Generative AI shifts technical knowledge production toward recombinant novelty

Neha Sharma* Simin Li†

November 4, 2026

Abstract

Generative AI can substitute for human expertise on problems it can solve, but it may also increase demand for human assistance by allowing users to resolve earlier problems and reach subsequent problems sooner. We develop a model that characterizes these opposing effects and study them using Stack Overflow, a large technical question-and-answer community. We define a technical specialization as the unique combination of knowledge domains associated with a question and use ChatGPT's November 2022 release as a change in the effectiveness of an external problem-solving technology. Following the release, questions in established specializations decline sharply, by approximately 13.4% at the average post-treatment horizon. In contrast, questions involving previously unseen specializations increase by approximately 3.9%. This increase is driven primarily by new combinations of existing knowledge domains rather than the introduction of entirely new domains, and is especially pronounced for combinations involving niche domains. The shift toward novel questions is also stronger among new users, consistent with AI helping individuals progress to problems they previously had difficulty reaching. Novel questions continue to receive human answers, while the community's knowledge network becomes less centered on historically popular domains. Together, the results show that generative AI changes not only the demand for human assistance on existing problems but also the types of problems individuals bring to human experts. Demand for human expertise can therefore increase for new combinations of knowledge domains even as overall question volume declines.

Keywords: Knowledge communities, Networks, Generative AI, Large language models

*University of Pennsylvania, nehash@wharton.upenn.edu
†Tulane University, simin.li@tulane.edu

1 Introduction

Generative AI, and large language models (LLMs) in particular, have rapidly altered how knowledge is produced, searched, and used. These changes are especially evident in software development, where models can generate functional code, debug existing code, and explain unfamiliar tools. Early evidence also raises concerns about their effects on entry-level work across industries Eloundou et al. [2023], Schneider et al. [2024], Brynjolfsson et al. [2025a]. This technological shift raises a central question about the role of humans and their expertise. A large literature studying technological change after the introduction of computers examines which tasks are automated and which new tasks emerge, changing the composition of human work Autor et al. [2003], Autor [2015], Acemoglu and Restrepo [2019]. We ask a related question: *what kinds of knowledge tasks and problems continue to require human expertise once LLMs are widely available, and how do these problems differ from those that required human assistance before?*

We study this question using StackOverflow, a large online community where users seek help from other users with programming and other technical problems. Questions asked in the community provide a record of problems for which individuals seek human assistance. Prior work documents a substantial decline in activity on technical question-and-answer communities following the introduction of ChatGPT Burtch et al. [2024], del Rio-Chanona et al. [2024]. Yet users continue posting questions even though they can obtain assistance from LLMs at relatively low cost. Posting requires users to formulate their problems, follow community guidelines, and wait for human answers. Moreover, existing evidence suggests that LLM-generated answers are inadequate for some technical questions Kabir et al. [2024]. The questions that continue to be asked can therefore help us understand how the demand for human expertise is changing.

We develop a model to explain how improvements in problem-solving technology can change the questions individuals bring to a human community. Individuals undertake projects in which resolving one problem allows them to encounter subsequent problems. At each problem, they first attempt to resolve it using their own knowledge and the available technology and may seek human assistance if they remain unsuccessful. Individuals who cannot resolve a problem remain at that problem and cannot progress until it is resolved. Better technology reduces the need to ask humans about problems it can solve, but also helps individuals resolve earlier problems and reach

subsequent ones sooner. We show that questions about these subsequent problems can increase even while questions about earlier problems decline. At the same time, improved technology can reduce the expected total number of questions an individual needs to ask to complete a project. The model also allows human answers to provide data that can improve future problem-solving technology, which may further change the types of questions asked in the community.

Methodology. We examine these predictions using questions posted on StackOverflow from 2018 to 2023 and a regression-discontinuity-in-time design, also known as an interrupted time-series design McDowall et al. [2019], around the November 2022 release of ChatGPT. Each question is associated with community-verified tags describing the knowledge domains relevant to the problem. Building on the recombinant view of knowledge production Weitzman [1998], Fleming [2001], we define a *specialization* as the combination of tags attached to a question and a *novel question* as one whose specialization has not previously appeared in the community. We distinguish novel questions involving entirely new knowledge domains from those that combine domains already present before the question was asked. We also examine whether changes in novel questioning differ between new and existing users, whether these questions continue to receive human answers, and how the community's knowledge network changes. We observe the questions users post, although we cannot observe the earlier problems they resolved before reaching them.

Results. Our analysis yields three main insights. First, the post-LLM decline in question volume is highly uneven. Questions with established specializations decline by approximately 13.4% at the average post-treatment horizon, while questions with previously unseen specializations increase by approximately 3.9%. The increase in novel questions is therefore not simply a consequence of established questions becoming a smaller share of total activity. More questions involving new specializations are posted even as overall question volume falls. This pattern is consistent with the model's prediction that improved technology can reduce questions about earlier problems while allowing more individuals to reach subsequent problems.

Second, the increase in novelty is disproportionately *recombinant*, involving new combinations of knowledge domains that already existed in the community. Within these combinations, the shift is particularly pronounced for those involving niche, or long-tail, domains rather than only

the most popular domains. Thus, the increase in novel questions is not primarily driven by users asking about entirely new technologies introduced alongside LLMs. These changes are also visible in the community's knowledge network. Popular domains lose connections and centrality, while niche domains become relatively more prominent in the combinations of knowledge represented by public questions. This pattern is consistent with research on the role of unfamiliar combinations in knowledge production Uzzi et al. [2013], Fortunato et al. [2018], Wagner et al. [2011].

Third, both new and existing users become more likely to ask novel questions following the introduction of ChatGPT. Four months after the release, the estimated increase is approximately 8.5 percentage points for new users, compared with 4.9 percentage points for existing users. While new-user status is not a direct measure of technical knowledge, this difference is consistent with the possibility that improved problem solving helps users with less experience reach problems they previously had difficulty accessing. Novel questions also continue to receive human answers. In a companion comparison of human and GPT-3.5 answers, we find suggestive evidence that human answers perform especially well on novel questions. Together, these findings indicate that the new combinations appearing in the community continue to generate demand for human assistance.

Implications. Our findings suggest that generative AI changes the distribution of problems for which individuals seek human expertise. After ChatGPT, demand shifts away from established specializations and toward new combinations of existing knowledge domains, particularly those involving niche domains. This change need not reflect the invention of entirely new problems. As technology makes earlier problems easier to resolve, individuals can progress further and encounter combinations that had not previously appeared in the community. The model provides one explanation for this pattern, although our data do not reveal individual AI use or the sequence of problems that led to each question.

