Voices from the Frontier: A Comprehensive Analysis of the OpenAI Developer Forum

Xinyi Hou*, Yanjie Zhao*, and Haoyu Wang†
Huazhong University of Science and Technology, Wuhan, China
xinyihou@hust.edu.cn, yanjie_zhao@hust.edu.cn, haoyuwang@hust.edu.cn

***Abstract*—OpenAI's advanced large language models (LLMs) have revolutionized natural language processing and enabled developers to create innovative applications. As adoption grows, understanding the experiences and challenges of developers working with these technologies is crucial. This paper presents a comprehensive analysis of the OpenAI Developer Forum, focusing on (1) popularity trends and user engagement patterns, and (2) a taxonomy of challenges and concerns faced by developers. We first employ a quantitative analysis of the metadata from 29,576 forum topics, investigating temporal trends in topic creation, the popularity of topics across different categories, and user contributions at various trust levels. We then qualitatively analyze content from 9,301 recently active topics on developer concerns. From a sample of 886 topics, we construct a taxonomy of concerns in the OpenAI Developer Forum. Our findings uncover critical concerns raised by developers in creating AI-powered applications and offer targeted recommendations to address them. This work not only advances AI-assisted software engineering but also empowers developer communities to shape the responsible evolution and integration of AI technology in society.**

I. Introduction

OpenAI [101], a leading AI research organization, has made significant contributions to the field of natural language processing with the development of large language models (LLMs) like GPT-4 [90]. These models have demonstrated remarkable capabilities in language understanding, generation, and task-solving. The release of ChatGPT [98], a conversational AI system, has further sparked interest and engagement from the developer community. To support this growing interest, OpenAI provides a rich suite of tools and resources to streamline AI integration [103], including the `Assistant API` [97], `.NET SDK` [100], `Vector databases` [104], etc. Additionally, the GPT Store [99] allows developers to create and share custom GPTs, fostering a vibrant ecosystem of innovation. As developers increasingly leverage OpenAI's models to build applications across various domains, they encounter a range of challenges and concerns specific to working with these AI systems. These challenges **span technical issues related to model integration, prompt engineering, and output quality, as well as broader ethical considerations about bias, fairness, and responsible deployment**. Understanding and addressing these challenges is crucial for the software engineering (SE) community to ensure the development of robust, reliable AI-powered applications.

The OpenAI Developer Forum [102] serves as a primary platform for developers to discuss their experiences, seek guidance, and share best practices related to working with OpenAI's technologies. Figure 1 illustrates the trend in topic counts on the forum over time, with notable spikes coinciding with key events such as the launch of ChatGPT, GPT-4, and GPT Store. This trend suggests that the forum acts as a barometer for developer interest and engagement with OpenAI's technologies, making it a valuable resource for understanding the real-world challenges developers face and identifying areas where additional research, tools, and support are needed. In this paper, we are the first to present a comprehensive analysis of the OpenAI Developer Forum, focusing on two main research questions (RQs):

RQ1: Popularity Trends. RQ1 aims to conduct a comprehensive analysis of the OpenAI Developer Forum's overall activity. We seek to provide a foundational understanding of the forum's dynamics, highlighting popular topics and user engagement trends, which is essential for identifying key areas of interest within the developer community.

RQ2: Taxonomy of Concerns. RQ2 focuses on classifying and analyzing the specific concerns raised by developers in the OpenAI Developer Forum. Identifying these concerns will help in understanding the common obstacles and issues developers encounter, offering valuable insights that can improve best practices and support the SE community.

To address RQ1, we quantitatively analyze forum metadata, examining topic creation times, category distribution, post scores, and active user numbers, providing insights into popularity trends and user engagement. For RQ2, we perform a qualitative analysis of topics in the forum to develop a taxonomy that categorizes the practical challenges and broader concerns of developers. Our contributions include:

1) We collected 29,576 topics from the OpenAI Developer Forum and provided an in-depth analysis of the forum's popularity trends, highlighting key areas of interest and user engagement patterns.
2) We filtered 9,301 recently active topics related to developers' challenges and concerns on the OpenAI Developer Forum. We then constructed a comprehensive taxonomy of these discussions based on a sample of 886 topics.
3) We are the first to conduct a large-scale OpenAI Developer Forum analysis, providing insights that guide future research, tool development, and best practices in AI-assisted software engineering.

*Xinyi Hou and Yanjie Zhao contributed equally to this work.
†Haoyu Wang is the corresponding author (haoyuwang@hust.edu.cn).

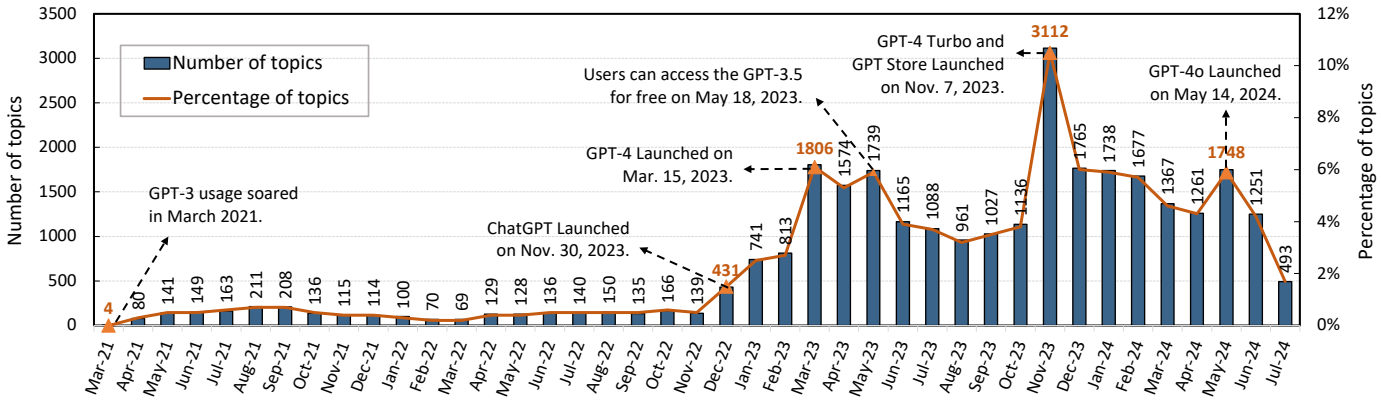


Fig. 1: Trend in topic counts on OpenAI Developer Forum over time (Up to July 16, 2024).

TABLE I: Topic categories on OpenAI Developer Forum.

Topic Category	Description	Subcategory
Announcements	Official updates related to OpenAI, the API, ChatGPT, and more.	/
API	Questions, feedback, and best practices around building with OpenAI's API.	Bugs, Feedback, Deprecations
ChatGPT	Questions or discussions about ChatGPT.	Bugs, Feature requests, Use cases, Support
Community	A place to connect with the OpenAI Developer community.	/
Forum feedback	Feedback on how to make this developer forum better for users.	/
GPT builders	Create tailored versions of ChatGPT for specific tasks and share them.	Plugins & Actions builders, Plugin store
Documentation	Share feedback on documentation and tutorials about OpenAI.	/
Prompting	Learn more about prompting by sharing best practices and more.	/

II. Background and Related Work

A. OpenAI Developer Forum

The OpenAI Developer Forum [102] is a pivotal platform where LLM developers, researchers, and enthusiasts converge to discuss, troubleshoot, and share insights on various aspects of LLM technology. As one of the most active online communities dedicated to LLMs, the forum serves as a microcosm of the broader trends, challenges, and concerns faced in the field of LLM for SE [96]. The forum is divided into main categories, as shown in Table I. These include: `Announcements` for updates from OpenAI; `API` for discussing bugs, feedback, and deprecations of OpenAI's APIs; `ChatGPT` for support and feature requests of ChatGPT; `Community` for general interactions; `Forum feedback` for platform improvement suggestions; `GPT builders` for topics on the GPT Store and GPT development; `Documentation` for discussions on OpenAI's resources; and `Prompting` for optimizing AI interactions. The OpenAI Developer Forum features a diverse community, from novices to leading researchers. This diversity promotes knowledge exchange, allowing beginners to learn from experts and experts to stay updated on practical challenges. The forum's collaborative nature encourages knowledge sharing, creating a dynamic knowledge base. For researchers, **it offers insights into real-world LLM usage, helping identify common issues and technology gaps**. This grounds theoretical research in practical needs, aligning academic progress with developer community requirements.

