxicity and negative sentiment of a substantial amount of answers on Stack Overflow (Asaduzzaman et al., 2013) was found to be related to the problem of retaining less experienced users. Moreover, Calefato et al. found that to enhance their chances of becoming a useful answer users need to carefully frame and pose their questions (Calefato et al., 2018).

---

[1]https://stackoverflow.blog/2018/04/26/stack-overflow-isnt-very-welcoming-its-time-for-that-to-change, last retrieved on September 06, 2025.

From the technical perspective, the quality of code examples from Stack Overflow has been also analyzed in past research. For example, Yang et al. (Yang et al., 2016) found that the majority of code examples in answers can not be compiled. Similarly, Zhong et al. (Zhong and Wang, 2024) analyzed the API use in the code examples from Stack Overflow answers and found that a substantial fraction of these examples violated the rules on how to write API calls. Moreover, the study by Fischer et al. (Fischer et al., 2017) investigated the security risks from the Stack Overflow code examples and found a substantial amount of examples containing insecure code snippets. While these studies suggest caution when using Stack Overflow and the code examples from the community, the value of this collaborative knowledge platform (Storey et al., 2010) as a support tool for software development (Vasilescu et al., 2013) is widely recognized. Moreover, this knowledge has been also utilized to build support tools for developers. For example, de Souza et al. (de Souza et al., 2014) presented an approach based on the "crowd knowledge" from Stack Overflow to recommend information to the developers supporting their current activities. In similar approaches, the researchers have used the vast Stack Overflow knowledge database to extract code and corresponding textual descriptions (Yin et al., 2018), or to summarize source code for future references (Iyer et al., 2016; Kou et al., 2023).

Apart from already mentioned studies on the interplay between Stack Overflow and ChatGPT (Burtch et al., 2024; Delile et al., 2023; del Rio-Chanona et al., 2024; Kabir et al., 2024; Widjojo and Treude, 2023), several other studies focused on a particular use case of LLMs as developer's assistant, a role typically occupied by platforms such as Stack Overflow. These studies found empirical evidence for an increased developer productivity while using LLMs (Peng et al., 2023; Ross et al., 2023), the ability of LLMs to act as pair programmers for expert developers (Moradi Dakhel et al., 2023), or acceptable code quality (Xu et al., 2022; Yetistiren et al., 2022). Such capabilities combined with the immediate response, infinite repeatability, and possibility to interactively refine questions could induce dramatic drops in the user-contributed content on software Q&A platforms, effectively also reducing the training data for future LLMs. This situation caused a prompt action (within the first week since the ChatGPT launch) from the Stack Overflow operators, banning LLMs when posting content [2].

In this paper, we build upon those studies by further investigating the interplay between ChatGPT and collaborative knowledge platforms in the domain of software development. While previous studies identified substantial changes in Stack Overflow following the ChatGPT launch such as decline in the contributed content (Burtch et al., 2024; del Rio-Chanona et al., 2024) or increase in user tenure and the text complexity as measured by the word lengths (Burtch et al., 2024), in our study we also analyze the contributed code examples and ask the question whether the community shifted towards more difficult questions due to the availability of LLMs.

ChatGPT as a disruptive technology. Since the launch of ChatGPT in November of 2022, there was an active public and scientific discussion of its potential disruptive consequences on knowledge creation, education, software development, medicine, or entertainment industry. For example, Garcia (García-Peñalvo, 2023) discussed the potential improvements or disruption of educational practices through ChatGPT concluding that ChatGPT will induce enduring changes in education by requiring from educators to learn and adopt this new technology in their teaching practice. Moreover, introduction of ChatGPT generated a lot of discussion in the academic community on the ability of ChatGPT and similar technologies to write scientific papers and abstracts. For example, Gao et al. (Gao et al., 2023) compared the ChatGPT generated abstracts with the genuine scientific abstracts and found that both automatic and human detectors can detect a large proportion of ChatGPT generated abstracts albeit not all of them. Hence, the authors concluded that journals and publishers will need to adopt new editorial and reviewing policies to deal with this new situation. In other scientific domains, such as medicine or healthcare, researchers investigated and discussed the effects of ChatGPT on medical education (Heng et al., 2023), medical chatbots (Chow et al., 2023), or the transformative influence on healthcare (Varghese and Chapiro, 2024). These and similar studies draw similar conclusions: while ChatGPT and similar LLM technologies have a potential as a disruptive technology to improve processes, practices, and the quality of interactions, there are also concerns related to the quality, correctness, transparency, fairness and ethical issues related to a wide adoption of this new technology (Guo et al., 2023; Shen et al., 2023; Ye et al., 2023).

The discussion of ChatGPT and LLMs as a disruptive technology, taps into a broader economic theory of disruptive technology and disruptive innovation, introduced by Christensen in his 1997 book "The innovator's dilemma: when new technologies cause great firms to fail" (Christensen, 1997). The original theory defined a disruptive technology as a technology that is inferior in the main attributes of the mainstream technology but concentrates on alternative

---

[2]https://meta.stackoverflow.com/questions/421831/policy-generative-ai-e-g-chatgpt-is-banned, last retrieved on September 06, 2025.

Table 1: *Dataset statistics.* We show the number of posts including questions and answers in our two years observation period before and after parsing the HTML content of the posts. We observe a downwards trend in the number of posts in the second year.

	Period	Total	Questions	Answers
Initial dataset	Both	$6,000,198$	$2,642,840$	$3,352,591$
Dataset after parsing HTML	Both	$5,997,763$	$2,642,730$	$3,352,332$
	21/22	$3,283,819$	$1,438,850$	$1,843,515$
	22/23	$2,713,944$	$1,203,880$	$1,508,817$

attributes that the mainstream technology neglects. Nevertheless, as the new technology matures, it surpasses the dominant technology, first in specific markets, and then potentially also in other more general markets as well. The theory of disruptive technology was later extended to the concept of a disruptive innovation, which is not only focused on technology but may include, among others, disruption in business models or products (Christensen and Raynor, 2003; Hang et al., 2015; Markides, 2006). A comprehensive survey on the economic aspects of the theory of disruptive technology and innovation can be found in (Si and Chen, 2020).

In this paper, we concentrate on the impact and the (disruptive) change that introduction of ChatGPT can potentially have on online knowledge creation communities, in particular, on collaborative software Q&A platform Stack Overflow. While prior work (Burtch et al., 2024; del Rio-Chanona et al., 2024) has already identified a negative quantitative impact of ChatGPT on user contributions on Stack Overflow, we focus on the content related aspects of user posts such as questions difficulty and the complexity of included code examples in the post-ChatGPT period.

3. Descriptive Analysis

Dataset. We study Stack Overflow, the largest question and answering (Q&A) community for seeking help with programming and software development problems. We obtain the data from the official data dumps from September 2008 until March 2024[3]. From these data dumps, we extract all posts from a two years period starting on May 31, 2021 until May 28, 2023. We split this data in two parts: (a) 21/22 period starting May 31, 2021 until May 29, 2022 and (b) 22/23 period from May 30, 2022 until May 28, 2023. Hence, 22/23 period includes six months of data prior to the ChatGPT launch on November 30, 2022 and six months of data after the launch. 21/22 period contains the same time interval one year prior to the ChatGPT launch and serves as the control data for our analysis.

The extracted data contains slightly more than six million ($6,000,198$) posts with $2,642,840$ questions and $3,352,591$ answers. The remaining $4,767$ posts are related to internal communication and administration of the system. The data contains time of the post, title, the content of the post included as HTML code, username, last edit time, score, view count, comment count, the accepted answer and tags (if the post is a question), and several other administrative fields. We start our analysis by parsing the HTML code to extract the textual content as well as the content of *<code>* elements that include verbatim code examples that users post in their questions and answers. After removing posts with invalid HTML code ($2,435$ posts) we have slightly less than six million posts ($5,997,763$). Further, after selecting only question and answer posts by eliminating administrative posts our final dataset includes $2,642,730$ questions and $3,352,332$ answers. In 21/22 period we have $3,283,819$ posts comprising $1,438,850$ questions and $1,843,515$ answers. In 22/23 period we obtain approximately half a million posts less ($2,713,944$) including $1,203,880$ questions and $1,508,817$ answers. We summarize these basic dataset statistics in Table 1.

Post volume and length. We start by comparing basic weekly statistics of posts between periods 22/23 and 21/22. Period 22/23 is treated with the ChatGPT launch while period 21/22 serves as a control period. In Figure 1 we show the weekly question volume (number of questions posted), and weekly means of question views, question scores and question length, which we compute by counting the number of lines in the post's HTML code. In Figure 2, we show the same statistics for answers, except for answer views, which are not recorded separately but are subsumed in the views of the corresponding questions. For both types of posts we reproduce the results from previous studies (Burtch et al., 2024; del Rio-Chanona et al., 2024) finding a sharp decline in the post volume since the launch of ChatGPT

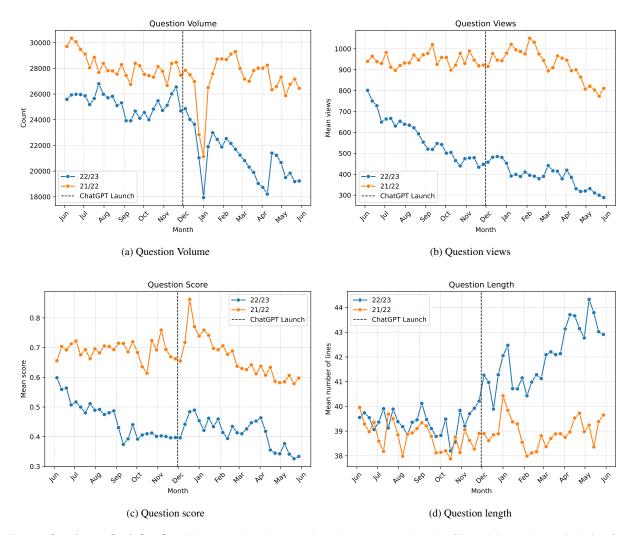[3]https://archive.org/details/stackexchange, last retrieved on September 06, 2025.