These results also raise questions about the future of online knowledge communities. Human answers to novel questions continue to provide data that may help future technologies resolve similar problems. At the same time, a decline in overall question volume reduces opportunities to produce new public answers. We do not observe whether the answers generated after ChatGPT are incorporated into later AI systems or whether the community can sustain human participation as routine questions decline. The evidence nevertheless shows that the demand for human assistance is

changing in both volume and composition. Understanding that change is important for explaining how human expertise and AI will continue to contribute to technical knowledge production.

1.1 Related Literature

Our study connects research on technological change and the allocation of expertise, online knowledge communities, and recombinant knowledge production. The central question is how improvements in problem-solving technology change the composition of problems that reach human experts.

Artificial intelligence, automation, and task allocation. Task-based models establish that technological change affects labor through the activities it substitutes for and the new tasks it makes economically valuable [Autor et al., 2003, Acemoglu and Autor, 2011]. Acemoglu and Restrepo [2019] formalize the opposing forces of displacement and reinstatement through the introduction of new tasks. Evidence on generative AI finds substantial productivity gains, often concentrated among less experienced workers, alongside marked heterogeneity in performance across tasks [Noy and Zhang, 2023, Brynjolfsson et al., 2025b, Dell'Acqua et al., 2023]. These findings motivate an analysis of how AI changes the work for which human expertise remains valuable.

The model's problem-solving structure draws on knowledge hierarchies, in which individuals resolve familiar problems and seek more knowledgeable specialists for exceptions [Garicano, 2000, Garicano and Rossi-Hansberg, 2012]. Ide and Talamàs [2025] extend this tradition to AI and study how machine capabilities reshape the organization of knowledge work. Related recent models examine the reallocation of scarce human expertise toward knowledge frontiers [Agrawal et al., 2026] and the role of complementary research tasks and bottlenecks in determining AI's effects on R&D. Our model adds sequential problem encounters to this problem-solving environment. Clearing an earlier bottleneck increases the probability that a user reaches a subsequent problem, even when the project and its possible specializations remain unchanged. This progression channel can increase demand for human assistance at downstream specializations as demand at established ones contracts. It also predicts a larger response among users with less independent problem-solving experience.

Online communities and decentralized knowledge production. Online knowledge communities rely on voluntary contributions that make individual expertise available as a reusable public resource [Wasko and Faraj, 2005, Faraj et al., 2016]. Research on participation examines group size, reputation, monetary incentives, and contributor careers [Zhang and Zhu, 2011, Xu et al., 2020, Wang et al., 2022], while work on epistemic position highlights the knowledge supplied by peripheral contributors [Safadi et al., 2021]. Closely related to our setting, Sharma et al. [2021] model the dynamics of asking, answering, and participation in an online Q&A community. We retain costly public asking and concentrate on the demand side, allowing the pool of problems reaching the community to depend on resolution through outside technologies.

Two studies provide the most direct empirical benchmarks for our Stack Overflow analysis. Burtch et al. [2024] document reduced visits and questioning after ChatGPT, with larger declines in popular programming topics. del Rio-Chanona et al. [2024] estimate a decline in public knowledge sharing relative to comparison communities and identify the potential loss of inputs for future AI training. Evidence from Wikipedia also shows that AI-enabled tools can redistribute knowledge production across communities [Zhu and Walker, 2026]. We examine the composition of the questions that continue to be posted. Using combinations of technical tags, we find that first-observed specializations increase relative to their pre-release trajectory, especially combinations involving existing niche domains, even as established questions decline. We also examine whether these new questions continue to elicit human answers.

The dynamic interaction between AI and shared knowledge has attracted renewed theoretical attention. [Acemoglu et al., 2026a] study how AI advice can weaken human learning incentives and the accumulation of collective knowledge. Acemoglu et al. [2026b] examine how AI aggregation feeds into social learning and the quality of shared information. Our knowledge-feedback channel operates through publicly answered problems becoming part of the stock available to later problem-solving technologies. The empirical analysis observes the production of those answers; their subsequent incorporation into AI remains a theoretical implication.

Recombinant novelty and knowledge production. A longstanding view of innovation emphasizes new combinations of existing ideas and components [Weitzman, 1998, Fleming, 2001, Uzzi et al., 2013]. Increasing specialization and the difficulty of integrating knowledge shape both the

search for solutions and their evaluation [Jones, 2009, Jeppesen and Lakhani, 2010, Boudreau et al., 2016]. Recent experimental work similarly studies the distinctive contribution of human crowds to novel solutions in the presence of generative AI [Boussioux et al., 2024]. These studies motivate our focus on previously unseen combinations of technical domains rather than treating the appearance of an entirely new domain as the only form of novelty.

Carnehl and Schneider [2025] develop a model in which researchers choose questions and discoveries shape subsequent inquiry, providing an especially relevant account of an evolving knowledge frontier. Our setting features a different route to novel questions. Users pursue technical projects, and resolving one problem reveals opportunities to encounter others. Improvements in AI can therefore bring previously latent combinations of familiar domains into public view. The resulting novelty is novel to the observed community, without requiring the invention of new domains or deliberate search for unusual combinations. Human answers to these questions can then enlarge the codified stock that supports subsequent AI problem solving, linking changes in the composition of public demand to a moving knowledge frontier.

2 A Model of AI and Public Knowledge Demand

The model builds on the problem-solving logic of knowledge hierarchies Garicano [2000], in which individuals address problems using their own knowledge and refer unresolved problems to more knowledgeable solvers. We adapt this framework to an environment in which users can also rely on an external problem-solving technology, such as AI, before seeking human assistance.

We explicitly endogenize the arrival of questions by allowing users encountering problems to be dependent on whether their earlier problems were resolved. Additionally, the questions asked in the online community can augment the data available for improving the future technology that captures the interplay between AI's problem resolution capability, knowledge production in online communities and users demand for human expertise.

2.1 Model Preliminaries

All individuals undertake projects consisting of sequences of potentially interdependent problems. Unlike conventional knowledge-hierarchy models, in which a distribution of problems is presented

to an individual for resolution, the set of problems encountered by a user depends on successful resolution of preceding problems. Each problem is characterized by its type $r \in \mathcal{R}$, where $\mathcal{R}$ denotes the finite set of possible problem types. Upon successfully resolving a problem of type r, an individual encounters a subsequent problem of type s with probability π_{rs}. If the problem remains unresolved, the individual cannot progress and remains at type r. We assume that the transition probabilities are exogenous and satisfy $\sum_{s \in \mathcal{R} \cup \{0\}} \pi_{rs} = 1$, where 0 denotes project completion.