B. Related Work on Forum Analysis

Online forums in the SE community, like Stack Overflow, contain a wealth of information about problems encountered during software development, solutions adopted, developer opinions, and experiences. Numerous studies have analyzed online forum data in SE communities to understand developers' challenges. Rosen et al. [105] investigated mobile developers' questions on Stack Overflow, providing insights for research and tool improvements. Abdellatif et al. [89] studied chatbot development challenges through Stack Overflow posts, while Wan et al. [108] examined blockchain platform discussions across Stack Exchange communities. Zhang et al. [110] and Han et al. [95] investigated challenges in deep learning applications and frameworks. Venkatesh et al. [107] analyzed client developers' concerns when using web APIs, and Ahmed et al. [91] explored concurrency-related questions on Stack Overflow.

These studies guide research directions, tool development, and best practices in various SE domains. Unfortunately, **no research has yet analyzed the OpenAI Developer Forum**. This gap motivates our study, which examines the experiences and challenges of developers working with cutting-edge AI technologies. Our work provides insights to help SE researchers identify key areas needing further investigation and support for effective AI adoption in software development.

III. Methodology

This section outlines the systematic methodology used to analyze the OpenAI Developer Forum, which consists of five primary steps, as illustrated in Figure 2.

A. Crawl Forum Topic Lists

To initiate our study, we developed a custom web crawler to retrieve comprehensive topic lists from the OpenAI Developer

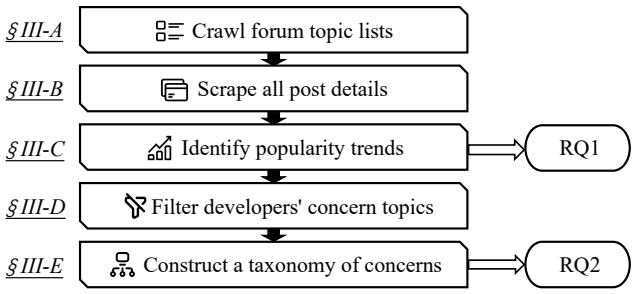


Fig. 2: Overview of methodology.

Forum. All topics must be assigned to a specific category upon creation, allowing us to systematically traverse the 17 categories listed in Table I (eight main categories and nine subcategories) to compile the complete topic list of the OpenAI Developer Forum. We identified the backend API endpoints for each category and paginated through them to collect basic information about each topic, including 33 attributes such as *topic_id*, *title*, *topic_slug*, *posts_count*, *created_at*, and *tags*. To verify the completeness of our retrieved topic list, we also crawled all topics from the homepage under the "Latest", "Top", and "Hot" sections, which are not restricted by topic categories. The final topic count obtained from these four different crawling methods was consistent, confirming the comprehensiveness of our topic list. Through this systematic collection process, we retrieved a total of 29,576 topics by July 16, 2024, which not only facilitated the subsequent scraping of posts but also provided data to support the analysis of popularity trends and developer concerns.

B. Scrape All Post Details

Based on the retrieved *topic_id*, *topic_slug*, and *posts_count*, we constructed precise URLs for all topic pages. We then used automated scraping tools to extract complete post information for each topic, including 52 attributes such as *post_id*, *username*, *cooked* (the post's actual content), *post_type*, *incoming_link_count*, *readers_count*, *trust_level*, and *score*.

C. Identify Popularity Trends

To identify trends in topic popularity and user engagement, we conducted a time series analysis to investigate the growth and decline of interest in specific topics over time. Additionally, we employed statistical methods to examine the relationships between various factors (e.g., topic category, author reputation, and post sentiment) and the level of engagement received by individual posts. These analyses provided valuable insights into the dynamics of popularity within the OpenAI Developer Forum.

D. Filter Developers' Concern Topics

To understand the current concerns of developers, we filtered the collected 29,576 topics through a four-step process. First, to ensure the topicality of the discussions, we retained 12,440 topics that were last updated in 2024 based on the *last_posted_at* attribute. Second, to ensure these topics are currently accessible to the public, we filtered down to 10,556 topics using the *closed* attribute. Given that the OpenAI Developer Forum is an active community with developers and OpenAI staff actively participating, we then focused on 9,417 topics that had not yet received an accepted answer, as indicated by the *has_accepted_answer* attribute. As shown in Table I, the forum comprises various categories, but `Announcements`, `Community`, `Forum feedback`, and `Documentation` mainly address product launches and community building, not specific developer concerns. Therefore, we focused our analysis on the remaining four categories: `API`, `ChatGPT`, `GPT builders`, and `Prompting`. After this final filtering step, we were left with 9,301 topics, distributed as follows: 5,865 topics in `API`, 1,724 topics in `ChatGPT`, 1,113 topics in `GPT builders`, and 599 topics in `Prompting`.

E. Construct a Taxonomy of Concerns

We constructed a taxonomy of concerns based on the OpenAI Developer Forum's official categories: `API/ChatGPT`, `GPT builders`, and `Prompting`. Given the similarity in topics between `API` and `ChatGPT` in the forum and the frequent comparisons made by developers between API and ChatGPT, we combined these two categories for analysis. According to Zhang et al. [110], to ensure a 95% confidence level and a 5% confidence interval, we sampled a total of 886 topics from the three categories (`API/ChatGPT` with 366 topics, `GPT Builders` with 286 topics, and `Prompting` with 234 topics) for constructing the taxonomy. The taxonomy construction process is described below.

Manual preliminary construction. Utilizing an open coding procedure [106], we inductively developed categories and subcategories for the taxonomy by analyzing topics from the OpenAI Developer Forum. Two researchers (referred to as inspectors), who both possess extensive experience in LLM development and API usage, collaborated on the preliminary construction. We randomly sampled 30% [93] of the 886 forum topics for this initial phase.

The inspectors thoroughly reviewed all sampled topics, taking into account the title, tags, body, replies, and any URLs referenced by forum participants. Topics unrelated to developer concerns, such as those promoting GPTs, were not retained. For the rest of the topics, the inspectors assigned brief phrases as initial codes to represent the underlying concerns related to API/ChatGPT usage, GPT builders, and prompting. The inspectors then grouped similar codes into categories, forming a hierarchical taxonomy that addresses concerns specific to OpenAI's products and services. This grouping process was iterative, with inspectors continuously refining the taxonomy by moving between categories and forum topics. Topics related to multiple concerns were assigned to all relevant categories. All disagreements were resolved through discussion with an experienced arbitrator knowledgeable of OpenAI's ecosystem.

Reliability analysis and extended construction. Based on the coding schema from the preliminary construction, the two inspectors independently coded the remaining 70% [93] of forum topics for reliability analysis. Each topic was assigned to the identified leaf categories in the taxonomy or discarded for being unrelated to developer concerns. Topics that could not be classified within the existing taxonomy were placed into a newly created category called `pending`. The independent labeling process yielded an inter-rater agreement of 0.835, as measured by Cohen's Kappa (κ). This high level of agreement

signifies almost perfect concordance, highlighting the reliability of our coding schema. Coding conflicts were discussed and resolved with the arbitrator. For `pending` topics, the arbitrator assisted in identifying the underlying concerns and deciding if new categories were required. Finally, six additional leaf categories were created, and all topics previously classified as `pending` were assigned to the taxonomy. The entire manual construction process took roughly 200 person-hours.

In summary, among the 886 sampled topics, 169 were unrelated to developer concerns, and 717 topics were covered in the final taxonomy. The resulting taxonomy consists of three root categories (i.e., `API/ChatGPT`, `GPT Builders`, and `Prompting`), 13 inner categories, and 50 leaf categories, as shown in Figure 4.

IV. RQ1: Popularity Trends

This section explores the popularity trends within the OpenAI Developer Forum. By analyzing various aspects of forum activity, we aim to uncover patterns in user engagement and topic discussions. We specifically examine the temporal trends in topic creation (§ IV-A), the popularity of topics across different categories (§ IV-B), and the contributions of users at various trust levels (§ IV-C).

A. Topic Trends Over Time

The OpenAI Developer Forum has become a hub for developers to discuss cutting-edge advancements in AI and share their insights on the latest developments from OpenAI. As seen in Figure 1, the forum's popularity and user engagement have grown significantly over time, **with spikes in activity often coinciding with major announcements and releases from OpenAI**. Key events such as the launch of GPT-4 in March 2023, and the introduction of the GPT Store in November 2023 have sparked widespread discussions among developers on the forum. These spikes in activity demonstrate the keen interest and excitement within the developer community regarding OpenAI's groundbreaking advancements in language models and AI technologies. The OpenAI Developer Forum has clearly become a focal point for developers who are passionate about staying up-to-date with the latest developments in AI. The concerns, ideas, and insights shared by this vibrant community of developers can provide valuable inspiration and guidance for the broader SE community.