Figure 1: **Questions on Stack Overflow.** We compare the volume (a), views (b), score (c), and number of lines (d) in questions on Stack Overflow in 22/23 (May 30, 2022 through May 28, 2023) and 21/22 periods (May 31, 2021 through May 29, 2022). The launch of ChatGPT is in the middle of the 22/23 period (November 30, 2022). Stack Overflow already experienced an ongoing downwards trend in the number of questions, question views, and question scores, even prior to ChatGPT (cf. orange lines, as well as blue lines before the ChatGPT launch in (a), (b), and (c)). Similarly to previous work (Burtch et al., 2024; del Rio-Chanona et al., 2024), we also observe an accelerating negative trend in (a) after the ChatGPT launch (cf. slope of the blue and orange lines post-ChatGPT). We do not observe such an acceleration in the question views in (b) and question scores in (c) but more of a continuing negative trend that already started in the six months period before ChatGPT. However, when comparing the length of the questions between the 22/23 and 21/22 periods, we see a substantial change around the ChatGPT launch in (d). Particularly, while throughout the whole 21/22 period the number of lines in questions is rather stable (orange line), we observe a strong upwards trend in the 22/23 period after the ChatGPT launch (positive slope of the blue line), suggesting a substantial and sustained increase in the question length in that period.

(cf. slopes of blue lines in Figures 1a and 2a). This decline accelerates an already existing negative trend (cf. slope of orange lines in Figures 1a and 2a) in post volume on Stack Overflow suggesting an essential shift in user posting behavior post-ChatGPT. Moreover, the trend remains visible over a long time span and persists until the end of our observation period in May 2023, six months after the ChatGPT launch.

Opposite to question volume, we do not observe an accelerating downwards trend in the mean question views post-ChatGPT (see Figure 1b). In particular, we observe a continuing and strong negative trend that already started prior to the ChatGPT launch (cf. blue line in Figure 1b). This negative trend results in a considerable mean question view drop in the last weeks of 22/23 period (around 300 weekly mean views) as compared to the same time one

(a) Answer Volume
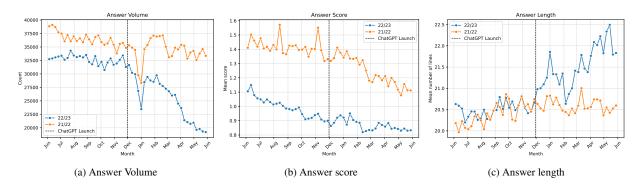
(b) Answer score

(c) Answer length

Figure 2: **Answers on Stack Overflow.** We compare the volume (a), score (b), and number of lines (c) in answers on Stack Overflow between 22/23 period and 21/22 periods. We observe a similar temporal evolution as in questions (cf. Figure 1). In particular, the negative trend in the answer volume accelerates after the ChatGPT start (cf. blue line slope in (a)), while there is no visible change of an already existing negative trend in the answer score in (b). Finally, in (c) we observe an accelerating positive trend in the length of answers after the ChatGPT start.

(a) Top 10 tags overall
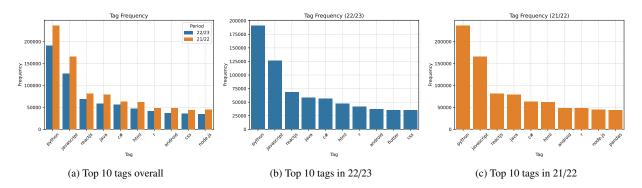
(b) Top 10 tags in 22/23

(c) Top 10 tags in 21/22

Figure 3: **Tags on Stack Overflow.** In (a) we show top 10 tags on Stack Overflow in our two years observation period and compare their volume for periods 22/23 and 21/22. We observe a substantially lower tag volume in the period 22/23 as compared to 21/22, similar to the decreasing question and answer volume. In (b) and (c) we show top 10 tags for both periods individually. While the top seven tags including e.g., *python*, *javascript*, or *reactjs* remain unchanged, there are some fluctuations among remaining tags with *node.js* and *pandas* dropping out of top 10 tags in 22/23.

year before (more than 800 weekly mean views), a drop of around 60% of the last year's weekly average views. We observe a similar behavior in question and answer scores in Figures 1c and 2b—an already existing downwards trend in both question and answer scores continues until the end of the observation period without a visible acceleration post-ChatGPT. This trend results in approximately 50% drop in question scores and 40% drop in the answer scores between the beginning and the end of our observation period.

However, unlike post volume and scores, or question views, the length of both questions and answers undergoes a major and enduring change after the ChatGPT introduction, see Figure 1d and Figure 2c. While both question and answer length exhibit a stable oscillating behavior (around 39 lines for questions, and 20.5 lines for answers) without any visible trends prior to the ChatGPT, both quantities demonstrate a strong, robust, and ongoing upwards trend in six months after the ChatGPT start. Mean question length at the end of the observation period is around 43.5 lines (11.5% increase) while mean answer length is approximately 22 lines (7% increase). This extensive change in post's length combined with dwindling post volume suggests a fundamental behavioral trend on Stack Overflow—while users do not seek help on Stack Overflow as often as before, when they do, they tend to write longer questions.

Tags. On Stack Overflow users can assign up to five unique tags per question for categorization purposes. The tags come from a predefined vocabulary, which more experienced users with enough reputation can extend by defining new tags[4]. To account for possibility of aggregation bias when analyzing heterogeneous data such as Stack Overflow

---

[4] https://stackoverflow.com/help/privileges/create-tags, last retrieved on September 06, 2025.

(a) Web: question volume   (b) Web: question length   (c) Web: lines of code

(d) Python: question volume   (e) Python: question length   (f) Python: lines of code

(g) Java: question volume   (h) Java: question length   (i) Java: lines of code

Figure 4: **Question tags on Stack Overflow.** We categorize questions by their tags (*python* and *java*) and tag groups (*web* group includes *javascript*, *reactjs*, *html*, *node.js*, and *css* tags) and show top three categories by question volume. In particular, we show *web* tags in the first, *python* tag in the second, and *java* tag in the third row. Similar to the aggregate of all questions, we observe an accelerating downwards trend in the question volume in (a), (d), and (g), as well as a substantial positive trend in the length of questions in (b), (e), and (h), confirming the ChatGPT association across the disaggregated categories. In addition, we also observe a similar change towards positive trend in the length of the code examples in (c), (f), and (i) after the ChatGPT launch. We compute the lines of code by parsing the HTML body of the questions and extracting content from *<code>* elements (used to format code examples on Stack Overflow). The positive trend in question and code length is somewhat weaker for smaller tags (*java*) than for the larger tag groups (*web* and *python*). Nevertheless, we observe a similar positive trend in the remaining tag groups, which we do not show here due to limited space. This finding demonstrates that after the ChatGPT launch Stack Overflow users include longer code examples in their questions, suggesting a considerable shift in the problems for which users seek help on Stack Overflow.

questions, we divide the questions into groups according to their tags. In particular, we are interested in analyzing whether an association observed in aggregated data (in our case the association between the ChatGPT launch and the volume or length of the questions) disappears or reverses when data is divided into the underlying groups (Mehrabi et al., 2021) (here, defined by the question tags), a paradox known as Simpson's paradox (Blyth, 1972).

In Figure 3 we show the top 10 tags in our complete observation period, as well as top 10 tags from 21/22 and 22/23 periods separately. The most frequent tag is *python*, followed by two javascript related tags (*javascript* and

reactjs), and *java* tag. In all cases, we observe a skewed tag distribution, which quickly falls off with higher tag ranks. The top 10 tags are quite stable over two periods with two tags dropping from top 10 in 22/23 (*node.js* and *pandas* are replaced by *flatter* and *css*). To further analyze the question categories, we group *javascript*, *reactjs*, *html*, *node.js*, and *css* tags into *web* tag group as they thematically belong to the Web development. In Figures 4a, 4d, 4g we show the question volume of the top three tag groups and tags (*web*, *python*, and *java*) and in all three cases, we observe a similar accelerating downwards trend as for aggregated questions and answers. The trend change is more prominent for larger question groups, i.e., *web* and *python* tags (see Figures 4a and 4d) have a stronger negative acceleration, while the trend change is weaker for *java* (cf. Figure 4g). In addition to these three largest tags groups, we observe a similar relation between the intensity of trend change and the question volume for all other tags from our dataset with smaller groups experiencing a weaker trend change, while larger question volumes generally tend to experience a higher acceleration of an already existing negative trend. Due to the space limitations we do not show here the figures for the remaining tag groups. In summary, we observe the same association between the acceleration of the negative trend in question and answer volume even after disaggregation of the data according to their thematic categories.

We continue and show the question length, measured in the number of lines, in Figures 4b, 4e, 4h. Similar to the aggregated questions and answers, we observe the same association between the ChatGPT start and the question length in all tag groups. After the launch, we observe a substantial upwards tilt in the weekly averages of question length that persists until the end of our observation period. The tilt is more prominent for *web* and *python* tags (Figures 4b and 4e), but it is still clearly visible also for *java* tag (Figure 4h). Similar dependence of the intensity of the tilt on the question volume in a given group is visible in the remaining tags, not shown here due to space constraints. Hence, a clearly visible shift related to an increasing question length post-ChatGPT is also present in all tag groups.