Individuals are endowed with heterogeneous knowledge $e \in [0, 1]$, distributed according to $f(e)$. For each problem, an individual first attempts to resolve the problem by themselves using the technological tools available to the individual. This probability depends on the individual's knowledge level e, the amount of data available for this problem type D_r and the efficiency of the technology ϕ in solving the problem with the available data. We denote this by $p_{own}(D_r, e, \phi)$. We assume that for a given D_r and e, a more efficient technology like LLMs increases the probability of problem resolution relative to a simple web search. Similarly, more available data D_r on the problem type and the individual's expertise also can weakly increase the probability of the problem getting resolved.

If the individual resolves the problem, they receive a reward $v > 0$. However, if the individual fails to resolve the problem by themselves, they can seek assistance from a human knowledge community. Asking a question to other experts on such community can be costly either due to community guidelines, or posting requirements. We denote this cost as $c > 0$ and assume that the probability getting the problem resolved in the community is simply $p_h \in (0, 1)$. Observe that to keep the model parsimonious and isolate the economic mechanism, we let p_h to be independent of the problem type and the community dynamics. Thus, an individual with an unresolved question would post the question if $p_h v \geq c$.

Questions asked in the community. We model the generation of questions as a dynamic process with discrete time, indexed by t. In each period, a mass M of individuals enters at problem type r_0, with knowledge distributed according to $f(e)$ and each individual attempts to resolve at most one problem. Individuals are myopic, and those who fail to resolve their current problem remain at the same problem type in the following period. Thus, given $D_{r,t}$ data available for problem type r at the beginning of period t, and ϕ_t technology, the probability that an individual

with knowledge e successfully resolves a problem of type r in period t is

$$\sigma_{r,t}(e) = p_{own}(D_{r,t}, e, \phi_t) + \mathbb{I}\{p_h v \geq c\}[1 - p_{own}(D_{r,t}, e, \phi_t)]p_h. \tag{1}$$

Therefore, distribution of individuals across problem types evolves endogenously based on their resolution outcomes in the preceding period. Consequently, the mass of individuals with knowledge e encountering a problem of type r in period t is given as

$$\Lambda_{r,t+1}(e) = Mf(e)\mathbb{I}\{r = r_0\} + \Lambda_{r,t}(e)[1 - \sigma_{r,t}(e)] + \sum_{s \in \mathcal{R}} \Lambda_{s,t}(e)\sigma_{s,t}(e)\pi_{sr}. \tag{2}$$

Similarly, the number of questions asked in the human knowledge community depends on the mass of individuals encountering each problem type and their probability of failing to resolve it using their own knowledge and available technology. The total demand for human assistance for problem type r in period t is

$$Q_{r,t} = \int_0^1 \Lambda_{r,t}(e)[1 - p_{own}(D_{r,t}, e, \phi_t)]\mathbb{I}\{p_h v \geq c\}\, de. \tag{3}$$

Equations (2) and (3) characterize how the distribution of public questions evolves over time. Observe that an improvement in technology reduces the probability that an individual seeks human assistance for a given problem, while increasing the probability of resolving that problem and progressing to subsequent ones. Finally, given that a resolved question on the community augments the available data for that problem type, we assume that

$$D_{r,t+1} = D_{r,t} + \kappa p_h Q_{r,t}, \qquad \kappa > 0. \tag{4}$$

The parameter κ captures the contribution of successful human resolutions to data. Through this process, questions answered in the community can improve the effectiveness of the future technology.

2.2 Substitution, Progression, and Recombinant Questions

We next study how improvements in technology affect the demand for human assistance. An improvement in technology can reduce the need to ask questions in the community, while allowing

individuals to resolve problems that previously prevented them from progressing to subsequent problems. We characterize these two effects by examining how an increase in ϕ changes the questions asked in the community, for a given amount of available data $D_{r,t}$.

Proposition 1 (Selective expansion) *Consider two problem types s and r, where individuals initially encounter s and, upon successful resolution, proceed to r with probability $\pi_{sr} > 0$. Suppose $p_h v \geq c$ and $p_{own}(D_r, e, \phi) < 1$. An improvement in ϕ reduces the questions asked about s in period 0. However, the questions asked about r in period 1 increase if and only if*

$$\frac{(1-p_h)\frac{\partial p_{own}(D_s,e,\phi)}{\partial \phi}}{p_h + (1-p_h)p_{own}(D_s,e,\phi)} > \frac{\frac{\partial p_{own}(D_r,e,\phi)}{\partial \phi}}{1-p_{own}(D_r,e,\phi)}. \tag{5}$$

The condition in (5) shows that questions of type r increase when an improvement in technology allows more individuals to resolve the preceding problem s and reach r, and this increase is sufficiently large to offset the reduction in questions due to improved problem resolution at r. Thus, the same improvement in technology can reduce questions about existing problems while increasing questions about subsequent problems that individuals were previously less likely to encounter.

This result is particularly relevant for problems that require combining existing knowledge domains in new ways. Consider, for example, $p_{own}(D, e, \phi) = 1 - (1-e)\exp(-\phi D)$. Under this specification, an improvement in technology reduces the probability of needing human assistance by a larger proportion for problem types with more available data. If existing problems have more data available than a new combination of knowledge domains, AI can help individuals resolve these earlier problems and progress to the new combination, where they may still require human assistance. The increase in questions can be particularly pronounced for individuals with lower levels of knowledge, who are more likely to encounter difficulties with the preceding problems.

2.3 Aggregate Contraction and a Moving Frontier

The previous result shows that improvements in technology can increase questions about subsequent problems, even as questions about earlier problems decline. We now examine how these improvements affect the total number of questions individuals ask as they complete their projects. Since individuals remain at the same problem until it is resolved, a better technology allows them to progress faster and reduces the number of times they need to seek human assistance.

Proposition 2 (Aggregate contraction) *Suppose $p_h v \geq c$ and all projects eventually reach completion. Consider two technologies, with the same available data D_r, distribution of individuals f, and transition probabilities π_{rs}, such that $p_{own}(D_r, e, \phi^1) \geq p_{own}(D_r, e, \phi^0)$ for every problem type r and knowledge level e. If data and technology remain unchanged during each project, the expected total number of questions individuals ask before completing their projects is weakly lower under technology ϕ^1 than under ϕ^0.*

The result follows because improvements in technology reduce the expected number of times an individual needs to ask the community before resolving each problem. The transition probabilities remain unchanged, so the problems individuals eventually encounter are the same, although they may reach them sooner. Consequently, with a constant flow of new individuals, the long-run number of questions per period also decreases when technology improves, if the available data and technology remain unchanged.