B. Topic Popularity by Category

The OpenAI Developer Forum organizes topics into eight main categories, as shown in Table I. Among these categories, `API`, `ChatGPT`, and `GPT builders` contain several subcategories. Table II presents the distribution of topic counts across these categories. The `Announcements` category has the fewest number of topics, primarily because only official announcements can be posted in this category. The `API` category far surpasses other categories in terms of both topic and post counts, indicating that **discussions related to APIs generate the highest level of interest and engagement on the OpenAI Developer Forum.**

Table II includes the Max Score and Avg. Score for each category. Forum posts are assigned scores, likely reflecting their popularity or relevance. These scores influence topic rankings in the Hot and Top channels. We hypothesize that scores are based on various factors like the number of reads, replies, quotes, incoming links, positive feedback, timestamps, and the user's trust level or role. Notably, the highest Max Score is in the `Community` category for the topic "ChatPDF.com - Chat with any PDF using the new ChatGPT API", indicating strong community interest in practical applications of OpenAI's technologies. The `Announcements` category has the highest Avg. Score, likely due to the important of official updates and the extensive discussions they generate.

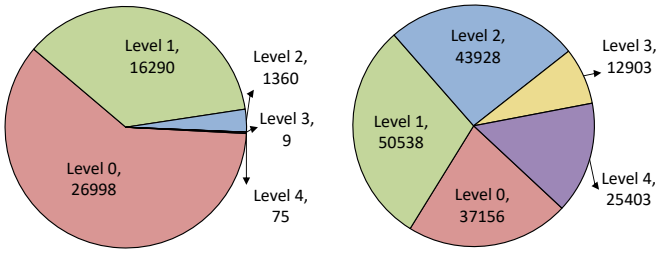
TABLE II: Topic categories and their statistics.

Category	# Topics	# Posts	Max Score	Avg. Score
Announcements	35	1,077	289846.40	5068.01
API	15,221	83,280	871119.40	890.53
ChatGPT	5,433	30,195	1406922.80	922.79
Community	3,499	23,361	4486547.40	987.42
Forum feedback	65	348	22987.80	160.21
GPT builders	2,704	16,413	228394.40	354.28
Documentation	336	1,594	157353.60	1073.32
Prompting	2,283	13,660	4019560	833.03
Total	29,576	169,928	4486547.40	1286.20

The analysis of topic popularity by category provides insights into the interests and priorities of the OpenAI Developer Forum community. It highlights **the significance of API-related discussions and showcases the community's enthusiasm for real-world applications of AI technologies**. Furthermore, the examination of the scoring system sheds light on the factors that contribute to the visibility and engagement of posts within the forum.

C. User Contributions by Trust Level

To further understand the popularity trends on the OpenAI Developer Forum, it is essential to examine the distribution of user contributions across different trust levels. The forum employs a trust level system [92] to categorize users based on their engagement and contributions. The trust levels are defined as follows: Level 0 (`Newuser`), Level 1 (`Basic`), Level 2 (`Member`), Level 3 (`Regular`), and Level 4 (`Leader`). Higher trust levels represent a greater degree of participation and influence within the community.



(a) User distribution by trust level. **(b) User post contributions by trust level.**

Fig. 3: User distribution and contributions by trust level.

Figure 3 illustrates the user distribution and post contributions by trust level. The majority of users (26,998) are in the `Newuser` category (trust level 0), but they account for a relatively small portion of the total post count (37,156).

Interestingly, as trust levels increase, the number of users decreases significantly, but their individual contributions grow substantially. For instance, the 75 users in the `Leader` category (trust level 4) have authored an impressive 25,403 posts, highlighting **the pivotal role played by a small group of highly engaged and trusted users in driving discussions and generating valuable content**.

V. RQ2: Taxonomy of Concerns

In this section, we construct a taxonomy based on 886 sample topics from the OpenAI Developer Forum, as shown in Figure 4. The taxonomy comprises three main categories: `API/ChatGPT` (291 topics), `GPT builders` (214 topics), and `Prompting` (212 topics). Additionally, 169 topics were unrelated to developer concerns and excluded from further analysis. The percentages in Figure 4 represent the proportion of topics in each subcategory relative to the total number of topics in that category.

A. API / ChatGPT

As shown in Figure 4a, the `API` and `ChatGPT` categories encompass several key areas of interest. The most prevalent concern, accounting for nearly half of all discussions, is **API issues** (49.5%). The second most significant area, **specific functionality** (26.5%), focuses on specialized features and their practical applications. **Model performance** (18.5%) is the third major topic, addressing the efficiency and accuracy of generated outputs. Lastly, **security, privacy, and bias** (5.5%) cover critical aspects of data protection and ethical usage.

1) API issues

Developers frequently encounter various API issues, including **request and response errors**, **rate limits and quotas**, **streaming responses**, and **file processing** challenges. For instance, the GPT 3.5-Turbo API can randomly hang indefinitely [1], and a bug in the `Assistant API` causes the temperature to default to 1 when set to 0, while other values work as intended. Errors such as "Error: 413 The data value transmitted exceeds the capacity limit" when calling v1/images/edits [2] and inappropriate function calls from the GPT-3.5-turbo-1106 model [3] are also reported. Rate limits and quotas cause substantial delays, with response times averaging around 30 seconds [4], and issues like insufficient quota errors on paid accounts [5] and billing bugs [6] persist. Streaming responses face performance decreases with large instructions [7], missing characters [8], and issues handling image URLs [9]. File processing problems include the code interpreter's inability to find uploaded files [10], persistent file appearance in lists despite deletion [11], and errors like "openai.BadRequestError: Error code: 400" during uploads [12].

2) Model performance

Many posts report **a noticeable decline in GPT-4 and GPT-4o's performance, especially via the API**, over the past few months [13]. Developers have experienced issues with context retention, code generation, and accuracy, similar to those in models like GPT-3.5-turbo and DALL-E. For example, GPT-4o often forgets its own modifications when generating Python code and fails to follow simple instructions [13]. It also introduces inconsistent logic and references non-existent variables and functions [13]. Additionally, there are complaints about outdated or incorrect code [14]. Developers have also reported instances of ***over-refusals, where GPT-4o and other models refuse to perform tasks well within their capabilities*** [15]. Common examples include refusing to handle PDF files, generate images or charts, or run code due to assumptions about missing Python packages. For instance, GPT-4 Turbo refused to refactor code due to an assumed platform limit, even though the task was within token limits [16]. Another example involved the model refusing to directly convert Lua scripts to Python, requiring multiple prompts to clarify the request [16]. Developers have also noted instances where the model refuses to work with "copyrighted material" or denies having analytical tools, even when such tasks should be feasible, leading to significant productivity loss [17].

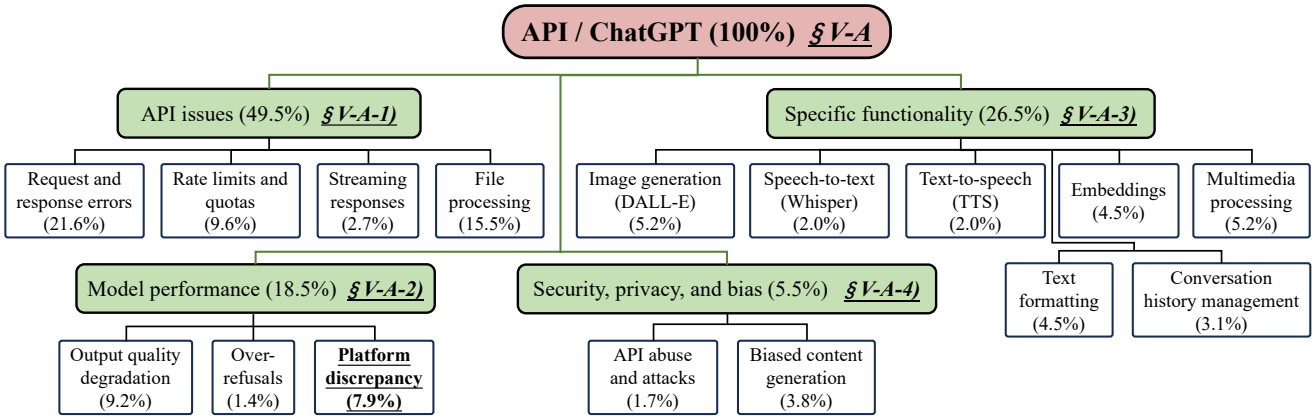
> ***Insight 1:*** *Increasingly, state-of-the-art LLMs [94], [109] exhibit over-refusals, suggesting a potential research direction to explore the root causes of these occurrences and how to mitigate them.*

Platform discrepancy. Significant performance differences exist between the API and web-based (ChatGPT) versions of GPT-4. Despite identical prompts and parameters, the API output is often considered less intelligent or relevant compared to the web version. For example, the GPT-4 API's outputs are described as "way worse" than those from ChatGPT on the website, even after experimenting with system prompts and different temperatures [18]. The API sometimes fails to retrieve information from files, returning responses like "I don't have any information about this" [19]. Additionally, the GPT-4 Turbo model generates random garbage text when using the async API, an issue not seen with GPT-4o or GPT-3.5 Turbo [20]. Concerns also exist about the API embedding engine performing worse than the one used by ChatGPT [21]. Table III illustrates that various ChatGPT applications on different platforms have their own specific issues. For instance, audio responses on the iOS app can be choppy and unclear, making it difficult to understand the voice content [22]. On the Mac OS app, developers cannot view conversation text when using the voice feature [23]. Additionally, there are problems with image generation/upload on the Mac and iOS apps, and inconsistencies in accessing old responses on Android.