Lines of code. We also analyze the code length of examples included in the Stack Overflow questions. In the dataset, the post body is stored as HTML code, and the code examples are included verbatim within *<code>* HTML elements. Hence, using an HTML parser, we extract all *<code>* elements from the question body and count the lines included in those code snippets. We extract the code length separately for each tag group and plot its temporal development for top three tags in Figures 4c, 4f, and 4i. In all three cases, we observe a similar development as with the overall question length. For larger tags, i.e., *web* and *python*, we see an extensive positive trend in code length after ChatGPT was introduced. In addition, we observe a similar, although somewhat weaker trend for a smaller *java* tag, as well as for the remaining tags. This considerable discontinuity post-ChatGPT suggests a major change in help-seeking behavior for programming tasks on Stack Overflow. While the total number of questions across various thematic question groups further decreases, the length of the questions, as well as the length of code examples consistently and sustainably increases on Stack Overflow. Potentially, this indicates that users increasingly often turn to other channels such as ChatGPT for shorter, and to Stack Overflow for longer programming questions.

Question and code difficulty. Finally, we analyze the difficulty of the Stack Overflow user questions together with the included code examples. To measure the question difficulty we first collect a dataset from LeetCode [5], which is an online platform collecting tasks for programmers' and software developers' training. The platform contains thousands of questions and programming tasks collected from programming courses and technical interviews in the industry. The questions are categorized into topics such as data structures, algorithms, or databases and by difficulty as being easy, medium, or hard tasks. Moreover, questions typically have solutions in several programming languages including *python*, *c++*, or *javascript*. We collect a prepared LeetCode dataset [6] from huggingface [7]. The dataset includes $2,360$ programming tasks with examples for all three difficulty levels together with solutions in *python*, *java*, *c++*, and *javascript*. For each task and a solution language, we create a separate instance containing the title of the question, the description of the task, and the solution in that language, resulting in the final dataset with $9,440$ examples. Second, we use a pretrained text/code encoder transformer model CodeT5 (Wang et al., 2021) that was trained on the CodeSearchNet dataset (Husain et al., 2020), a large collection of text (query-like questions) and functions in several programming languages [8]. With this model, we embed the examples from our LeetCode dataset. In particular, we use CodeT5-Base [9] from huggingface that computes 768-dimensional embeddings of the input text/code content.

---

[5] `https://leetcode.com/`, last retrieved on September 06, 2025.

[6] `https://huggingface.co/datasets/greengerong/leetcode`, last retrieved on September 06, 2025.

[7] `https://huggingface.co/`, last retrieved on September 06, 2025.

[8] `https://github.com/github/CodeSearchNet`, last retrieved on September 06, 2025.

[9] `https://huggingface.co/Salesforce/codet5-base`, last retrieved on September 06, 2025.

(a) Web: medium difficulty	(b) Python: medium difficulty	(c) Java: medium difficulty

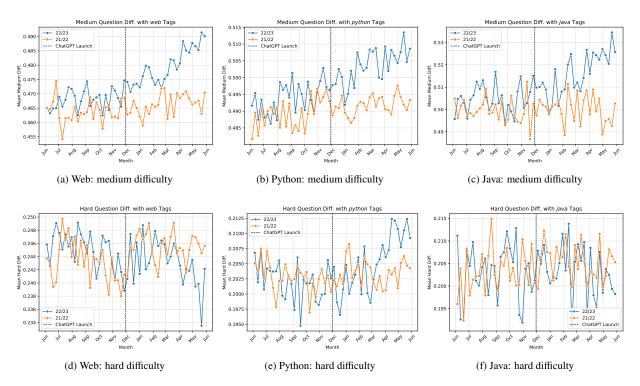| (d) Web: hard difficulty	(e) Python: hard difficulty	(f) Java: hard difficulty

Figure 5: **Question and code complexity on Stack Overflow.** We classify questions by their difficulty as easy, medium, and hard. We first compute embeddings from questions including the code examples by using a pretrained question/code embedding model CodeT5 (Wang et al., 2021). Using a labeled question/code dataset from LeetCode with easy, medium, and hard classes, we train an XGBoost classifier that we apply on the questions from Stack Overflow. The classifier computes the probabilities of questions belonging to given difficulty classes. In the top row, we show weekly means of the classifier probabilities for the medium difficulty on the top three question tag groups including *web* (a), *python* (b), and *java* (c) tags. Similar to the question and code length for those tag groups we see an upward trend in the medium difficultly that accelerates around the ChatGPT launch suggesting a shift in user posting behavior in this period. The same analysis for the remaining tags, indicates, as previously with the question and code length, that smaller tag groups exhibit weaker changes in trends. In the bottom row, we depict the classifier probabilities for the hard difficulty. We do not observe any patterns or trend changes in these plots, as the probability for hard questions seems to develop completely at random, in particular for *web* and *java* tags in (d) and (f). In case of the *python* tag in (e), we observe an upward trend starting in March 2023, already four months after the ChatGPT launch, hence, this change might be associated by some event other than introduction of ChatGPT.

Third, using the computed embeddings as the input features we train an XGBoost classifier (Chen and Guestrin, 2016) using the question difficulty as the classification target. We split the LeetCode dataset in 80% training and 20% test dataset and train an XGBoost classifier using the default parameters[10]. On the test dataset, we achieve ROC-AUC of 0.99 and weighted macro F1-average of 0.95. Fourth, as we achieve high accuracy rates on the test dataset, we retrain the XGBoost classifier on the whole dataset using again the default classifier parameters. Fifth, we embed all questions from our Stack Overflow dataset using the question title, the question body, the tag as an indication of the programming language, and the code examples. Finally, we predict the question difficulty with our XGBoost classifier to obtain the probabilities that a given question belongs to the easy, medium, or hard difficulty class.

We depict the temporal evolution of the predicted question difficulty (as the weekly means of the probability of the medium and hard difficulty) for the three largest tag groups and for both of our observation periods in Figure 5. In the top row (Figures 5a, 5b, and 5c), we show the weekly mean probability for the medium difficulty and in the bottom row (Figures 5d, 5e, and 5f) the weekly mean probability for the hard difficulty. For the medium difficulty, we observe a significant upward trend around and after the ChatGPT start in all three tag groups. This indicates a substantial shift in the user posting behavior post-ChatGPT associated with the difficulty of the questions. This increase in probability of the medium difficulty comes at the cost of the probability of easy questions (which we do not show here) as the

---

[10]`https://xgboost.readthedocs.io/en/stable/python/python_intro.html`, last retrieved on September 06, 2025.

10

probabilities of the hard questions are quite noisy but without any visible trends (cf. bottom row in Figure 5). Hence, on Stack Overflow in the post-ChatGPT period, the probability of medium difficulty questions increases while the probability of easy questions decreases without any visible changes in the probability of hard questions. Potentially, this indicates that the users turn more frequently to other channels such as ChatGPT for easy programming questions, but ask community more often in the case of more difficult questions. However, users seem to further differentiate in their reliance on the community response—in cases of the medium problem difficulty users turn more often to the community, while in cases of the hardest questions no significant difference in behavior is visible.

4. Effect of ChatGPT on Stack Overflow

To quantify the effect of ChatGPT on Stack Overflow we use a difference-in-differences (DiD) regression setup. DiD analysis allows us to estimate the ChatGPT effect on various Stack Overflow metrics while controlling for temporal trends. In particular, we take the 21/22 period as a control group and compare temporal developments of the period 22/23 before and after the ChatGPT launch to that control group. We run our DiD regression for multiple outcome variables including the question and answer length, question and answer scores and views, the length of questions and code examples for all tag groups, as well as question difficulty estimates.

Hence, denoting our outcome variable with Y_i, we model this variable as a linear function of period (P) denoting whether the timestamp of the question is before the ChatGPT launch or after (we use one year prior to the launch, i.e., November 30, 2021 as the event date for the control period), ChatGPT treatment (T) denoting whether the period is 22/23 (treated) or 21/22 (control) and the interaction between P and T. The interaction coefficient is the DiD estimate and it quantifies the change in slope of the linear function of the outcome variable as an effect of the ChatGPT launch. In addition, to account for seasonal and observed temporal trends (cf. Figures 1, 2, 4, and 5), we use the week of the year (W) as a control variable. Our final DiD model is given by:

$$Y_i = \beta_0 + \beta_1 T + \beta_2 P + \beta_3 (T \cdot P) + \beta_4 W + \epsilon_i, \tag{1}$$

where β_1 quantifies the average change in prediction of the outcome variable between the 21/22 and 22/23 periods, β_2 the average change in the outcome variable before and after the launch (or before and after Novemeber 30, 2021 for the 21/22 period), and β_3 is our DiD estimate that quantifies the effect of ChatGPT on a given Stack Overflow metric after controlling for temporal trends. Finally, β_4 quantifies any residual weekly temporal trend in a given metric.

Distributions of the outcome variables. Before computing DiD regressions, we check the distributions of the outcome variables. Similar to other user-generated data, we find highly skewed distributions for several outcome variables such as question and answer length, as well as code examples length across all tag groups. In these cases, majority of questions and answers have small values of the outcome variable and a much smaller portion of questions and answers have large values for those particular quantities. Therefore, we log transform all of these outcome variables to obtain more symmetric distributions closer to a normal distribution. We leave other outcome variables without significant distributional skew such as scores or difficulty probability unchanged. The log transformation of the outcome variable turns these regression models into multiplicative models and changes the interpretation of the coefficients. In particular, coefficients close to zero, e.g., 0.02 can be interpreted directly as the 2% change in the outcome variable for one unit change in the corresponding variable while all other variables are being held constant, with the exact percentage change being equal to $(e^\beta - 1) \times 100\%$ for coefficient values not close to zero. We discuss the particular interpretation of the corresponding regression models directly in the results section.