The model shows how improvements in technology can reduce the total demand for human assistance while increasing questions about problems that individuals were previously less likely to encounter. We examine these predictions using questions posted on Stack Overflow before and after the introduction of ChatGPT. We first study how the number of questions changes for existing and previously unobserved combinations of knowledge domains. We then examine whether the changes are more pronounced among less experienced users and whether the questions that continue to be posted receive human answers.

3 Empirical Setting, Measurement, and Identification

3.1 StackOverflow as a problem-solving environment

We study the model's predictions using StackOverflow, a large technical question-and-answer community. Individuals post questions when they need human assistance to resolve problems they encounter while working on technical projects. Posting a question requires effort, and users must follow community guidelines. The questions posted on StackOverflow therefore allow us to observe the types of problems for which individuals seek human assistance.

We use the official Stack Exchange public data dumps[1] from January 2018 to December 2023

[1] `https://data.stackexchange.com/`

and retain all user questions (`PostTypeId = 1`). For each question, we collect its creation date, tags, question text, score, number of answers, and user information. We normalize tags by removing version numbers and alphabetically order them to identify unique combinations. Our final dataset contains approximately 9.33 million questions. Additional details on data processing and summary statistics are provided in Appendix 2.

3.2 Mapping the Model to the Data

We use the tags associated with each question to identify the problem types in our model. Each tag represents a knowledge domain, and a unique combination of tags represents a problem type r. We distinguish existing problem types from those that appear for the first time in the community, which we refer to as novel questions. We further distinguish new combinations of existing knowledge domains from questions involving entirely new domains.

Questions posted on StackOverflow measure the demand for human assistance $Q_{r,t}$, while answers and accepted answers allow us to examine whether these questions are resolved. We use the introduction of ChatGPT in November 2022 as an improvement in technology efficiency ϕ. We also compare new and existing users to examine whether the changes in questions differ by experience e.

We do not observe individuals' use of AI, their knowledge levels, or the problems they resolve before posting a question. Therefore, we use the model to interpret the observed changes in question demand rather than estimate its parameters.

3.3 Knowledge domains, specializations, and novelty

In the StackOverflow community, users who ask questions manually assign *tags* that best describe the knowledge domains a user must possess to answer their questions. These tags serve as a human-curated, community-moderated taxonomy that plays a central role in searchability, moderation, and knowledge organization. Throughout the study, we treat an individual tag as a *knowledge domain* and the combination of tags attached to a question as its *specialization*. For example, a question tagged `python`, `json`, and `elasticsearch` requires knowledge at the intersection of these three domains. Appendix 2 provides an example and additional details on the tag construction.

Let $q \in \mathcal{Q}(t)$ denote a question asked in month t, and let $s(q)$ be its specialization. We call

$s(q)$ *new* in month t if that exact specialization has not been observed in any prior month, and *existing* otherwise. We refer to a question carrying a new specialization as a *novel question*. This measure is defined using the complete observed history available before month t, rather than the contemporaneous popularity of the question.

We further decompose novel questions to distinguish the introduction of new knowledge domains from recombination among domains that already exist. In month t, a knowledge domain is *new* if it has never appeared in a question before. Among domains that already exist, we classify a domain as *niche* if its question count is at or below the median pre-treatment domain count and *popular* otherwise. Each novel question is then assigned to one of three mutually exclusive groups: *new-domain* if it contains at least one new knowledge domain; *niche-containing* if it contains no new domain but contains at least one niche domain; and *popular-only* if it recombines only existing popular domains. The latter two groups capture new combinations of existing knowledge domains.

We use user histories to distinguish a *new user*, who joins the community in the month in which the question is asked, from an *existing user*. To study knowledge supply, we use the number of answers, whether a question receives an answer or accepted answer, and the time to the first and accepted answer. Finally, to characterize the aggregate knowledge structure, we build monthly tag co-occurrence networks in which a knowledge domain is a node and two domains are connected when they co-occur in a question. The network construction and additional measures are described in Appendix 6.

3.4 GPT-3.5 release and empirical design

We focus on the public launch of ChatGPT 3.5 in November 2022. Earlier tools, including GitHub Copilot, had already introduced generative models into software-development workflows, but GPT-3.5 marked the first large-scale, general-purpose LLM made broadly available through a conversational interface. We use its public release as a discrete shock to the accessibility and effectiveness of an external problem-solving technology. We define

$$\text{GPT-3.5}_t = \mathbf{1}\{t \geq T^*\},$$

where T^* is November 2022.

We analyze monthly outcomes before and after the release using a regression-discontinuity-in-time (RDiT) specification, also referred to as an interrupted time-series design. Our baseline model is

$$y_{xt} = \beta_1 f(\tau_t) + \beta_2 \text{GPT-3.5}_t + \beta_3 \text{GPT-3.5}_t \times f(\tau_t) + \varepsilon_{xt}, \tag{6}$$

where τ_t is a continuous month variable centered on the GPT-3.5 release and $f(\tau_t)$ captures the underlying time trend. The interaction term permits the post-release effect to evolve gradually rather than imposing a one-time level shift. The post-release level and slope changes are captured by β_2 and β_3, and the treatment contrast at a given value of t is $\beta_2 + \beta_3 f(\tau_t)$. We use a linear time trend in the main specifications; Appendix 3 compares this specification with a quadratic trend.

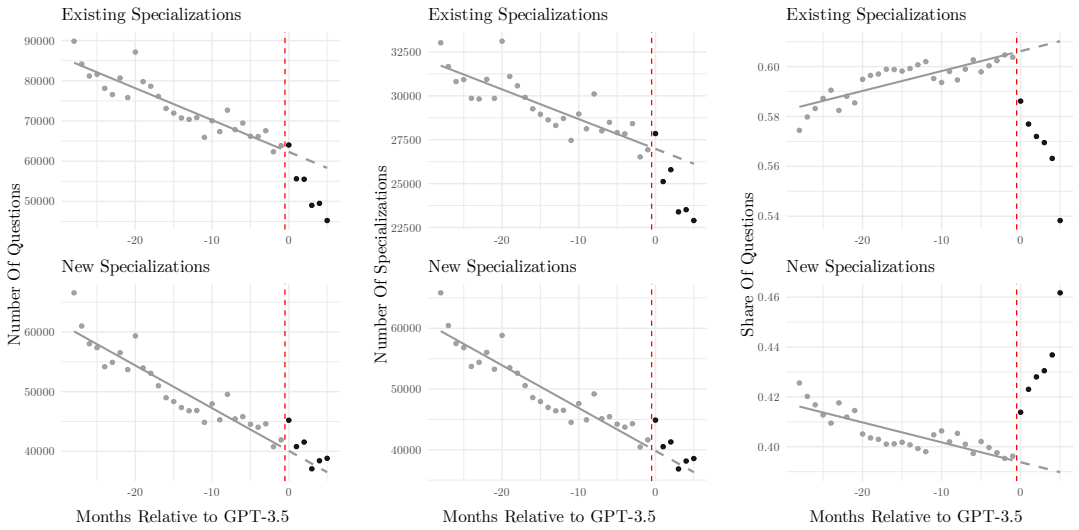
For analyses that compare groups or domain characteristics, we estimate