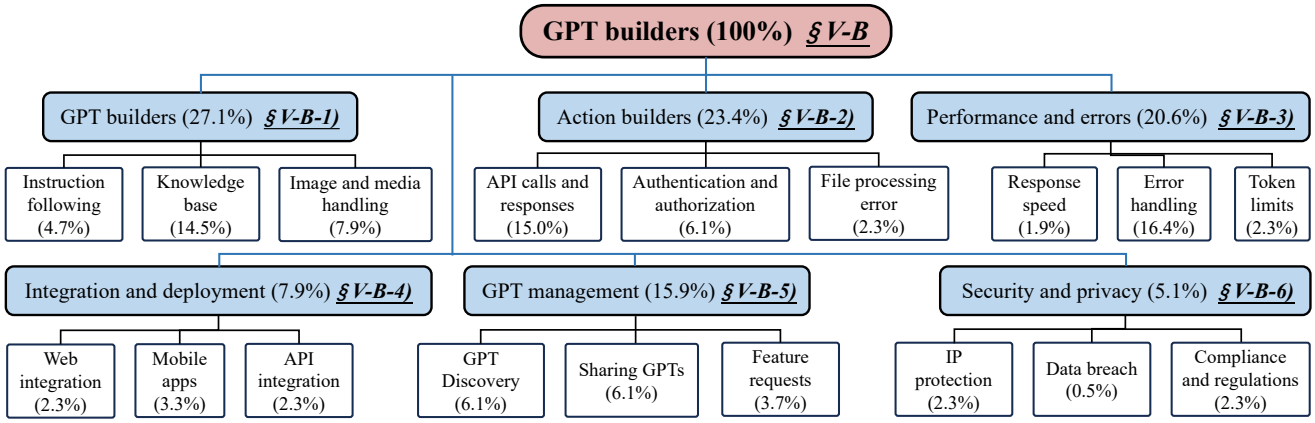
> ***Insight 2:*** *The API version of GPT-4 often underperforms compared to the web version (ChatGPT). Furthermore, there are noteworthy issues with ChatGPT applications across different platforms, as shown in Table III.*

3) Specific functionality

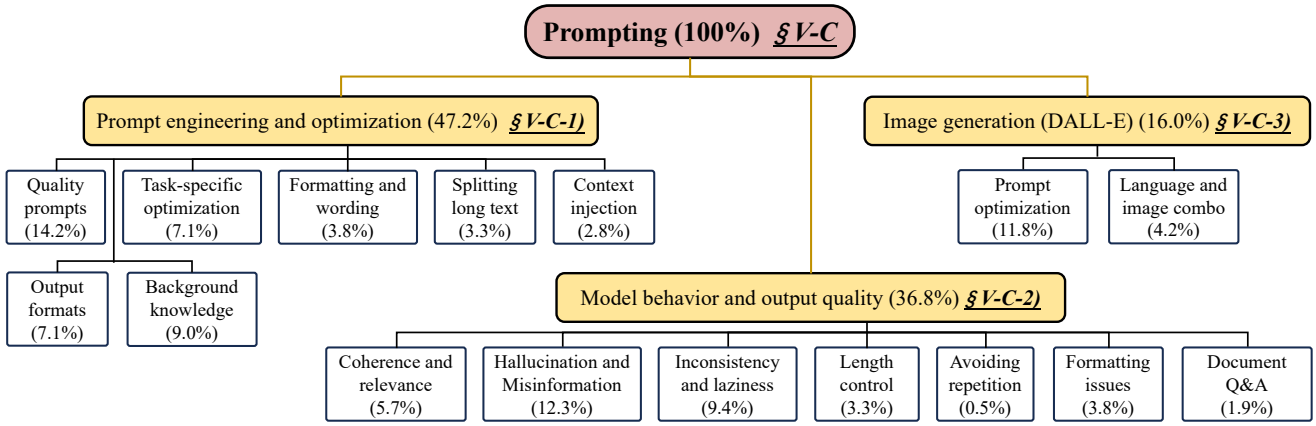
Several widely-used functionalities, including **DALL-E**, **Whisper**, **TTS-1**, and **Embeddings**, have encountered various issues. There is a significant drop in quality when using the `DALL-E API`, particularly with DALL-E3, where results are noticeably inferior to those from Bing and the original model [24]. The `Whisper API`'s performance has deteriorated over time, showing hallucinations and dropped sentences, especially with significant background noise. Attempts to clean audio files using `ffmpeg` have been insufficient, and developers suggest appending a pattern to the prompt to filter out random text generated during silent segments [25]. The `TTS-1 API` struggles with text extraction from PDFs



(a) Taxonomy of concerns in `API` and `ChatGPT` topics.



(b) Taxonomy of concerns in `GPT builders` topics.



(c) Taxonomy of concerns in `Prompting` topics.

Fig. 4: Taxonomy of concerns on OpenAI Developer Forum.

and produces tinny, distorted voices compared to the standard offering. Developers recommend first converting PDFs to text files and cleaning them up before processing through the `TTS` model. Additionally, there are issues with incomplete audio outputs and inconsistent responses for non-English requests [26]. `Embeddings` are cost-efficient, but developers face batch processing rejections due to a full queue, inconsistent `embeddings` for the same content, and performance differences between small and large models. These issues impact tasks such as search, clustering, recommendations, anomaly detection, diversity measurement, and classification [27].

Multimedia processing. Developers have reported several problems when processing multimedia using OpenAI's API. One common issue is the "invalid_image" error during batch processing of images stored in public S3 buckets, despite the images being valid and accessible [28]. This issue appears random, with different requests failing in each batch. Some developers speculate that the API struggles with certain JPEG file structures and use services like Cloudinary to preprocess

TABLE III: Issues across different ChatGPT applications on various platforms.

Category	Issue	Win	Mac	Andr.	iOS	Web
Account & Subscription	Subscription/billing			✓	✓	✓
	Login/authentication	✓	✓	✓	✓	✓
	Limited GPT-4 access	✓		✓	✓	✓
	Access denied		✓			✓
Performance & Technical	Slow response		✓	✓		✓
	Errors/crashes	✓	✓	✓	✓	✓
Functionality	Image generation/upload		✓		✓	✓
	Custom GPT/plugins	✓	✓	✓	✓	✓
	PDF/file reading					✓
	Copy/paste		✓	✓		
	Search		✓	✓		
	Model switching			✓	✓	
	Voice chat	✓	✓	✓	✓	
	Hyperlink retrieval		✓			
	Screen sharing		✓			
	Text-to-speech		✓		✓	
	LaTeX rendering			✓	✓	
	Data export/history			✓		
	Access old responses			✓		
	Voice widget			✓		
	Image loading				✓	
	Shortcuts integration				✓	
	Voice input time limit				✓	
	Specific voices				✓	
UI/UX	UI/UX improvements	✓	✓		✓	✓
	Foreground, auto-scroll		✓			
	Desktop vs web differences	✓	✓			✓
	Text area/font size	✓	✓			
	Sidebar not showing					✓
Integration & Compatibility	App/shortcut conflicts	✓	✓		✓	
	Device compatibility		✓	✓	✓	
	Android studio integration			✓		
Privacy & Data	Content consistency					✓
	Data retention					✓
	App/extension data sharing				✓	
Miscellaneous	Download availability	✓	✓	✓	✓	
	Date/time settings		✓			
	Region/language support				✓	✓

images before sending them to OpenAI's API [29]. Additionally, intermittent failures occur when generating images with the `Assistant API` using a code interpreter to create graphs from CSV data, even though the code executes correctly, suggesting potential provisioning issues with the sandbox environments [30].