Regression setup. To properly estimate short-term and long-term effects of the ChatGPT launch, as well as the sensitivity of the exact launch date on Stack Overflow, we adopt the following DiD setup. We fit multiple regression models, always taking all the data from the pre-launch period into regression analysis, i.e., data where $P = 0$, which includes six months of data from May 31, 2021 until November 29, 2021 for the control group and six months of data from May 30, 2022 until November 29, 2022 for the treated group. In addition, we use a sliding window of one month of data in the post-launch period and iterate over the post-launch data starting with November 30, 2021 respectively November 30, 2022 until April 29, 2022 respectively April 29, 2023. In total, we fit 151 regression models for each metric and obtain a daily time series of DiD coefficients starting with the ChatGPT launch and extending for six months in future, allowing us to estimate the short-term effects with data selections closer to the launch, as well as, long-term effects with data selections further from the launch. This "secret weapon" methodology, introduced by
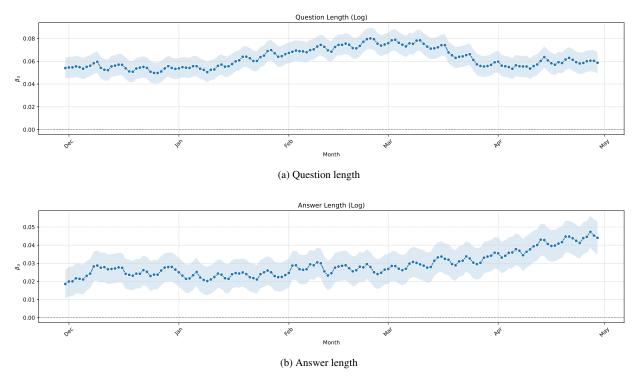
(a) Question length

(b) Answer length

Figure 6: **Effect of ChatGPT on the question and answer length over time.** We fit multiple DiD regressions using periods 22/23 as the treated group and 21/22 as the control group. We fit our models using all the data prior to the ChatGPT launch (or prior to November 30, 2021 for the control group) and one month of data after the launch, starting with the launch date and iterating with a daily sliding window until the end of our observation period. In total, for every outcome variable we fit 151 regression models and plot the β_3 coefficient as our DiD estimate, quantifying the effect of ChatGPT on the particular outcome variable. We accompany the estimates with the confidence intervals ($\pm 2 \times se$) and check for intersection of those confidence intervals with the zero line. We interpret the estimates without intersection as statistically significant effects of ChatGPT on the given quantity. For the question length, depicted in (a), we observe a strong effect that persists over complete six months observation period. Similarly, in (b) we observe a strong positive and continuous effect of ChatGPT on the answer length. We conclude that, in addition to ChatGPT negatively affecting the volume of question and answers on Stack Overflow (Burtch et al., 2024; del Rio-Chanona et al., 2024), it had, at the same time, a profound effect on the user content length, resulting in longer user questions as well as answers from the community.

Gelman and Huang (Gelman and and, 2008), allows us to quantify effect trends rather than mere point estimates. As the final preprocessing step of our regression setup, we standardize all the outcome variables such that the regression coefficients are measured on the scale of the standard deviation of a given metric. This makes the regression results comparable between different outcome variables.

Parallel trend assumption. Difference-in-differences models assume a parallel trend between the control and treatment groups in the period before treatment. We check for this trend visually (cf. Figures 1, 2, 4, and 5, blue vs. orange lines in before ChatGPT periods) and see similar and stable trends in before treatment period. To provide further evidence for parallel trend assumption we additionally fit the following regression model:

$$Y_i = \beta_0 + \beta_1 T + \beta_2 W + \beta_3 (T \cdot W) + \epsilon_i, \tag{2}$$

where T and W denote, as before, treated vs. control group and the week of the year, respectively. We fit this model only with the data from the period before the treatment begins, i.e., data with $P = 0$. Coefficient β_3 estimates the difference in slopes between the treated and the control group, and the test whether $\beta_3 = 0$ provides information on whether trends between two groups are different prior to treatment.

4.1. Regression results

In Figures 6, 7, 8. and 9, we show the results of our DiD regressions for question and answer lengths, question and answer lengths for three largest tag groups, code length of examples from those largest tag groups, and question

(a) Web: question length

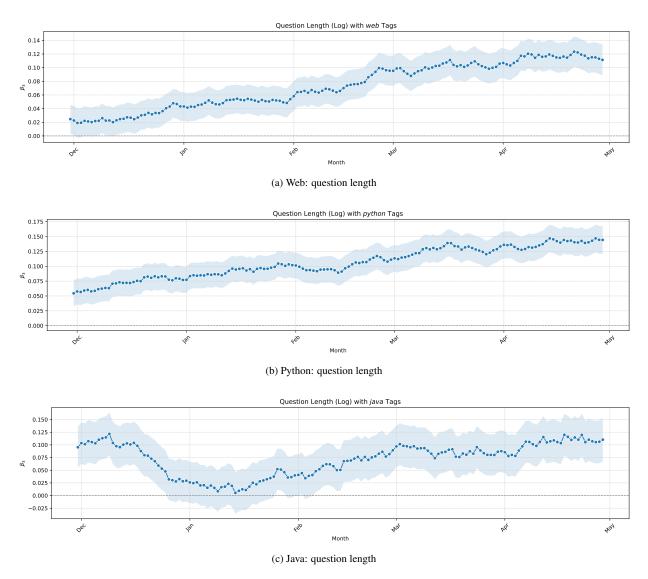(b) Python: question length

(c) Java: question length

Figure 7: **Effect of ChatGPT on the question length over tags and time.** Using the same regression setup as before, we depict the effect of ChatGPT on question length across tag groups. We observe strong positive and lasting effects on the question length for *web* tags in (a), *python* tag in (b) and *java* tag in (c). In particular, the effect sizes at the end of the observation period in May 2023 are 12% for *web* tag, 16% for *python* tag, and 9% for *java* tag of the individual standard deviations of the question length. We also observe a slightly smaller trend for the smallest *java* tag than for large volume *web* and *python* tags. By dividing the data into thematic categories we provide further evidence for a foundational change in the Stack Overflow's Q&A practices since the introduction of ChatGPT by ruling out the possibility of Simpson's paradox (Blyth, 1972), in which an association or effect observed in the aggreagated data disappears once when data is dividied into categories.

difficulty, respectively. In all figures, we show the temporal development of the ChatGPT effect as measured by the interaction coefficient β_3 from Equation 1, together with the confidence intervals (blue shaded region around the DiD estimate) computed as $\pm 2 \times se$, *se* being the standard error of the DiD coefficient estimate. In cases where confidence intervals do not intersect the zero line, the coefficient is statically significant, indicating that the ChatGPT launch had an effect on the given quantity. The coeffcient values can be interpreted as the percentage change (in Figures 6, 7, 8) or as the increase in probability of the outcome variable (in Figure 9) measured at the scale of the standard deviation of that variable. This setup allows for comparison of the effect size between different outcome variables.

Question and answer length. In Figure 6, we observe a positive, significant DiD coefficient over the whole observation period, indicating a persistent positive effect of ChatGPT on the length of the questions and corresponding
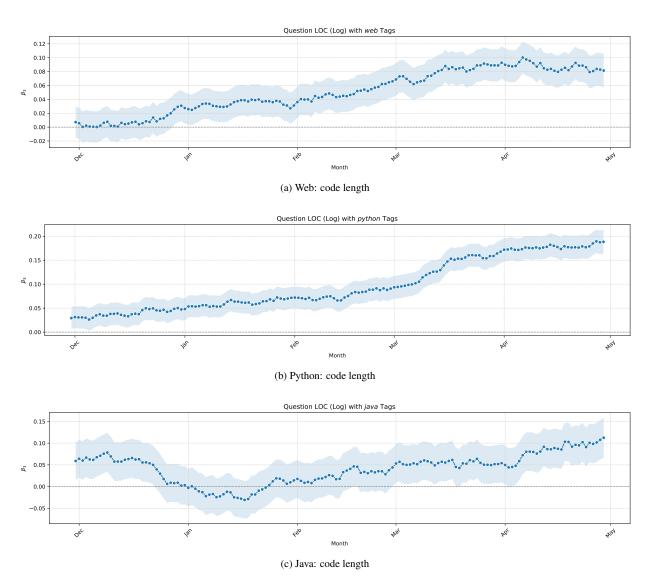
(a) Web: code length

(b) Python: code length

(c) Java: code length

Figure 8: **Effect of ChatGPT on the code length over programming language and time.** Again, utilizing our DiD setup we estimate the effect of ChatGPT on the code length users post in their code examples on Stack Overflow. Similar to the question length, we observe a strong and consistent positive effect for *web* tag in (a) and *python* tag in (b). In case of *java* tag in (c), the effect oscillates in the first few months until it reaches similar sizes as for the other two tags in the last month of our observation period. While larger tag groups experience larger effect sizes, for all three programming languages the example code length increases substantially in post-ChatGPT period.

answers after ChatGPT was introduced. The effect is stronger for questions and oscillates between 6% and 8% of the standard deviation of the question length, and amounts to an increase of 2% to 3% of the standard deviation of the answer length, with a slight upwards trend towards the end of the observation period, reaching almost 5% increase in the end signaling a persisting long term effect of ChatGPT on question and answer length on Stack Overflow.