$$\begin{aligned} y_{xt} = {} & \beta_1 f(\tau_t) + \beta_2 \text{GPT-3.5}_t + \beta_3 \text{GPT-3.5}_t \times f(\tau_t) \\ & + \beta_{1n} f(\tau_t) M_{xt} + \beta_{2n} \text{GPT-3.5}_t \times M_{xt} + \beta_{3n} \text{GPT-3.5}_t \times f(\tau_t) \times M_{xt} + \varepsilon_{xt}, \end{aligned} \tag{7}$$

where M_{xt} denotes the relevant moderator. Standard errors in the main question-level specifications are clustered at the year-month level; the domain-level network analyses cluster at the domain and year-month levels.

Our key variables use the full history from 2018 through 2023, while the main RDiT analysis and displayed figures use the post-COVID period and end in May 2023. We exclude later months because subsequent model releases and platform changes make the GPT-3.5 break increasingly difficult to isolate. We also conduct placebo tests using a pseudo-treatment within the pre-period, compare alternative time trends, and apply the same analysis to Academia Stack Exchange; these checks are reported in the Appendix.

The empirical design should be interpreted in line with the model. We do not observe individual AI use or knowledge levels and do not estimate $p_{own}(D_{r,t}, e, \phi_t)$, p_h, c, κ, or π_{rs}. The estimates capture changes in questions posted to StackOverflow coincident with the GPT-3.5 release and its rapid diffusion. We use the model to interpret these changes rather than estimate its parameters.



(a) Number of Questions. (b) Number of Specializations. (c) Share of Questions.

Figure 1: The top panel of each figure illustrates outcomes for existing specializations and the lower panel illustrates outcomes for new specializations. Existing specializations experience a steeper decline than the temporal trend after the LLM shock, whereas novel questions are stable or increase relative to trend.

4 AI Substitution and Endogenous Problem Generation

Proposition 1 shows that an improvement in technology can reduce questions about an earlier problem while increasing questions about a subsequent one. Proposition 2 shows that better technology reduces the expected number of questions asked over a completed project under its assumptions. We examine how the number and types of questions posted on StackOverflow change after the GPT-3.5 release.

4.1 Established specializations: evidence of substitution

Figure 1a presents model-free evidence that the post-2022 decline in StackOverflow activity is highly uneven across types of knowledge demand. Questions with existing specializations decline significantly after the GPT-3.5 release. Figure 1b shows that the same pattern appears in the number of established specializations active each month.

The interrupted time-series estimates confirm this decline. At $t = 4$, we estimate a drop of 13.8% in questions with existing specializations relative to the pre-treatment period baseline. Questions involving combinations already observed in the community therefore decline after GPT-3.5, consistent with the substitution effect described in Proposition 1.

	Total Questions	New Specialization
	(1)	(2)
	OLS	Logit
GPT-3.5	-666.5	0.079***
	(1,976.2)	(0.010)
GPT-3.5 $\times t$	-2,598.6***	0.036***
	(476.7)	(0.006)
GPT-3.5 $\times$ New Spec.	4,002.1***	
	(728.0)	
GPT-3.5 $\times$ t $\times$ New Spec.	2,070.1***	
	(258.2)	
Observations	68	4,039,806

Table 1: Heterogeneous Impact of LLM on Question Traffic by Specialization Types
This table presents estimation results from the RDiT model on both the total number of questions and the probability of a question carrying new specifications. Model (1) further contains t, *New Spec.*, t $\times$ *New Spec.* as the controls; while model (2) further contains t. Standard errors are clustered at the year-month level. ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, †: $p < 0.1$.

This result is consistent with prior evidence that overall StackOverflow question volume fell after the introduction of generative AI. Our contribution here is to separate that overall decline according to the specialization carried by each question. The decline is concentrated in established specializations rather than being uniform across the knowledge space.

4.2 Novel specializations: evidence of problem generation

The pattern for novel questions is different. Questions carrying new specializations, as well as the set of new specializations active each month, are stable or increase relative to the pre-release trend. At $t = 4$, we estimate a significant 2.8% increase in questions with new specializations. In other words, while activity in existing specializations contracts sharply after the LLM shock, questions involving new specializations show resilience and increase relative to the month prior to the GPT release.

The decline in established questions and increase in novel questions are consistent with the two effects in Proposition 1. Technology can help individuals resolve earlier problems and reach subsequent ones, while also reducing the need to ask about problems they can resolve themselves. The increase in novel questions is not simply a consequence of the decline in established questions. A change in composition can raise the *share* of novel questions, but cannot by itself increase their

absolute number.

The uneven change in levels also produces a sharp shift in question composition. As shown in Figure 1c, the share of novel questions rises immediately after the GPT-3.5 release. Our interrupted time-series estimates show that the novelty share increases by **10.1** percentage points within four months of the release and, by May 2023, reaches **46.2 %**. This shift is driven by both the decline in questions using established specializations and the increase in questions using new specializations.

Appendix 3 reports placebo estimates, alternative polynomial time trends, and monthly Lorenz curves. The placebo break does not reproduce the main post-GPT pattern, and the linear and quadratic specifications yield comparable treatment contrasts. The Lorenz curves provide a complementary description: after the GPT-3.5 release, a smaller share of questions is concentrated in the most frequently observed specializations and more activity shifts toward the long tail of less frequently used specializations.

4.3 What Kinds of Problems Do Users Reach?

The previous section establishes that novel questions do not merely become a larger share of a shrinking platform; their absolute number also increases. The model provides a more specific implication about the composition of this growth. When external assistance removes bottlenecks in established component problems, downstream demand can arise through new interactions among those existing components. We therefore ask first whether the observed novelty comes from entirely new knowledge domains or from new combinations of domains that already exist, and then whether the change is stronger among users for whom intermediate bottlenecks are more likely to bind.