Text formatting. The AI sometimes fails to emit its stop sequence in "JSON-mode", leading to excessive text generation, and the GPT-4o API often produces invalid JSON responses [31]. Additionally, LaTeX formatting for matrices fails consistently [32], and copying text from Microsoft Word into the Mac desktop app results in the text being pasted as an image [33]. Uploading ".js" files also causes *BadRequestError* (error code 400) [12].

Conversation history management. Developers have reported issues with the `Assistant API` where ***responses are influenced by instructions from previously deleted assistants*** [34]. This problem manifests even after creating new threads and deleting old ones. In some cases, the `assistant`'s responses are skewed towards instructions from the earliest `assistant` created in the project. To mitigate this, developers have added extensive new instructions and used completely new `assistant` names and rephrased instructions for each iteration. Despite these efforts, the issue raises concerns about potential guardrails between different `assistants`. ***This issue reveals a disregard for user rights by OpenAI.*** The persistence of deleted instructions undermines user control and data privacy, raising serious ethical concerns about the platform's respect for user autonomy and consent. Such behavior not only violates basic principles of responsible AI development but also potentially infringes on data protection regulations, highlighting a critical need for transparency and accountability in OpenAI's data management practices.

> ***Insight 3:*** *The SE community can focus on developing robust AI integration frameworks and advanced testing methodologies for AI-powered applications. These efforts may address cross-platform inconsistencies, enhance multimedia processing, and refine conversation management.*

4) Security, privacy, and bias

API abuse and attacks. Developers have reported ***significant security vulnerabilities that have persisted for over a year***, allowing bad actors to hack accounts, create organizations using the victim's payment method, and demote the original developer to "Reader" status [35]. This demotion means the attacker takes over the "Owner" role within the organization, gaining full control over account settings, billing, and access management, while the "Reader" cannot affect the "Owner's" actions. This vulnerability may be caused by the developer's key leakage and flaws in OpenAI's authentication mechanism. Unlike OpenAI, other cloud platforms such as AWS, GCP, and Microsoft Azure implement stricter security configurations, including mandatory MFA for sensitive operations. It is crucial for developers to ***adopt best practices in key distribution*** and for OpenAI to enhance security measures to prevent such abuses and protect developer accounts.

> ***Insight 4:*** *Persistent security vulnerabilities in the OpenAI API account highlight the need for stricter security measures and best practices to prevent account takeovers and protect developer accounts.*

Biased content generation. Concerns have been raised about bias in OpenAI's products like DALL-E and ChatGPT, particularly regarding restrictions on generating illustrations for "Eid al-Adha" with Islamic symbols [36]. While the system cites content restrictions to avoid misuse, similar symbols from other religions (e.g., Christmas, Easter, Star of David) do not face the same limitations. This inconsistency suggests unequal treatment of cultural symbols, prompting calls for OpenAI to review and address these biases to ensure fairness.

B. GPT builders

The taxonomy of concerns within the `GPT builders` category encompasses several key areas, as shown in Figure 4b. **GPT builders**, accounting for 27.1% of the concerns, focus on creating and managing various GPTs. **Action builders** represent 23.4% of the concerns, dealing with API interactions and security measures. **Performance and errors** constitute 20.6%, addressing response speeds and error handling. **GPT management** accounts for 15.9%, involving the discovery and sharing of GPTs as well as handling feature requests. **Integration and deployment** make up 7.9%, covering the incorporation of GPTs into web and mobile applications. Although representing 5.1% of the concerns, **security and privacy** remain crucial, focusing on intellectual property (IP) protection and regulatory compliance.

1) GPT builders

Instruction following. Custom GPTs often struggle to follow instructions consistently. For example, despite being within the 128,000 token limit, some custom GPTs ignore certain instructions, causing frustration [37]. There are instances where GPTs disregard instructions regularly, affecting their reliability [38]. To mitigate these issues, it is recommended to provide concise and clear instructions, use specific dialog starters, and structure the instructions effectively.

Knowledge base. Issues with the knowledge base functionality are common among GPT builders. Files uploaded to the GPT's knowledge section occasionally disappear or are not utilized during interactions [39]. In some cases, financial GPTs fail to use the provided Quickbooks API knowledge, leading to incorrect query generation [40]. Integrating external databases via REST API can be technically challenging, although there are solutions to simplify this process [41]. Additionally, there are discrepancies between preview and live performance, with GPTs struggling to access and use data files consistently [42]. These challenges highlight the need for improved reliability and integration of knowledge bases in custom GPTs.

> ***Insight 5:*** *A potential research direction could explore performance improvements of GPTs with integrated knowledge bases compared to ChatGPT and identify failure points when the knowledge base is not effectively utilized.*

2) Action builders

GPT `Actions` allow developers to interact with third-party services by executing API calls, converting input to JSON schemas, and handling specified authentication.

API calls and responses. Issues with API calls and responses are common in GPT development. Problems include unsupported content types, inconsistent success between tools like Postman and GPT, and hallucinations of parameter values. Errors such as *UnrecognizedKwargsError* or *ApiSyntaxError* occur when GPT-generated requests deviate from the expected format [43]. Additionally, the requirement for matching root domains in OAuth2 setups can hinder integration with services like Yahoo [44]. These challenges necessitate robust debugging tools and more flexible API handling mechanisms.

Authentication and authorization. Developers frequently encounter significant challenges related to authentication and authorization when integrating GPT with third-party services. Issues such as changing callback URL, infinite auth loops, and OAuth errors like "Missing token" [45], [46] are common. Problems with API key validation, misconfigured endpoints, and permissions-policy headers further complicate the process [47], [48].

> ***Insight 6:*** *For developers with limited programming experience, these obstacles make seamless integration challenging, suggesting the need for a comprehensive support toolset to facilitate smoother API calls and authentication handling.*

File processing errors. Common issues include validation errors indicating improperly passed parameters, multipart/form-data upload failures leading to 422 errors despite successful tests with tools like cURL and Postman, and empty responses when trying to download files through redirects [49]. Additionally, there are problems with GPT not recognizing uploaded files in the conversation context, and generated download links being non-clickable in responses [50]. These difficulties highlight the need for improved documentation and tools to streamline file-handling processes in GPT integrations.

3) Performance and errors

Developers frequently encounter **delays in GPT response times**, particularly when using services like Ngrok, which can introduce over 30 seconds of latency [51]. Additionally, there is a need for features to track the percentage of received responses during streaming to enhance user experience by buffering appropriately [52]. **Error handling** is also challenging in GPT integrations, especially when dealing with large responses that exceed processing limits [53]. For instance, custom `Actions` like accessing Google Calendar events can result in responses too large for GPT to handle, regardless of time range limitations [53]. Furthermore, **token limits** present significant hurdles for GPT development. Developers face frustrations with request caps, which can halt progress for hours after sending a relatively small number of messages [54]. The lack of guidance on managing these limits effectively leads to widespread developer dissatisfaction.

4) Integration and deployment

Developers have reported several critical issues affecting the integration and deployment of GPTs with `Actions`. URLs served by custom `Actions` are not clickable on desktop but work fine on mobile devices, with errors related to the Permissions-Policy header [55]. Additionally, a critical bug has affected GPTs with `Actions` on the ChatGPT app for Android since early May, causing them to malfunction despite successful API interactions, which does not occur in the browser version [56]. There are also issues with function calling, particularly on Android devices like Samsung, where permissions and security settings cause persistent errors [57].

> ***Insight 7:*** *Integration of GPTs with `Actions` face critical issues, especially on Android, and accessing custom GPTs via APIs or external applications is a possible extension.*

5) GPT management

The discoverability of GPTs in the store is currently limited, as only the top 12 GPTs are displayed, making it difficult for users to find lesser-known but valuable GPTs. Suggestions include showing all GPTs in a category with filtering options and implementing user recommendations based on usage history [58]. Additionally, promoting GPTs with added value, such as those featuring `Actions` and knowledge files, over those popular due to their names, could help users explore and benefit from enhanced functionalities [58]. Furthermore, developers have reported issues with **sharing custom GPT** links, including login loops and redirection to the basic ChatGPT 3.5 interface instead of the intended custom GPT, causing confusion and undermining the value of custom GPTs, especially for non-Plus users [59]. There are also **feature requests** for enhanced capabilities, such as allowing AI to interact with web content more effectively by segmenting web pages with accessibility grids and providing dashboards to show statistics of their GPTs directly from the GPT website or app [60].