Tags. In Figure 7, we show the DiD results for question length after dividing the data according to their tag groups. We observe almost identical results as before. In all tag groups, DiD coefficients are positive and significant during whole observation period, except for a few days in the beginning of December 2022 for *web* tags, as well as approximately two months period between end of December 2022 and beginning of February 2023 for *java* tag, where the confidence intervals intersect with the zero line although the estimated effect remains positive at all times. Apart from that, we observe a strong positive effect with an upwards trend for *web* and, in particular, *python* tag, and a slightly oscillating estimate in the beginning of the observation period for *java* tag, which also stabilizes at high positive values around

14

(a) Web: medium difficulty

(b) Python: medium difficulty
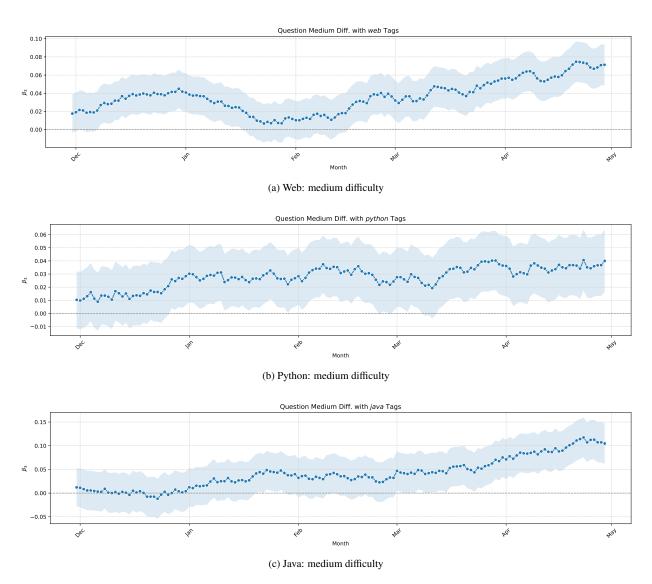

(c) Java: medium difficulty

Figure 9: **Effect of ChatGPT on the question difficulty over programming language and time.** With our DiD setup, we estimate the effect of ChatGPT on the question difficulty (here the probability of the medium question difficulty) on Stack Overflow. Similar to the question length and code length, we observe a substantial upward trend in the DiD estimate for *web* tag in (a) and *python* tag in (b) in the first two months after the introduction of ChatGPT. While the trend remain stable for *python* tag, there is a short period of time (around February 2023), in which the DiD estimate is not significant for *web* tag, before it again tilts upwards reaching almost 8% of increase in the probability of medium question difficulty at the end of the observation period. In case of *java* tag in (c), the effect oscillates around zero in the first month until it reaches sizes similar to *python* and *web* tag in May 2023. As previously, we observe that larger tag groups experience larger effect sizes.

February 2023. The effect sizes are higher for three largest tag groups than for the aggregated data and we corroborate this with the DiD results for the remaining tag groups where we observe stronger effect sizes for larger tags and smaller for tags with a smaller number of questions (due to the space restrictions we do not show these results here). In particular, the effect sizes at the end of the observation period are around 12% for *web* tag, 16% for *python* tag, and 9% for *java* tag (measured at the scale of the individual standard deviations of the question length), again signaling a strong and consistent long term effect of ChatGPT on the question length across the tags.

Code length. In Figure 8, we depict the DiD regression results for the code length of examples included in the questions that we divide according to their tags. These results almost completely mirror the results of the question length for specific tags. In particular, for *python* tag we observe a consistent positive effect with an increasing trend.

The effect size at the end of the observation period is around 21% increase in the python code examples length. For the *web* tag after a short period of approximately one month after the start of ChatGPT, the coefficient increases and remains positive and significant until the end of the observation period. The final effect sizes is around 8%. In case of the *java* examples, the coefficient oscillates in the first few months, but reaches a steady positive value from March 2023 onward. The final effect size is around 12% of the standard deviation of the *java* code examples. Similar to the question length for the individual tags, we compute DiD coefficients for all remaining tags and observe that, in general, the effect sizes are higher for larger tags (we do not show these results here due to space limitations). In summary, our results suggest a persistent and a strong positive effect of ChatGPT on the length of the code examples included in the questions on Stack Overflow.

Question and code difficulty. Finally, in Figure 9 we show the temporal evolution of the DiD estimate for the XGBoost classifier probability of the question medium difficulty for three largest tag groups. Again, we standardize the outcome variables such that the results can be interpreted as the increase in the probability of predicting medium question difficulty measured at the scale of the standard deviation of that probability. We observe, a consistent and positive trend for the DiD estimate over the whole observation period. There are slight oscillations for the *web* tag (see Figure 9a) around February 2023, and for the *java* tag in the first month of the observation period, but overall the DiD estimate is consistently significant and positive in our observation period. Again, the effect sizes is associated with the post volume, as our calculation with the smaller tags show smaller effect sizes or inconsistent and non-significant DiD coefficients for prolonged periods of time (we omit these plots from the paper due to space limitations). However, the effect sizes in May 2023 for the three largest tag groups are around 7% for *web*, 4% for *python*, and 11% for *java* tag of the standard deviation of the medium difficulty probability. This suggests a strong shift towards more difficult questions posted on Stack Overflow when compared to the pre ChatGPT period. The coefficients for the hard question difficulty are noisy and non-significant for longer periods suggesting that Stack Overflow users increasingly posted questions of the medium difficulty at the expenses of easy questions after the ChatGPT launch.

Remaining outcome variables. We note here that we also fitted DiD regressions for further outcome variables such as question views, question scores, and answer scores for the aggregated data as well as for tag groups. We do not find any consistent effects for most of these outcome variables except for a strong negative effect on question views for the aggregated data and the largest tag groups such as *web* and *python*, confirming once more results from the previous studies (Burtch et al., 2024; del Rio-Chanona et al., 2024) on the declining question volume post-ChatGPT.

Parallel trends. Following the Equation 2, we fit regression models for all combinations of the outcome variables and our data groups including the aggregated data and the data separated by the tags. The β_3 coefficient indicating the difference in trends between the control and the treated group prior to ChatGPT introduction is close to zero in all cases. Specifically, for the aggregated data the largest magnitude of β_3 is -0.0028 (p-value < 0.0001) for question views with all other estimated coefficients being closer to zero. For the three largest tag groups, we obtain -0.0033 (p-value < 0.0001) for *web* question views, -0.0036 (p-value < 0.0001) for *python* question scores, and -0.0018 (p-value $= 0.058$) for *java* question views as the largest magnitudes of β_3 coefficient, with coefficients in all other cases being even closer to zero. Hence, apart from the visual inspection of temporal plots of the quantities in question, these tests provide further evidence in favor of parallel trend assumption.

4.2. Question content drift

To gain additional insight in the nature of the change in the difficulty of the programming questions on Stack Overflow, we extend our analysis to the content of the questions. Specifically, we use BERTopic (Grootendorst, 2022) to cluster the CodeT5 embeddings that we computed earlier. To gain insight in the question content, we also extract the most representative words from each cluster after removing common stopwords, as well as highly frequent words. More specifically, starting with the data from the 22/23 period, we divide that data into pre-ChatGPT and post-ChatGPT periods. Then, from each of those periods we extract eight topics to analyze the topical shifts in questions before and after the ChatGPT launch. We repeat the topic analysis for each of our tag groups. Finally, we evaluate the quality of our topic extraction by computing the silhouette coefficient for each of the 16 topic models. We obtain the silhouette coefficients ranging from 0.299 (smallest value obtained for *C#* tag post-ChatGPT) to 0.401 (largest value obtained for *web* tag group post-ChatGPT) indicating fairly well divided topics.

In Figure 10 we show the extracted topics sorted by the number of questions that they include, together with their most representative words for *python* questions (we omit remaining topic models due to space constraints). We create the topic labels by prompting ChatGPT with the most representative words of the corresponding topic. We find that

16

(a) Python: topic analysis pre-ChatGPT

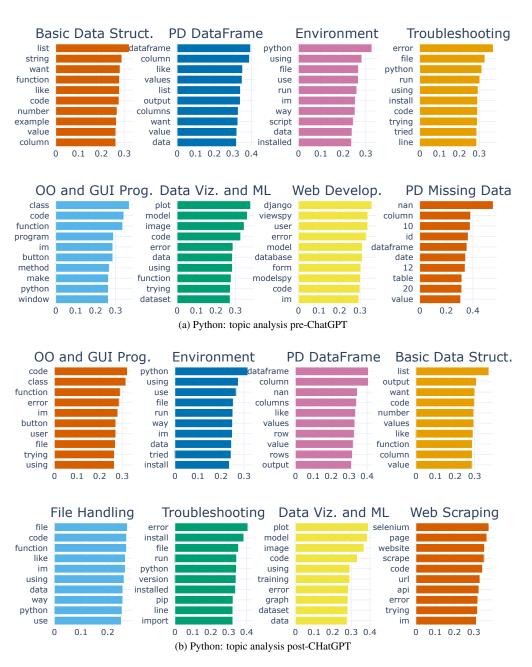(b) Python: topic analysis post-CHatGPT

Figure 10: **Content drift on Stack Overflow.** We extract eight topics from *python* questions before the ChatGPT launch and another eight topics after the launch. While the topics remain relatively stable, their relative sizes change. Before ChatGPT launch, the largest category of questions was related to *Basic Data Structures*, which are typically related to easier programming tasks. After the introduction of ChatGPT, the largest topical category is related to more difficult questions on *Object-oriented and GUI programming*.

the *python* topics are fairly stable across the periods with slight changes in the smaller topics (cf. shift from *Web development* to *Web scraping* and shift from *Pandas missing data* to *File handling* in Figure 10). However, we observe a shift in the number of questions per topic, as well as slight distributional changes regarding the most representative words. Most prominent changes are related to topics *Object-oriented and GUI programming* and *Basic Data Structures*. While *Basic Data Structures* drops from the largest topic pre-ChatGPT (17, 75% of questions) to fourth largest post-ChatGPT (12, 98% of questions), *Object-oriented and GUI programming*, climbs from the fifth largest topic pre-

17

ChatGPT (13, 95%) to the largest topic post-ChatGPT (20, 10%). In programming, basic programming concepts such as variables, loops, or basic data structures, or functions are considered a subset of more sophisticated object-oriented programming concepts (cf. ACM Computer Science Curricula (Kumar et al., 2024)). As such, object-oriented concepts are typically taught later in computer science curricula than the introductory programming concepts. We observe similar shifts towards more sophisticated topics in other tag groups, but they are typically tag specific. For example, in *java* we observe an increase in questions on the Spring Web development framework, while *C#* also experiences an increase in questions about object-oriented programming albeit a weaker one than *python*.