4.3.1 New knowledge domains and new combinations of existing knowledge

A novel question can arise through two distinct pathways. It can arise when a user asks a question that contains an *entirely new domain*, so that the question combines new knowledge with existing domains and creates a new specialization. Alternatively, novelty can arise when a user asks a question that *recombines* only existing domains that have not previously co-occurred.

We decompose a novel question into one of three mutually exclusive types: (i) *new-domain*, where the question contains at least one new knowledge domain; (ii) *niche-containing*, where the question recombines existing niche domains with other existing knowledge domains in a new way;

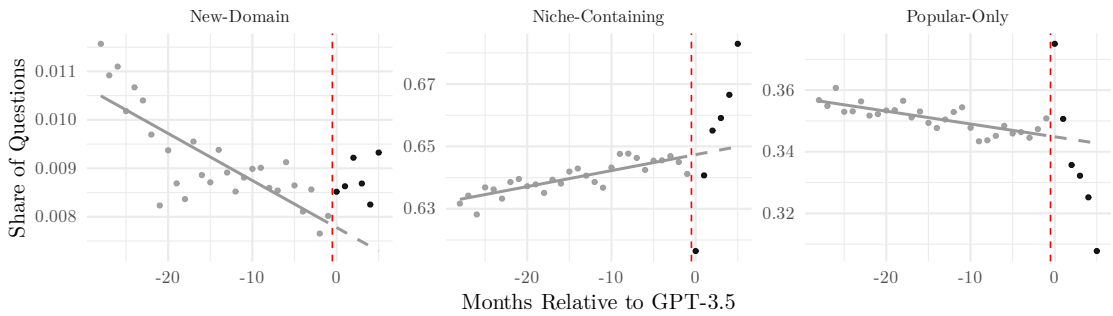


Figure 2: Change in the composition of novel questions in StackOverflow. The vertical line in November 2022 represents the LLM shock.

and (iii) *popular-only*, where the question recombines only existing popular domains in a new way. A new-domain question may involve an emerging package or technology that is new to the community. A niche-containing question may combine a lesser-known database or operating system with another existing domain, whereas a popular-only question combines well-established tags that had not previously occurred together.

Specialization Type (baseline: Popular only)	New-Domain	Niche-Containing
GPT-3.5	0.030	-0.104^{***}
	(0.043)	(0.008)
GPT-3.5 $\times$ t	0.053^{***}	0.052^{***}
	(0.013)	(0.003)
Change in $\mathbb{P}(d_{s(q)=\cdot,t=4})$	$+0.14^{***}$	$+2.19^{***}$
Change in $\mathbb{P}(d_{s(q)=\cdot,t=7})$	$+0.18^{**}$	$+5.43^{***}$
Observations	1,654,431	1,654,431

Table 2: Effect of GPT-3.5 on Novel Question Composition

This table reports estimates from the multinomial logit model, with popular-only novel questions as the baseline category. All specifications include an intercept and a linear time trend t. Standard errors are clustered at the year-month level. The last two rows show the changes in predicted probabilities (percentage points, pp) for each outcome when GPT-3.5 is introduced, evaluated at $t = 4$ and $t = 7$, with significance assessed via the delta method. ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, †: $p < 0.1$.

Figure 2 illustrates the shift in the composition of novel questions. Questions that combine only popular domains decline sharply as a share of novel questions. The composition shifts significantly toward niche-containing questions and more modestly toward new-domain questions. The multinomial estimates show that, four months after the GPT-3.5 release, the probability that a novel question is niche-containing rises by about **2.19** percentage points, compared with a smaller **0.14**

percentage-point increase for new-domain novel questions. The corresponding decline is borne by popular-only novel questions.

Thus, the post-GPT increase in novelty is not explained only by the arrival of technical domains created by the new technology itself. Novel questions are disproportionately recombinant and increasingly draw on infrequently used existing knowledge domains. This pattern is consistent with the progression effect in Proposition 1. When technology helps individuals resolve earlier problems involving existing domains, they can reach new combinations of those domains and ask the community for assistance.

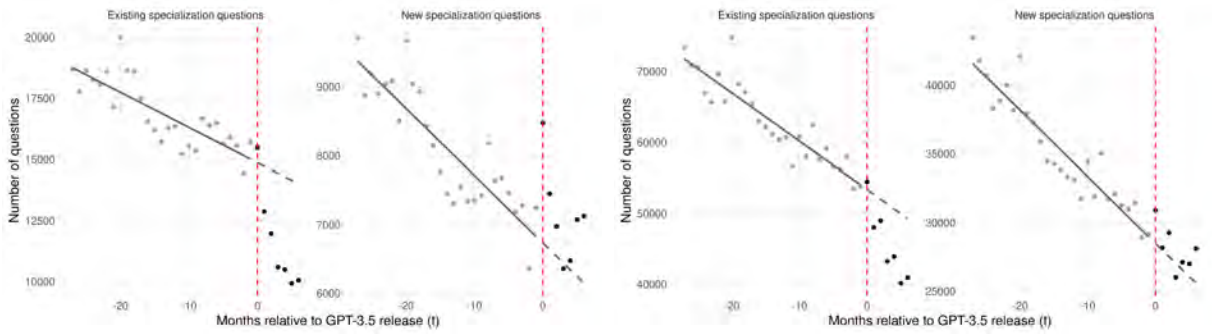
The result is not unique to the exact post-treatment break. Falsification tests in Appendix 4 do not reproduce the same shift during the pre-LLM period. As a separate validation exercise, Appendix 8 compares GPT-3.5-generated answers with StackOverflow reference answers on a balanced sample of general and novel questions. Human reference answers win 65.9% of pairwise comparisons on novel questions, compared with 62.2% on general questions. We treat this exercise as supporting evidence only; the main evidence remains the observed change in the composition of public knowledge demand.

4.4 Who contributes the novel questions?

Individuals with less knowledge may have more difficulty resolving earlier problems, so improvements in technology may help them reach more subsequent problems. We compare the change in novel questioning for new and existing StackOverflow users. Whether a user is new to the community measures experience on StackOverflow, not the individual's knowledge level e in the model.

Figures 3a and 3b show that both new and existing users become more likely to ask novel questions after the GPT-3.5 release. The magnitude, however, is substantially larger among new users. Four months after the release, the predicted probability of asking a novel question increases by 8.51 percentage points among new users and by 4.91 percentage points among existing users. This difference is consistent with the possibility that technology helps individuals who have difficulty resolving earlier problems reach subsequent ones. Since we do not observe e, this comparison does not establish a separate result about knowledge levels.