Insight 8: *Exploring methods to enhance the added value of GPTs and optimizing recommendation systems are crucial for improving discoverability and user engagement.*

6) Security and privacy

IP protection. Protecting the intellectual property (IP) of GPT models is a significant concern for developers. ***One major risk is the potential for instructions to be stolen, allowing others to create and publish similar GPTs.*** Developers are particularly concerned that their carefully crafted instructions could be exposed if system prompt protections are bypassed [61]. Issues such as instruction leakage and prompt injection can compromise the security and privacy of custom GPTs [62]. Despite robust strategies, it is difficult to ensure complete immunity against "cracking" attempts. Developers must stay vigilant and continuously update their protection methods. Community contributions and collaborative efforts are vital in refining these security measures [63]. ***Implementing structured prompts and utilizing tools like Prompt-Defender can provide additional layers of security*** [64].

Insight 9: *Protecting GPT intellectual property is crucial, requiring constant vigilance and updates against risks like instruction theft and prompt injection.*

Data breach. Security and privacy breaches between plugins present a significant risk. For instance, users may inadvertently share sensitive data across plugins, such as confidential medical information between DrSmithPlugin and DrJohnsonPlugin, or legal strategies between HusbandLawyerPlugin and WifeLawyerPlugin [65]. This innate issue arises from either granting ChatGPT the autonomy to route messages for optimal performance or restricting it to protect user privacy. Although plugins have been discontinued, similar concerns may arise with GPTs. If a user engages multiple GPTs within a single conversation, there is a risk of data being shared between them. Additionally, enabling multiple third-party services within GPTs may lead to data being shared across these services.

Insight 10: *Data breaches between GPTs and third-party services may pose risks, necessitating stringent privacy protections to address potential data-sharing concerns.*

Compliance and regulations. Navigating compliance and regulatory issues is crucial for GPT development and deployment. Developers face challenges such as unexpected GPT removals due to policy violations, prolonged appeal processes, and functionality restrictions possibly influenced by competitive strategies [66].

C. Prompting

The concerns within the `Prompting` category, as shown in Figure 4c, encompass three primary areas. **Prompt engineering and optimization**, accounting for 47.2% of the concerns, focuses on enhancing prompt quality, task-specific optimization, and appropriate formatting. **Model behavior and output quality** represents 36.8% of the concerns, addressing critical issues such as coherence, relevance, hallucination, and misinformation in model outputs. The remaining 16.0% is dedicated to **image generation** (DALL-E), which involves prompt optimization specifically for image creation and the integration of language with visual elements.

1) Prompt engineering and optimization

Many developers have expressed concerns about the quality of prompts used in LLMs. Creating high-quality prompts is essential for eliciting accurate and relevant responses. **Task-specific optimization** involves tailoring prompts to each task's unique requirements, ensuring the AI understands the context and desired outcome [67]. Proper formatting and wording can significantly impact prompt clarity and effectiveness, often requiring iterative refinement [68]. **For lengthy texts, splitting them into manageable sections** helps maintain coherence [69]. **Context injection** involves embedding relevant background information directly into the prompt [70]. **Specifying output formats** clearly ensures the generated content meets expected standards [71]. Providing **background knowledge** within the prompt can enhance the AI's understanding and output quality [72]. Developers face challenges with version control and testing prompt variations, often relying on direct API calls and GitHub for sharing [73]. There's a demand for specialized tools for prompt management and testing, akin to a "CodePen for prompts" [73]. Techniques like chunking and context summarization help manage long content without exceeding token limits.

Insight 11: *There is a need for specialized tools and techniques to optimize prompts and manage prompt variations for better LLM responses.*

2) Model behavior and output quality

Developers have raised concerns about **coherence and relevance issues, hallucinations, inconsistency, and laziness in model outputs**. GPT-4 has been reported to struggle with rotated text in images [74], forget the previous context, and generate nonsensical or incomplete outputs despite negative feedback. Hallucinations have been observed when processing non-structured data from CSV files [75] and when chat history becomes too long or there are too many input sources [76]. Inconsistency in performance has been noted, with ChatGPT providing different answers to the same task or refusing to complete tasks after a certain number of requests [77]. Models have shown reluctance to browse web pages, provided overly lengthy responses with unsolicited code, and omitted necessary example code when updating [78]. Complex prompts have led to deteriorating results, and models have been found to override user-defined instructions, resulting in suboptimal outputs that fail to address users' needs comprehensively [79].

Another set of concerns raised by developers revolves around **length control, repetition avoidance, formatting issues, and document question-answering**. Users have reported difficulties in controlling the output length of models, with GPT-4 generating overly long responses or failing to extract all relevant categories from a given text. Repetition has been observed in model outputs, particularly when dealing with low-quality audio signals in transcription tasks [80]. Formatting issues have arisen, with models including unwanted introductory phrases or failing to adhere to the desired output format, such as bullet points or tables [81]. Lastly, developers have noted inconsistencies and inaccuracies in document question-answering tasks, with models fabricating information, overlooking specific details, or struggling to process large JSON files effectively [82].

Insight 12: *Robust context management, advanced prompt engineering, and innovative approaches for dynamic output adjustment based on user feedback are needed. The community should explore efficient architectures for diverse data processing and develop comprehensive evaluation frameworks to improve model performance across tasks.*

3) Image generation

Prompt optimization. Developers have expressed frustration with DALL-E's inability to consistently follow explicit instructions, such as avoiding the addition of text in generated images [83], maintaining consistency in character appearance and key objects across a sequence of images [84], or adhering to specific lighting and shadow requirements [85]. Users have also reported difficulties in generating full-body images of characters despite providing detailed descriptions [86].

Language and image combo. Challenges have arisen in seamlessly combining text generation and image generation within a single prompt, with developers struggling to create a coherent flow between the two modalities [87]. Additionally, users have encountered issues with DALL-E when attempting to create infographics, noting that the generated statistical data and text often contain errors, omissions, and spelling mistakes, requiring significant manual editing to rectify [88]. Developers have also sought guidance on extracting data from graphical images using ChatGPT.

Insight 13: *Researchers should investigate prompt optimization differences between image and text generation, enhancing DALL-E's instruction adherence. Focus areas include improving image generation model interpretation, developing multi-modal integration frameworks, and exploring techniques for text-image coherence and accurate visual information representation.*

VI. Implications

Our analysis of the OpenAI Developer Forum reveals implications for researchers, developers, and LLM providers.

A. Implications for Researchers

The analysis reveals several promising research directions for the academic community. The phenomenon of over-refusals in state-of-the-art LLMs warrants investigation into its underlying causes and potential mitigation strategies. Developing robust AI integration frameworks and advanced testing methodologies for AI-powered applications is crucial for addressing cross-platform inconsistencies and improving multimedia processing and conversation management. Researchers could explore performance differences between LLMs with and without integrated knowledge bases, identifying failure points and optimization strategies. The field of prompt engineering offers opportunities for developing specialized tools to optimize prompts and manage variations, potentially improving LLM response quality and consistency.

B. Implications for Developers

Our analysis highlights key focus areas for developers. They should be aware of performance discrepancies between different versions of LLMs (e.g., API vs. web versions) and address cross-platform inconsistencies. Implementing security measures is critical for preventing account takeovers and safeguarding API credentials. Developers should adhere to industry best practices in key distribution and management. When working with custom GPTs and `Actions`, developers should be mindful of potential integration issues and explore ways to enhance GPT value and discoverability. Finally, developers should protect their intellectual property when creating custom GPTs, implementing safeguards against instruction theft and prompt injection attacks.

C. Implications for LLM Providers

We identify several areas for improvement for LLM providers (especially OpenAI). They should address over-refusals in their models, balancing caution with functionality. Consistency across model versions and platforms is crucial, with API versions performing comparably to web versions. Providers should offer comprehensive support toolsets and clear documentation to aid developers and improve GPT integration with `Actions` on mobile platforms. Privacy and security should be prioritized, implementing stringent protections for data-sharing and strengthening measures to protect developer accounts (e.g., mandatory MFA for sensitive operations). This also includes addressing critical issues like the persistence of deleted instructions in the `Assistant API`, which raises ethical concerns about user autonomy and data privacy. Providers must ensure proper isolation between different `assistants` and respect user rights in data management. Finally, providers should develop robust context management systems, advanced prompt engineering tools, and innovative approaches for dynamic output adjustment, along with comprehensive evaluation frameworks to enhance model performance across various tasks.

VII. Threats to Validity

Omission of topic selection. One limitation of our study is the potential omission of topics related to developer concerns during the topic selection process. We directly filtered out topics belonging to the `Announcements`, `Community`, `Forum feedback`, and `Documentation` categories based on the official description of topic categories. However, these categories may still contain some bug reports or other issues posted by developers. To avoid any impact on the completeness of our taxonomy, we manually examined these topics, including 1,126 belonging to `Community`, 27 to `Forum feedback`, and 102 to `Documentation`. Our examination revealed that the majority of developer concerns are `API/ChatGPT` related (69 topics), followed by `GPT builders` (36 topics) and then `Prompting` related concerns (18 topics). Moreover, these concerns all fit within our taxonomy's categories, further validating its comprehensiveness.