5. Discussion

Summary of results. With our study, we confirm previous findings of an accelerated decline of questions and answers on Stack Overflow post-ChatGPT (Burtch et al., 2024; del Rio-Chanona et al., 2024). However, we find that the length of the questions and answers, as well as, the length of code examples in questions substantially increases in both, short-term (e.g., a few weeks after the launch), as well as long-term (up to six months after the launch) periods. Moreover, we find that question difficulty after the ChatGPT introduction significantly increases and shifts from easy questions towards medium difficulty questions as estimated by our XGBoost classifier. Lastly, we observe a content drift across thematic categories towards questions related to more sophisticated concepts such as object-oriented programming.

Ask ChatGPT for simpler and the crowd for more difficult questions. We suggest that Stack Overflow community experiences an ongoing fundamental behavioral shift related to the introduction of ChatGPT. In particular, our results indicate that users, at an accelerating rate, turn to the Stack Overflow crowd in case of more difficult questions after the ChatGPT launch. On the other hand, they ask ChatGPT for answers to simpler or well-known programming questions as ChatGPT provides immediate and reasonably good answers in majority of such cases. We corroborate our large-scale quantitative results and our conclusions with the results of several recent studies related to ChatGPT, Stack Overflow, and programming tasks and questions. For example, Kabir et al. (Kabir et al., 2024) analyze ChatGPT answers to programming questions from Stack Overflow and find that more than a half of these answers contain incorrect, redundant, or irrelevant information due to its incapability to understand the larger context of the question. Often, code errors produced by ChatGPT are because of wrong logic, wrong reasoning, wrong API calls, or wrong function calls, which may indicate that ChatGPT is struggling when logical or algorithmic reasoning or coordination of external APIs or function calls is required, all defining features of more advanced programming questions. At the same time, ChatGPT makes fewer syntax errors and generally makes fewer errors when answering more popular or older Stack Overflow questions, suggesting that it handles simpler or well-established question and answers better. In another study by Widjojo and Treude (Widjojo and Treude, 2023), the authors find that LLMs can compete and outperform Stack Overflow answers on questions related to compiler errors. However, a higher quality in LLM responses to coding questions and tasks is typically related to experience in creating efficient prompts. In particular, the authors find that carefully designed prompts substantially increase the quality of LLM answers, indicating that a certain level of experience in creating prompts is needed to achieve more precise results. Along similar lines, further studies found evidence for usefulness of ChatGPT, Copilot and other general-purpose LLMs for a variety of programming and programming assistance tasks (Peng et al., 2023; Ross et al., 2023). In particular, the participants of the study by Ross et al. reported the usefulness of an LLM programming assistant for ordinary, simpler, small, and repetitive tasks, as well as for short chunks of code, little coding problems, or quick lookups (Ross et al., 2023). Similarly, Chen et al. (Chen et al., 2021), as well as Yetistiren et al. (Yetistiren et al., 2022) found that fine-tuned LLMs can produce a high quality code on easy interview problems for software developers.

ChatGPT improvements and user experience. As we only analyzed the first six months after start of the ChatGPT, shift of users to Stack Overflow for more difficult questions may be potentially due to the initial ChatGPT-related issues, and may cease with improved model versions. For example, some of the early ChatGPT problems were frequently related to deficiencies in analytic thinking and reasoning, problems later addressed by, among other approaches, chain-of-thought prompting (Wei et al., 2022). However, difficulties of LLMs in logical and algorithmic reasoning (Naveed et al., 2025; Shojaee et al., 2025; Valmeekam et al., 2022) are still present and combined with their limited domain knowledge (recently increasingly addressed by techniques such as Retrival Augmented Generation (Gao et al., 2024)), frequent hallucinations (Huang et al., 2025), or their scaling limitations (Naveed et al., 2025), still remain limiting factors in their adoption for various tasks including problem solving tasks in programming.

In addition to the limitations of the LLMs in handling more sophisticated problems, the experience in working with LLMs and the ability of users to refine prompts substantially contributes to the quality of answers (Widjojo and Treude, 2023). Combination of those two factors, i.e., intrinsic LLM limitations as well as users requiring time to gain experience, may potentially explain the accelerating trend towards Stack Overflow in case of more complex tasks (cf. Figures 7 and 8 for trend in the question length and the length of code examples, and Figure 9 for trend in question difficulty) as more and more users are able to obtain satisfactory results from ChatGPT on more sophisticated questions with time, practically setting the benchmark for Stack Overflow questions higher.

ChatGPT's disruption of Stack Overflow. The original theory of disruptive technology introduced by Christensen (Christensen, 1997) defined a disruptive technology as a technology inferior in the main attributes to the mainstream technology but with innovative features that the mainstream technology neglects. Later definitions of the disruptive technology focused more on the characteristics of the new technology and less on the niche aspects introduced by the original theory. For example, Nagy et al. (Nagy et al., 2016) define a disruptive innovation as "An innovation that changes the performance metrics, or consumer expectations, of a market by providing radically new functionality, discontinuous technical standards, or new forms of ownership." In that sense, due to LLMs' remarkable performance in various domains (Teubner et al., 2023) such as education (García-Peñalvo, 2023), medicine (Chow et al., 2023; Heng et al., 2023), research (Gao et al., 2023), or software development (Liu et al., 2023), ChatGPT and other LLMs may be already considered as a disruptive innovation.

More specifically, we now compare our study on effects and disruptive aspects of ChatGPT on help-seeking in software development with studies analyzing effects of ChatGPT on other activities in collaborative knowledge work. For example, in traditional knowledge work, studies assessing the worker productivity in consulting (Dell'Acqua et al., 2023) found differential effects of artificial intelligence tools on the consultants' productivity, suggesting that ChatGPT had a positive effect only on some of the tasks (e.g., creative product innovation), while using ChatGPT in other tasks (e.g., data and interview analysis) had deteriorating effects. Along those lines, Noy and Zhang (Noy and Zhang, 2023) found that on writing tasks ChatGPT increased the authors' productivity as well as the quality of the written texts suggesting a positive effect of ChatGPT on this particular knowledge creation task. Potentially most related to our study, a recent study (Reeves et al., 2024) analyzed page views, visitors, edits, and editors across languages to quantify the effect of ChatGPT on Wikipedia, another popular collaborative knowledge creation platform. The authors found that across all languages, all metrics increased after the introduction of ChatGPT but the increase was smaller for languages where ChatGPT is available, suggesting a diverse effect dependent on the Wikipedia language edition. Similar to those works, our study gives insight into differential effects of ChatGPT on help-seeking platforms for software developers—while previous studies of ChatGPT and Stack Overflow found decaying number of questions and answers as the consequence of ChatGPT (Burtch et al., 2024; del Rio-Chanona et al., 2024), we find that the questions, answers, and code examples are longer. Additionally, the question difficulty increases and the question content shifts towards more advanced topics after the ChatGPT start.

As our results for the first six months of ChatGPT suggest, we expect the observed disruption of Stack Overflow due to ChatGPT to continue in future. In particular, we argue that (i) as LLMs mature, (ii) as they are further fine-tuned and developed, (iii) as new achievements in prompt engineering are achieved, and (iv) as users gain more understanding and experience in interacting and working with LLMs, we may experience sustained and ongoing shifts in the user behavior with ChatGPT and on question answering platforms. Specifically, we expect users to rely on LLMs for, not only, a wide range of well-established, simpler problems, but also for an ever larger number of more and more difficult questions. This development will set the bar for Stack Overflow questions higher—as more difficult questions are answered by LLMs in a satisfactory manner, the users will continue to turn towards online help-seeking communities for even harder, more complex, more recent, or logically or algorithmically more challenging questions. Hence, we conclude that while ChatGPT disrupted Stack Overflow platform leading to less questions in total, it also started a transformation of the platform, in particular, a transformation of the content posted on Stack Overflow.

Limitations. Our work is not without limitations. As in all causal inference, we cannot rule out that unmeasured confounders still affect our results. For example, other external events such as new versions of the programming languages, new releases of the standard libraries, or introduction of new programming frameworks may also lead to longer and more difficult questions on Stack Overflow. Moreover, during the first six months after ChatGPT introduction, several versions of the ChatGPT model have been released, potentially distorting our results and conclusions.