The same analysis also helps separate this pattern from changes in the number of new users.



(a) New users. (b) Existing users.

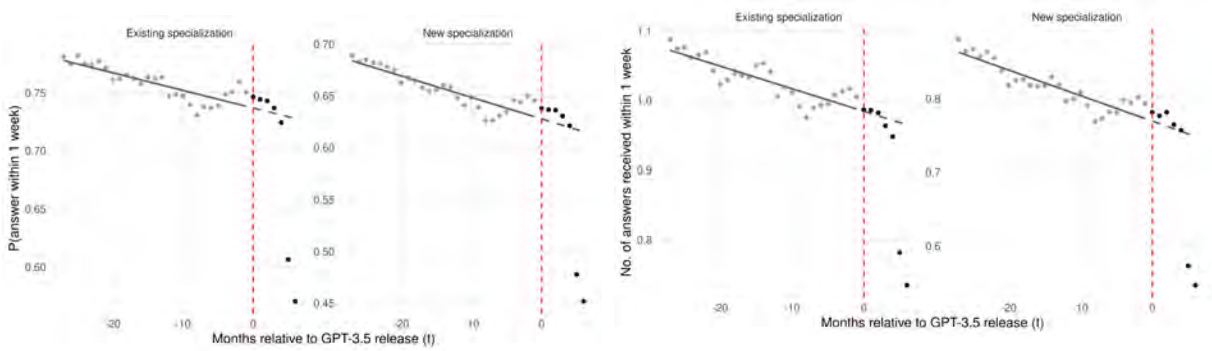
Figure 3: Change in the number of existing-specialization and novel questions on StackOverflow by user type. The vertical line in November 2022 represents the LLM shock.

	New users	New Specialization
	(1)	(2)
	Logit	Logit
GPT-3.5	0.112***	0.064***
	(0.030)	(0.009)
GPT-3.5 $\times t$	−0.003	0.035***
	(0.012)	(0.005)
GPT-3.5 × New User		0.114***
		(0.013)
GPT-3.5 $\times t\times$ New User		0.009
		(0.006)
Change in $\mathbb{P}(g_{s(q),t} = \text{new} \mid u_{s(q),t} = \text{new}, t = 4)$		+8.51***
Change in $\mathbb{P}(g_{s(q),t} = \text{new} \mid u_{s(q),t} = \text{existing}, t = 4)$		+4.91***
Observations	1,654,431	4,039,806

Table 3: Heterogeneous Impact of User Type on Questions.

This table reports estimation results from logit models comparing the log odds ratio of a novel question being asked. The "New users" specification (1) estimates the log odds of a new user asking a novel question with existing users as the baseline category, run on the subset of new specialization questions. The "New Specialization" specification (2) uses questions carrying existing specialization as the baseline category and includes interactions with the new-user indicator. The last two rows show the changes in predicted probabilities (percentage points, pp) for each outcome when GPT-3.5 is introduced, evaluated at $t = 4$. All specifications include an intercept and a linear time trend t. Standard errors clustered at the year-month level. Marginal probability changes assessed via the delta method. ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, †: $p < 0.1$.

Restricting attention to novel questions, the positive coefficient on GPT-3.5 indicates an increase in the relative presence of new users. At the same time, existing users also become more likely to ask novel rather than existing-specialization questions after the release. Thus, the change in the types of questions asked is not explained only by new users joining the community.



(a) Probability of receiving an answer.

(b) Average number of answers received within one week.

Figure 4: Answer outcomes for questions with existing and new specializations around the GPT-3.5 release. The vertical line in November 2022 represents the LLM shock.

4.5 Do Novel Questions Continue to Produce Human Answers?

In the model, questions successfully answered in the community increase the data available for future technology. Our seven-month post-treatment window does not allow us to test whether these answers improve subsequent AI systems. We therefore examine whether questions involving new specializations continue to receive human answers after the GPT-3.5 release.

We examine whether questions receive an answer or accepted answer, the number of answers received, and how quickly the first and accepted answers arrive. Across these measures, we do not find evidence that responsiveness to novel-specialization questions deteriorates after the release. Novel questions remain answerable by the community. In the timing specifications, the introduction of GPT-3.5 is associated with a reduction of about 1.3 days in time to first answer and 0.75 days in time to accepted answer for questions overall. The interaction with new specializations is negative and sizable for time to first answer, indicating that, conditional on being asked, novel questions are answered at least as quickly as before and in some specifications faster.

These results show that questions involving new specializations continue to receive human answers. They do not establish whether these answers become part of the data available to later AI systems or improve technology's ability to resolve the same problems. Full answer-outcome specifications are reported in Appendix 5.

$\times 10^{-3}$	Degree Centrality (1)	Closeness Centrality (2)	Assortativity Degree (3)
GPT-3.5	0.800***	0.100	2.000
	(0.200)	(2.200)	(1.500)
GPT-3.5 $\times$ t	-0.700***	-1.400**	0.800***
	(0.0667)	(0.600)	(0.300)
GPT-3.5 $\times$ log(rank)	-0.095***	-0.013	-0.00681
	(0.0224)	(0.200)	(0.0303)
GPT-3.5 $\times$ t $\times$ log(rank)	0.0836***	0.100	-0.00238
	(0.00849)	(0.0682)	(0.00686)
GPT-3.5	-1.85***	-5.43**	5.35***
GPT-3.5 $\times$ log(rank)	0.24***	0.40†	-0.02
Observations	649,983	649,983	649,983

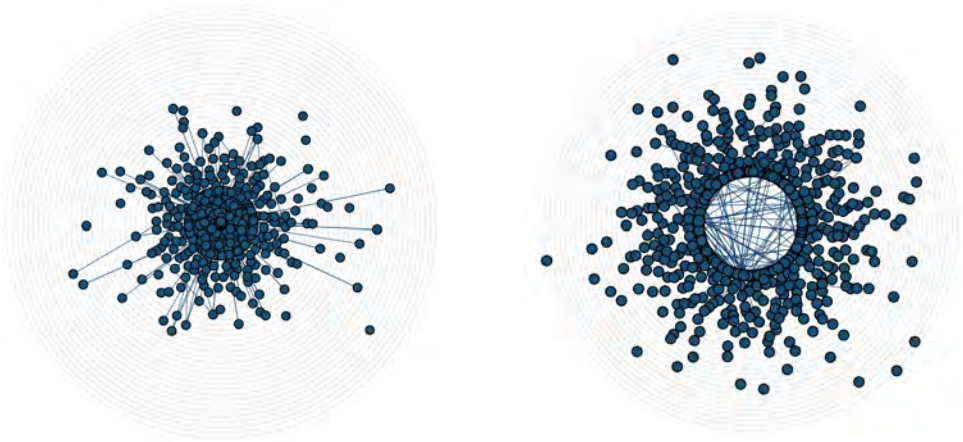
Table 4: Heterogeneous Impact of LLM on Centrality by Domain-level Question Traffic
This table presents estimation results from centrality analysis with logged static rank as the moderator. Standard errors clustered at domain and year-month level. The overall GPT-3.5 effect on baseline and the moderator are shown in the last two rows. These effects are estimated at $t = 4$. ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, †: $p < 0.1$.