Subjectivity of researchers. Another limitation is the subjectivity introduced by the researchers during the manual analysis. To mitigate this threat, two inspectors independently analyzed and labeled the sample topics. Any conflicting results were discussed with an experienced arbitrator until a consensus was reached. Fortunately, the independent labeling process yielded a high inter-rater agreement, demonstrating the reliability of our coding schema and procedure.

VIII. Conclusion

In this paper, we comprehensively analyzed the OpenAI Developer Forum, addressing two main research questions: the popularity trends and a taxonomy of developer concerns. By quantitatively analyzing forum metadata and qualitatively categorizing developer discussions, we identified key areas of interest, engagement patterns, and specific concerns raised by developers. Our findings provide valuable insights for future research, tool development, and best practices, ultimately enhancing AI-assisted software engineering and promoting the responsible integration of AI technologies.

References

[1] https://community.openai.com/t/gpt-3-5-turbo-api-call-randomly-hangs-indefinitely/293385, Accessed July 16, 2024.

[2] https://community.openai.com/t/bug-error-413-the-data-value-transmitted-exceeds-the-capacity-limit-when-calling-v1-images-edits/464513, Accessed July 16, 2024.

[3] https://community.openai.com/t/gpt-3-5-turbo-1106-model-consistently-responds-with-unnecessary-and-inappropriate-function-calls-confirmed-bug-jan-26/603102, Accessed July 16, 2024.

[4] https://community.openai.com/t/assistant-run-remains-queued-for-a-long-time-ii/547875, Accessed July 16, 2024.

[5] https://community.openai.com/t/api-key-bug-insufficient-quota-error-on-paid-account-with-available-balance/717097, Accessed July 16, 2024.

[6] https://community.openai.com/t/consistent-billing-bug-causing-account-to-be-negative/706641, Accessed July 16, 2024.

[7] https://community.openai.com/t/slow-stream-when-assistant-instructions-exceed-9-000-characters/851660, Accessed July 16, 2024.

[8] https://community.openai.com/t/bug-assistants-streaming-responses-thread-message-delta-events-characters-missing/797939, Accessed July 16, 2024.

[9] https://community.openai.com/t/gpt-4o-api-bug-cant-take-in-image-url-from-assistant-in-messages-only-user/748357, Accessed July 16, 2024.

[10] https://community.openai.com/t/file-upload-and-acting-on-it-in-an-assistant-v2-conversation/856370, Accessed July 16, 2024.

[11] https://community.openai.com/t/can-not-add-files-to-vector-store/738145, Accessed July 16, 2024.

[12] https://community.openai.com/t/bug-when-upload-js-file-server-replies-openai-badrequesterror-error-code-400/769418, Accessed July 16, 2024.

[13] https://community.openai.com/t/has-anyone-noticed-gpt4o-quality-drop-last-few-days/829203, Accessed July 16, 2024.

[14] https://community.openai.com/t/gpt-4o-performing-poorly-for-code-related-tasks-why/805602, Accessed July 16, 2024.

[15] https://community.openai.com/t/help-openai-fix-over-refusals/409799, Accessed July 16, 2024.

[16] https://community.openai.com/t/gpt-4-turbo-refusing-to-follow-instructions/548716, Accessed July 16, 2024.

[17] https://community.openai.com/t/refusal-to-work-with-copyrighted-material-in-4-turbo-0125-preview/671453, Accessed July 16, 2024.

[18] https://community.openai.com/t/is-api-gpt4-way-less-intelligent-than-chatgpt4/253218, Accessed July 16, 2024.

[19] https://community.openai.com/t/problem-of-retrieving-information-from-file-search-playground-vs-api-usage/722245, Accessed July 16, 2024.

[20] https://community.openai.com/t/async-api-random-garbage-text-with-gpt-4-turbo-but-not-gpt-4o-or-gpt-3-5-turbo/791641, Accessed July 16, 2024.

[21] https://community.openai.com/t/difference-between-gpts-and-using-api/780947, Accessed July 16, 2024.

[22] https://community.openai.com/t/audio-issues-on-ios-app-chatgpt/815847, Accessed July 16, 2024.

[23] https://community.openai.com/t/chatgpt-app-unable-to-view-conversation-text-while-using-voice-features/792894, Accessed July 16, 2024.

[24] https://community.openai.com/t/why-the-quality-of-dall-e3-api-is-significantly-lower-compared-to-the-original/492970, Accessed July 16, 2024.

[25] https://community.openai.com/t/whisper-hallucinations-dropped-sentences-help/473368, Accessed July 16, 2024.

[26] https://community.openai.com/t/seeking-advice-on-integrating-tts-1-api-for-reading-pdf-book-aloud/766983, Accessed July 16, 2024.

[27] https://community.openai.com/t/batch-api-custom-id-does-not-support-uuid/742152, Accessed July 16, 2024.

[28] https://community.openai.com/t/getting-invalid-image-url-in-batch-api/859709, Accessed July 16, 2024.

[29] https://community.openai.com/t/error-you-uploaded-an-unsupported-image-please-make-sure-your-image-is-below-20-mb-in-size-and-is-of-one-the-following-formats-png-jpeg-gif-webp/849277, Accessed July 16, 2024.

[30] https://community.openai.com/t/getting-intermittent-failed-runs-in-assistant-api-when-generating-images-with-code-interpreter/713062, Accessed July 16, 2024.

[31] https://community.openai.com/t/gpt-4o-api-giving-wild-responses/749391, Accessed July 16, 2024.

[32] https://community.openai.com/t/chatgpt-fails-applying-wrong-latex-formatting-on-matrices-100-of-times/853837, Accessed July 16, 2024.

[33] https://community.openai.com/t/cgpt-mac-desktop-bug-copy-text-from-mac-pasting-into-cgpt-chat-bar-appears-as-image/854119, Accessed July 16, 2024.

[34] https://community.openai.com/t/assistants-api-is-responding-based-on-instructions-of-previously-deleted-assistants/863634, Accessed July 16, 2024.

[35] https://community.openai.com/t/paid-account-hacked-someone-else-is-now-the-owner/299460, Accessed July 16, 2024.

[36] https://community.openai.com/t/feedback-on-bias-in-content-generation-for-cultural-symbols/822373, Accessed July 16, 2024.

[37] https://community.openai.com/t/why-does-my-custom-bot-not-follow-instructions-given-to-it/529666, Accessed July 16, 2024.

[38] https://community.openai.com/t/gpt-disregards-instructions-regularly/551858, Accessed July 16, 2024.

[39] https://community.openai.com/t/custom-gpts-knowledge-files-dont-get-persisted/546431, Accessed July 16, 2024.

[40] https://community.openai.com/t/gpt-is-not-using-the-knowledge-base-while-performing-actions/569057, Accessed July 16, 2024.

[41] https://community.openai.com/t/integrating-gpt-with-external-database-s/580313, Accessed July 16, 2024.

[42] https://community.openai.com/t/custom-gpt-performs-differently-in-preview-than-when-live/552586, Accessed July 16, 2024.

[43] https://community.openai.com/t/getting-strange-unrecognizedkwargserror-error/169600, Accessed July 16, 2024.

[44] https://community.openai.com/t/gpt-actions-error-auth-url-token-url-and-api-hostname-must-share-a-root-domain/491072, Accessed July 16, 2024.

[45] https://community.openai.com/t/issue-with-oauth-missing-access-token/601033, Accessed July 16, 2024.

[46] https://community.openai.com/t/gpt-oauth-callback-url-keeps-changing/493236, Accessed July 16, 2024.

[47] https://community.openai.com/t/error-with-permissions-policy-header-unrecognized-feature-document-domain/604653, Accessed July 16, 2024.

[48] https://community.openai.com/t/gpts-is-not-invoking-the-token-endpoint-to-get-access-token/531822, Accessed July 16, 2024.

[49] https://community.openai.com/t/download-file-through-redirect-with-external-api/848948, Accessed July 16, 2024.

[50] https://community.openai.com/t/creating-a-file-download-link/641616, Accessed July 16, 2024.

[51] https://community.openai.com/t/ngrok-delays-or-actions-delays/857902, Accessed July 16, 2024.

[52] https://community.openai.com/t/streaming-response-with-percentage/775167, Accessed July 16, 2024.