While we can not account for all of such confounders, we tried as good as possible to obtain statistically valid and robust results with our DiD regression setup. For example, by estimating the trends in the ChatGPT's effect on

Stack Overflow with the "secret weapon", we account for any after period sensitivity (e.g., in the exact starting date of ChatGPT) of the effect. On the other hand, we do not take the same approach in handling potential sensitivities in the before period as we always work with the complete six months data prior to ChatGPT. With this design decision we aim to (i) reduce uncertainty in the estimates by analyzing more data, and to (ii) account for any long-term trends on Stack Overflow by finding evidence for the parallel trend assumption between the treatment and the control group.

Our control group is another limitation of our work as this is only a quasi-control group not obtained via an experimental design. With our control group we make an implicit assumption that there are no significant distributional drifts in the users, their behavior, and the content they post (apart from the drift caused by the ChatGPT intervention), i.e., that the control and treatment groups are fairly comparable to each other in the majority of their defining features. We believe that by taking one year of data for both groups, we account for all seasonal aspects, as well as that sudden changes in behavior are averaged out over such prolonged periods of time. However, as is common in causal inference, we can not guarantee the absence of systemic behavioral changes in the data. Potentially, such changes could be eliminated by combining our DiD regression setup with matching on the user, topic, or content level. We see this as an interesting avenue for future research in this area.

6. Conclusions

Summary & implications. With our large-scale analysis of two years of Stack Overflow data we found that ChatGPT sustainably changed content creation on Stack Overflow. In particular, users tend to write longer questions with longer code examples and questions of an increased difficulty. As ChatGPT is able to provide quick and satisfactory results to simpler, shorter questions, these new circumstances may be also a manifestation of a deeper user behavioral change— modern artificial intelligence tools are relieving the user work burden, in particular burden of repetitive or elementary work activities in software development. While these new conditions result in higher efficiency, more resources, and more time for work on more sophisticated problems, they also raise the question of the future of online collaborative platforms such as Stack Overflow. With ChatGPT and similar tools around, will such programming help-seeking platforms still be necessary in the future? Our results suggest that at the moment they still are, and that they will remain necessary also in the future. However, these platforms will continue to profoundly change as they transition towards new topics, new and more sophisticated content, or new ways of help-seeking in software development.

Future work. The analysis of such transformative future changes represents an interesting direction for future work. For example, a deeper analysis of the Stack Overflow content can reveal more specific drifts in the question topics including emergence of the new topics such as topics related to the use of LLMs themselves. Also, the analysis of user activities, their searching or tagging behavior may provide insights into the specific cases in which users turn to Stack Overflow instead of ChatGPT and may help improve information organization and retrieval functionality on the platform. Finally, analyzing user social interactions, their voting or commenting behavior may provide further actionable information for the operators of collaborative knowledge creation platforms on how to adjust or add new features to better reflect new actuality induced by modern artificial intelligence tools.

References

Abdalkareem, R., Shihab, E., Rilling, J., 2017. What do developers use the crowd for? a study using stack overflow. IEEE Software 34, 53–60. doi:10.1109/MS.2017.31.

Allamanis, M., Sutton, C., 2013. Why, when, and what: Analyzing stack overflow questions by topic, type, and code, in: 2013 10th Working Conference on Mining Software Repositories (MSR), pp. 53–56. doi:10.1109/MSR.2013.6624004.

Asaduzzaman, M., Mashiyat, A.S., Roy, C.K., Schneider, K.A., 2013. Answering questions about unanswered questions of stack overflow, in: 2013 10th Working Conference on Mining Software Repositories (MSR), pp. 97–100. doi:10.1109/MSR.2013.6624015.

Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q., Sutton, C., 2021. Program synthesis with large language models. arXiv:2108.07732.

Blyth, C.R., 1972. On simpson's paradox and the sure-thing principle. Journal of the American Statistical Association 67, 364–366. doi:10.1080/01621459.1972.10482387.

Burtch, G., Lee, D., Chen, Z., 2024. The consequences of generative ai for online knowledge communities. Scientific Reports 14, 10413. doi:10.1038/s41598-024-61221-0.

Calefato, F., Lanubile, F., Novielli, N., 2018. How to ask for technical help? evidence-based guidelines for writing questions on stack overflow. Information and Software Technology 94, 186–207. doi:10.1016/j.infsof.2017.10.009.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P.S., Yang, Q., Xie, X., 2024. A survey on evaluation of large language models. ACM Trans. Intell. Syst. Technol. 15. doi:10.1145/3641289.

Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H.P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F.P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W.H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A.N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., Zaremba, W., 2021. Evaluating large language models trained on code. arXiv:2107.03374.

Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM. p. 785–794. doi:10.1145/2939672.2939785.

Chow, J.C., Sanders, L., Li, K., 2023. Impact of chatgpt on medical chatbots as a disruptive technology. Frontiers in Artificial Intelligence 6, 1166014. doi:10.3389/frai.2023.1166014.

Christensen, C., Raynor, M., 2003. The innovator's solution: Creating and sustaining successful growth. Harvard Business School Press.

Christensen, C.M., 1997. The innovator's dilemma: when new technologies cause great firms to fail. Harvard Business School Press.

Delile, Z., Radel, S., Godinez, J., Engstrom, G., Brucker, T., Young, K., Ghanavati, S., 2023. Evaluating privacy questions from stack overflow: Can chatgpt compete?, in: 2023 IEEE 31st International Requirements Engineering Conference Workshops (REW), IEEE. pp. 239–244. doi:10.1109/REW57809.2023.00048.

Dell'Acqua, F., McFowland III, E., Mollick, E.R., Lifshitz-Assaf, H., Kellogg, K., Rajendran, S., Krayer, L., Candelon, F., Lakhani, K.R., 2023. Navigating the jagged technological frontier: Field experimental evidence of the effects of ai on knowledge worker productivity and quality. Harvard Business School Technology & Operations Mgt. Unit Working Paper .

Fischer, F., Böttinger, K., Xiao, H., Stransky, C., Acar, Y., Backes, M., Fahl, S., 2017. Stack overflow considered harmful? the impact of copy&paste on android application security, in: 2017 IEEE Symposium on Security and Privacy (SP), IEEE. pp. 121–136. doi:10.1109/SP.2017.31.

Gao, C.A., Howard, F.M., Markov, N.S., Dyer, E.C., Ramesh, S., Luo, Y., Pearson, A.T., 2023. Comparing scientific abstracts generated by chatgpt to real abstracts with detectors and blinded human reviewers. NPJ digital medicine 6, 75. doi:10.1038/s41746-023-00819-6.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., Wang, H., 2024. Retrieval-augmented generation for large language models: A survey. arXiv:2312.10997.

García-Peñalvo, F.J., 2023. The perception of artificial intelligence in educational contexts after the launch of chatgpt: Disruption or panic? Education in the Knowledge Society 24, e31279. doi:10.14201/eks.31279.

Gelman, A., and, Z.H., 2008. Estimating incumbency advantage and its variation, as an example of a before–after study. Journal of the American Statistical Association 103, 437–446. doi:10.1198/016214507000000626.

Grootendorst, M., 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. arXiv:2203.05794.

Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., Yue, J., Wu, Y., 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. arXiv:2301.07597.

Hang, C.C., Garnsey, E., Ruan, Y., 2015. Opportunities for disruption. Technovation 39-40, 83 – 93. doi:10.1016/j.technovation.2014.11.005.

Haque, M.U., Dharmadasa, I., Sworna, Z.T., Rajapakse, R.N., Ahmad, H., 2022. I think this is the most disruptive technology: Exploring sentiments of chatgpt early adopters using twitter data. arXiv:2212.05856.

Heng, J.J.Y., Teo, D.B., Tan, L.F., 2023. The impact of chat generative pre-trained transformer (chatgpt) on medical education. Postgraduate Medical Journal 99, 1125–1127. doi:10.1093/postmj/qgad058.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., Liu, T., 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. ACM Trans. Inf. Syst. 43. doi:10.1145/3703155.

Husain, H., Wu, H.H., Gazit, T., Allamanis, M., Brockschmidt, M., 2020. Codesearchnet challenge: Evaluating the state of semantic code search. arXiv:1909.09436.

Iyer, S., Konstas, I., Cheung, A., Zettlemoyer, L., 2016. Summarizing source code using a neural attention model, in: Erk, K., Smith, N.A. (Eds.), Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany. pp. 2073–2083. doi:10.18653/v1/P16-1195.

Jesse, K., Ahmed, T., Devanbu, P.T., Morgan, E., 2023. Large language models and simple, stupid bugs, in: 2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR), IEEE. pp. 563–575. doi:10.1109/MSR59073.2023.00082.

Jimenez, C.E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., Narasimhan, K., 2024. Swe-bench: Can language models resolve real-world github issues? arXiv:2310.06770.

Kabir, S., Udo-Imeh, D.N., Kou, B., Zhang, T., 2024. Is stack overflow obsolete? an empirical study of the characteristics of chatgpt answers to stack overflow questions, in: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, ACM. pp. 1–17. doi:10.1145/3613904.3642596.

Kou, B., Chen, M., Zhang, T., 2023. Automated summarization of stack overflow posts, in: 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), pp. 1853–1865. doi:10.1109/ICSE48619.2023.00158.

Kumar, A.N., Raj, R.K., Aly, S.G., Anderson, M.D., Becker, B.A., Blumenthal, R.L., Eaton, E., Epstein, S.L., Goldweber, M., Jalote, P., Lea, D., Oudshoorn, M., Pias, M., Reiser, S., Servin, C., Simha, R., Winters, T., Xiang, Q., 2024. Computer Science Curricula 2023. ACM. doi:10.1145/3664191.