[53] https://community.openai.com/t/how-you-deal-with-response-limit-in-a-single-api-request/500506, Accessed July 16, 2024.

[54] https://community.openai.com/t/request-cap-makes-impossible-to-build-gpts/575136, Accessed July 16, 2024.

[55] https://community.openai.com/t/urls-served-by-action-gpt-not-clickable-in-desktop-but-are-clickable-in-ios-app/574476, Accessed July 16, 2024.

[56] https://community.openai.com/t/critical-bug-with-gpts-with-actions-in-the-chatgpt-app/737472, Accessed July 16, 2024.

[57] https://community.openai.com/t/function-calling-fails-just-on-android-samsung/857325, Accessed July 16, 2024.

[58] https://community.openai.com/t/feedback-discoverability-of-gpts-needs-improvement/585652, Accessed July 16, 2024.

[59] https://community.openai.com/t/custom-gpt-sign-up-bug-when-sharing-link/502216, Accessed July 16, 2024.

[60] https://community.openai.com/t/gpt-web-browser-method-im-new-is-there-a-reason-why-people-havent-thought-of-this/755741, Accessed July 16, 2024.

[61] https://community.openai.com/t/gpt-4o-broken-security-gpt-store-read-most-any-system-prompt-here-we-go-again/770720, Accessed July 16, 2024.

[62] https://community.openai.com/t/custom-gpts-gpt-store-and-instructions-protection/616927, Accessed July 16, 2024.

[63] https://community.openai.com/t/the-prompt-defender-initiative-advancing-gpt-safety-standards/587766, Accessed July 16, 2024.

[64] https://community.openai.com/t/what-are-the-latest-strategies-for-prevening-prompt-leaks/725650, Accessed July 16, 2024.

[65] https://community.openai.com/t/found-a-real-case-of-plugin-innate-security-privacy-breaches/469532, Accessed July 16, 2024.

[66] https://community.openai.com/t/shouldnt-custom-gpt-remember-what-im-talking-to/740289, Accessed July 16, 2024.

[67] https://community.openai.com/t/i-need-help-in-generating-reports-using-chatgpt-similar-to-given-template-structure/861669, Accessed July 16, 2024.

[68] https://community.openai.com/t/text-parsing-and-producing-the-stable-json-output/853059, Accessed July 16, 2024.

[69] https://community.openai.com/t/ways-to-deal-with-prompts-larger-than-models-context-length/854697, Accessed July 16, 2024.

[70] https://community.openai.com/t/assistant-api-dont-access-the-database/852560, Accessed July 16, 2024.

[71] https://community.openai.com/t/prompt-for-image-to-json-conversion/866883, Accessed July 16, 2024.

[72] https://community.openai.com/t/how-can-i-refer-to-a-specific-document-in-the-vector-store/766021, Accessed July 16, 2024.

[73] https://community.openai.com/t/tools-for-testing-prompts/305309, Accessed July 16, 2024.

[74] https://community.openai.com/t/worse-ocr-on-rotated-text/461827, Accessed July 16, 2024.

[75] https://community.openai.com/t/read-a-csv-file-and-parse-unstructured-data-line-by-line/22280, Accessed July 16, 2024.

[76] https://community.openai.com/t/llm-forgetting-part-of-my-prompt-with-too-much-data/244698, Accessed July 16, 2024.

[77] https://community.openai.com/t/some-of-the-main-bugs-in-mygpt-among-several/664193, Accessed July 16, 2024.

[78] https://community.openai.com/t/gpt4-not-browsing-the-web-or-is-very-reluctant-to-do-so/688884, Accessed July 16, 2024.

[79] https://community.openai.com/t/gpt-4-getting-lazy-again-prompted-it-for-detailed-respones-getting-short-responses-now/742558, Accessed July 16, 2024.

[80] https://community.openai.com/t/openai-api-error-prompt-issue/571739, Accessed July 16, 2024.

[81] https://community.openai.com/t/cant-control-new-generated-text-with-prompts/821841, Accessed July 16, 2024.

[82] https://community.openai.com/t/obtaining-correct-pdf-page-number-in-the-response-using-gpts/524103, Accessed July 16, 2024.

[83] https://community.openai.com/t/dall-e-is-illiterate-with-the-text-it-adds-in-images/570654, Accessed July 16, 2024.

[84] https://community.openai.com/t/prompting-images-in-chatgpt-4/691466, Accessed July 16, 2024.

[85] https://community.openai.com/t/dall-e-and-lighting-problems/829022, Accessed July 16, 2024.

[86] https://community.openai.com/t/issue-with-generating-less-hair-using-dall-e-model/786849, Accessed July 16, 2024.

[87] https://community.openai.com/t/inline-images-in-gpts-between-text/693200, Accessed July 16, 2024.

[88] https://community.openai.com/t/dall-e-is-sooo-bad-at-recognizing-letters-and-numbers-any-advice/507925, Accessed July 16, 2024.

[89] A. Abdellatif, D. Costa, K. Badran, R. Abdalkareem, and E. Shihab, "Challenges in chatbot development: A study of stack overflow posts," in *Proceedings of the 17th international conference on mining software repositories*, 2020, pp. 174–185.

[90] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[91] S. Ahmed and M. Bagherzadeh, "What do concurrency developers ask about? a large-scale study using stack overflow," in *Proceedings of the 12th ACM/IEEE international symposium on empirical software engineering and measurement*, 2018, pp. 1–10.

[92] J. Atwood, "Discourse trust levels," https://blog.discourse.org/2018/06/understanding-discourse-trust-levels/, 2018.

[93] Z. Chen, Y. Cao, Y. Liu, H. Wang, T. Xie, and X. Liu, "A comprehensive study on challenges in deploying deep learning based software," in *Proceedings of the 28th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering*, 2020, pp. 750–762.

[94] J. Cui, W.-L. Chiang, I. Stoica, and C.-J. Hsieh, "Or-bench: An over-refusal benchmark for large language models," *arXiv preprint arXiv:2405.20947*, 2024.

[95] J. Han, E. Shihab, Z. Wan, S. Deng, and X. Xia, "What do programmers discuss about deep learning frameworks," *Empirical Software Engineering*, vol. 25, pp. 2694–2747, 2020.

[96] X. Hou, Y. Zhao, Y. Liu, Z. Yang, K. Wang, L. Li, X. Luo, D. Lo, J. Grundy, and H. Wang, "Large language models for software engineering: A systematic literature review," *arXiv e-prints*, pp. arXiv–2308, 2023.

[97] OpenAI, "Assistants api overview," https://platform.openai.com/docs/assistants/overview, 2024.

[98] ——, "Chatgpt," https://chat.openai.com/, 2024.

[99] ——, "Introducing the gpt store," https://openai.com/index/introducing-the-gpt-store/, 2024.

[100] ——, ".net library," https://platform.openai.com/docs/libraries/dotnet-library, 2024.

[101] ——, "Openai," https://openai.com, 2024.

[102] ——, "Openai developer forum," https://community.openai.com, 2024.

[103] ——, "Openai developer platform," https://platform.openai.com/docs/overview, 2024.

[104] ——, "Vector databases," https://cookbook.openai.com/examples/vector_databases/readme, 2024.

[105] C. Rosen and E. Shihab, "What are mobile developers asking about? a large scale study using stack overflow," *Empirical Software Engineering*, vol. 21, pp. 1192–1223, 2016.

[106] C. B. Seaman, "Qualitative methods in empirical studies of software engineering," *IEEE Transactions on software engineering*, vol. 25, no. 4, pp. 557–572, 1999.

[107] P. K. Venkatesh, S. Wang, F. Zhang, Y. Zou, and A. E. Hassan, "What do client developers concern when using web apis? an empirical study on developer forums and stack overflow," in *2016 IEEE International Conference on Web Services (ICWS)*. IEEE, 2016, pp. 131–138.

[108] Z. Wan, X. Xia, and A. E. Hassan, "What do programmers discuss about blockchain? a case study on the use of balanced lda and the reference architecture of a domain to capture online discussions about blockchain platforms across stack exchange communities," *IEEE Transactions on Software Engineering*, vol. 47, no. 7, pp. 1331–1349, 2019.

[109] T. Xie, X. Qi, Y. Zeng, Y. Huang, U. M. Sehwag, K. Huang, L. He, B. Wei, D. Li, Y. Sheng *et al.*, "Sorry-bench: Systematically evaluating large language model safety refusal behaviors," *arXiv preprint arXiv:2406.14598*, 2024.

[110] T. Zhang, C. Gao, L. Ma, M. Lyu, and M. Kim, "An empirical study of common challenges in developing deep learning applications," in *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2019, pp. 104–115.