Liu, J., Xia, C.S., Wang, Y., ZHANG, L., 2023. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation, in: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc.. pp. 21558–21572. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/43e9d647ccd3e4b7b5baab53f0368686-Paper-Conference.pdf.

Luo, Z., Xu, C., Zhao, P., Sun, Q., Geng, X., Hu, W., Tao, C., Ma, J., Lin, Q., Jiang, D., 2023. Wizardcoder: Empowering code large language models with evol-instruct. arXiv:2306.08568.

Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G., Hartmann, B., 2011. Design lessons from the fastest q&a site in the west, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM. p. 2857–2866. doi:10.1145/1978942.1979366.

Markides, C., 2006. Disruptive innovation: In need of better theory. Journal of Product Innovation Management 23, 19 – 25. doi:10.1111/j.1540-5885.2005.00177.x. cited by: 842.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A., 2021. A survey on bias and fairness in machine learning. ACM Comput. Surv. 54. doi:10.1145/3457607.

Moradi Dakhel, A., Majdinasab, V., Nikanjam, A., Khomh, F., Desmarais, M.C., Jiang, Z.M.J., 2023. Github copilot ai pair programmer: Asset or liability? Journal of Systems and Software 203, 111734. doi:https://doi.org/10.1016/j.jss.2023.111734.

Nagy, D., Schuessler, J., Dubinsky, A., 2016. Defining and identifying disruptive innovations. Industrial Marketing Management 57, 119–126. doi:https://doi.org/10.1016/j.indmarman.2015.11.017.

Naveed, H., Khan, A.U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., Mian, A., 2025. A comprehensive overview of large language models. ACM Trans. Intell. Syst. Technol. 16. doi:10.1145/3744746.

Noy, S., Zhang, W., 2023. Experimental evidence on the productivity effects of generative artificial intelligence. Science 381, 187–192. doi:10.1126/science.adh2586.

Peng, S., Kalliamvakou, E., Cihon, P., Demirer, M., 2023. The impact of ai on developer productivity: Evidence from github copilot. arXiv:2302.06590.

Perry, N., Srivastava, M., Kumar, D., Boneh, D., 2023. Do users write more insecure code with ai assistants?, in: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, ACM. p. 2785–2799. doi:10.1145/3576915.3623157.

Rahman, M.M., Barson, J., Paul, S., Kayani, J., Lois, F.A., Quezada, S.F., Parnin, C., Stolee, K.T., Ray, B., 2018. Evaluating how developers use general-purpose web-search for code retrieval, in: Proceedings of the 15th International Conference on Mining Software Repositories, ACM. p. 465–475. doi:10.1145/3196398.3196425.

Reeves, N., Yin, W., Simperl, E., 2024. Exploring the impact of chatgpt on wikipedia engagement. arXiv:2405.10205.

del Rio-Chanona, R.M., Laurentsyeva, N., Wachs, J., 2024. Large language models reduce public knowledge sharing on online q&a platforms. PNAS Nexus 3, pgae400. doi:10.1093/pnasnexus/pgae400.

Ross, S.I., Martinez, F., Houde, S., Muller, M., Weisz, J.D., 2023. The programmer's assistant: Conversational interaction with a large language model for software development, in: Proceedings of the 28th International Conference on Intelligent User Interfaces, ACM. p. 491–514. doi:10.1145/3581641.3584037.

Sandoval, G., Pearce, H., Nys, T., Karri, R., Garg, S., Dolan-Gavitt, B., 2023. Lost at c: A user study on the security implications of large language model code assistants, in: 32nd USENIX Security Symposium (USENIX Security 23), USENIX Association, Anaheim, CA. pp. 2205–2222. URL: https://www.usenix.org/conference/usenixsecurity23/presentation/sandoval.

Santos, T., Burghardt, K., Lerman, K., Helic, D., 2020. Can badges foster a more welcoming culture on q&a boards? Proceedings of the International AAAI Conference on Web and Social Media 14, 969–973. doi:10.1609/icwsm.v14i1.7368.

Santos, T., Walk, S., Kern, R., Strohmaier, M., Helic, D., 2019a. Activity archetypes in question-and-answer (q8a) websites—a study of 50 stack exchange instances. Trans. Soc. Comput. 2. doi:10.1145/3301612.

Santos, T., Walk, S., Kern, R., Strohmaier, M., Helic, D., 2019b. Self- and cross-excitation in stack exchange question & answer communities, in: The World Wide Web Conference, ACM. p. 1634–1645. doi:10.1145/3308558.3313440.

Shen, X., Chen, Z., Backes, M., Zhang, Y., 2023. In chatgpt we trust? measuring and characterizing the reliability of chatgpt. arXiv:2304.08979.

Shojaee, P., Mirzadeh, I., Alizadeh, K., Horton, M., Bengio, S., Farajtabar, M., 2025. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. arXiv:2506.06941.

Si, S., Chen, H., 2020. A literature review of disruptive innovation: What it is, how it works and where it goes. Journal of Engineering and Technology Management 56, 101568. doi:https://doi.org/10.1016/j.jengtecman.2020.101568.

Siddiq, M.L., Majumder, S.H., Mim, M.R., Jajodia, S., Santos, J.C.S., 2022. An empirical study of code smells in transformer-based code generation techniques, in: 2022 IEEE 22nd International Working Conference on Source Code Analysis and Manipulation (SCAM), IEEE. pp. 71–82. doi:10.1109/SCAM55253.2022.00014.

de Souza, L.B.L., Campos, E.C., Maia, M.d.A., 2014. Ranking crowd knowledge to assist software development, in: Proceedings of the 22nd International Conference on Program Comprehension, ACM. p. 72–82. doi:10.1145/2597008.2597146.

Storey, M.A., Treude, C., van Deursen, A., Cheng, L.T., 2010. The impact of social media on software engineering practices and tools, in: Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research, ACM. p. 359–364. doi:10.1145/1882362.1882435.

Teubner, T., Flath, C.M., Weinhardt, C., Van Der Aalst, W., Hinz, O., 2023. Welcome to the era of chatgpt et al. the prospects of large language models. Business & Information Systems Engineering 65, 95–101. doi:10.1007/s12599-023-00795-x.

Treude, C., Barzilay, O., Storey, M.A., 2011. How do programmers ask and answer questions on the web? (nier track), in: Proceedings of the 33rd International Conference on Software Engineering, ACM. p. 804–807. doi:10.1145/1985793.1985907.

Vaithilingam, P., Zhang, T., Glassman, E.L., 2022. Expectation vs experience: Evaluating the usability of code generation tools powered by large language models, in: Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems, ACM. pp. 1–7. doi:10.1145/3491101.3519665.

Valmeekam, K., Olmo, A., Sreedharan, S., Kambhampati, S., 2022. Large language models still can't plan (a benchmark for llms on planning and reasoning about change), in: NeurIPS 2022 Foundation Models for Decision Making Workshop. URL: https://openreview.net/forum?id=wUU-7XTL5XO.

Varghese, J., Chapiro, J., 2024. Chatgpt: The transformative influence of generative ai on science and healthcare. Journal of Hepatology 80, 977–980. doi:https://doi.org/10.1016/j.jhep.2023.07.028.

Vasilescu, B., Filkov, V., Serebrenik, A., 2013. Stackoverflow and github: Associations between software development and crowdsourced knowledge, in: 2013 International Conference on Social Computing, pp. 188–195. doi:10.1109/SocialCom.2013.35.

Wang, Y., Wang, W., Joty, S., Hoi, S.C., 2021. CodeT5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation, in: Moens, M.F., Huang, X., Specia, L., Yih, S.W.t. (Eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic. pp. 8696–8708. doi:10.18653/v1/2021.emnlp-main.685.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., ichter, b., Xia, F., Chi, E., Le, Q.V., Zhou, D., 2022. Chain-of-thought prompting elicits reasoning in large language models, in: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc.. pp. 24824–24837. URL: `https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf`.

Widjojo, P., Treude, C., 2023. Addressing compiler errors: Stack overflow or large language models? `arXiv:2307.10793`.

Xu, F.F., Alon, U., Neubig, G., Hellendoorn, V.J., 2022. A systematic evaluation of large language models of code, in: Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming, ACM. p. 1–10. doi:`10.1145/3520312.3534862`.

Yang, D., Hussain, A., Lopes, C.V., 2016. From query to usable code: an analysis of stack overflow code snippets, in: Proceedings of the 13th International Conference on Mining Software Repositories, ACM. p. 391–402. doi:`10.1145/2901739.2901767`.

Ye, J., Chen, X., Xu, N., Zu, C., Shao, Z., Liu, S., Cui, Y., Zhou, Z., Gong, C., Shen, Y., Zhou, J., Chen, S., Gui, T., Zhang, Q., Huang, X., 2023. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. `arXiv:2303.10420`.

Yetistiren, B., Ozsoy, I., Tuzun, E., 2022. Assessing the quality of github copilot's code generation, in: Proceedings of the 18th International Conference on Predictive Models and Data Analytics in Software Engineering, ACM. p. 62–71. doi:`10.1145/3558489.3559072`.

Yin, P., Deng, B., Chen, E., Vasilescu, B., Neubig, G., 2018. Learning to mine aligned code and natural language pairs from stack overflow, in: Proceedings of the 15th International Conference on Mining Software Repositories, ACM. p. 476–486. doi:`10.1145/3196398.3196408`.

Zhong, L., Wang, Z., 2024. Can llm replace stack overflow? a study on robustness and reliability of large language model code generation. Proceedings of the AAAI Conference on Artificial Intelligence 38, 21841–21849. doi:`10.1609/aaai.v38i19.30185`.