5 Aggregate Reorganization of the Knowledge Structure

The preceding analyses show that human knowledge demand shifts away from established specializations and toward new combinations, particularly combinations involving niche knowledge domains. We next ask whether these question-level changes are large enough to reorganize the aggregate knowledge structure of the community.

We represent each month's knowledge structure as a domain co-occurrence network where nodes are knowledge domains and an undirected edge connects two domains when they co-appear in a question posted in that month. The edge weight is the number of questions in which the two domains co-occur. We focus on degree centrality, closeness centrality, and degree assortativity. Degree centrality captures the number of other domains with which a domain co-occurs; closeness centrality captures a domain's proximity to the rest of the network; and assortativity captures whether domains tend to connect to other domains with similar connectivity. Additional construction details and robustness checks are in Appendix 6.

The estimates show that the most popular, highly connected domains experience the largest losses in degree and closeness centrality after GPT-3.5. By the fourth month after the release,



(a) Knowledge structure before November 2022.

(b) Knowledge structure after November 2022.

Figure 5: Network structure representing co-occurrence patterns of technical domains in StackOverflow before and after LLMs. We sample $n = 600$ nodes from the original network using the metric-ranked core sampling algorithm. The circles represent knowledge domains and the edges represent important connections between them.

the lowest-rank domain, i.e., the domain with the largest pre-shock question traffic, experiences a significant decline in degree centrality. The decline is substantially attenuated for higher-rank domains, so niche and new domains gain relative connectivity compared with popular domains. Closeness centrality shows a similar pattern: well-connected popular domains lose the most proximity to the rest of the network, whereas niche and new domains lose closeness at a slower rate and by a smaller magnitude.

Degree assortativity rises after the shock across the distribution of domains. In other words, the network becomes more like-with-like: high-degree domains increasingly connect with other high-degree domains, while niche domains connect more often with other niche domains rather than routing through popular domains. Taken together, these facts point to a reorganization of the community's knowledge network after GPT-3.5. Popular domains lose connections and proximity with other domains, whereas niche domains expand locally.

Figure 5 provides a visual representation of this change. Knowledge domains are placed on concentric rings according to k-coreness, with more embedded domains closer to the center and the same ring radii used before and after the release. The post-GPT network has fewer highly connected domains in the dense inner core and more mass in intermediate and outer shells. The visualization is descriptive; the main claims rest on the centrality and assortativity estimates. Alternative graph-sampling procedures, other centrality measures, and placebo estimates are reported in Appendix 6.

These structural changes are the aggregate counterpart of the question-level results. The knowledge network does not collapse; instead, the knowledge represented in public questions is reallocated away from a centralized, hub-and-spoke structure toward a more decentralized and fragmented structure in which niche combinations play a larger role.

6 Implications for the Persistence of Open Knowledge Production

The model and empirical results together raise a question about whether online communities will continue generating data through human answers. In the model, individuals ask the community for assistance only if $p_h v \geq c$. If asking becomes too costly, unresolved problems are not posted and the community generates no new data from answers to those problems.

This implication matters in the empirical setting because the questions that continue to reach the community are increasingly selected toward new specializations and new combinations of existing knowledge domains. These questions continue to receive human answers, so the community continues to generate data about problems for which individuals seek assistance. At the same time, the overall decline in question volume means that the public knowledge-production system is becoming smaller. It is therefore unclear whether a community can sustain the participation needed to expose and resolve frontier problems when a large share of established questions is redirected to low-cost external tools.

The model does not imply that a platform should preserve routine questions simply to maintain traffic, nor do our data identify an optimal platform policy. The narrower implication is that platform design affects a dynamic input into future problem solving. When unresolved problems are costly to post, they may remain with individuals without becoming public questions. The community then produces fewer answers that could provide data for future search, retrieval, or AI tools.

This creates a managerial tension for knowledge platforms that increasingly integrate AI into the user experience. Making existing knowledge easier to access can improve immediate problem solving and reduce repetitive questions, while the continued value of the human community increasingly comes from questions for which less data are available. The relevant design problem is therefore not simply how to restore aggregate question volume, but how to keep the remaining difficult-to-resolve

questions observable and worth answering.

6.1 Discussion and Conclusion

Our results track changes in technical knowledge demand after the public release of GPT-3.5. Overall question volume falls, but the decline is concentrated in questions that use established specializations. Questions with new specializations are resilient and in our main estimates increase in absolute number. The composition of those novel questions also changes: novelty increasingly comes from new combinations of existing, often niche, knowledge domains rather than only from entirely new technical domains. The increase in novel questioning is larger among new users, although existing users also shift the composition of what they ask. Novel questions continue to receive human answers, and the aggregate domain co-occurrence network reorganizes as popular domains lose connections and proximity while niche domains gain relative importance.

The model provides an interpretation of these empirical patterns. Better technology reduces the need to ask about problems individuals can resolve themselves, while helping them resolve earlier problems and reach subsequent ones sooner. Proposition 1 shows how questions about a subsequent problem can increase even when questions about an earlier problem decline. Proposition 2 shows that the expected total number of questions needed to complete a project can decrease. The decline in established questions and increase in novel questions are consistent with these effects. The larger change among new users and the shift toward new combinations of existing domains are also consistent with the progression mechanism.

The analysis has several limitations. The empirical design is quasi-experimental and does not observe individual LLM usage, so the estimates identify a discontinuity coincident with the GPT-3.5 release rather than a user-level treatment effect of LLM adoption. We use community-assigned tags as a proxy for the knowledge domains implicated by a question. While tags are moderated, interpretable, and stable across the shock, they do not capture all dimensions of technical difficulty or context. Finally, our main post-treatment window ends in May 2023. This is useful for isolating the initial GPT-3.5 break but is too short to observe whether human answers become part of the data available to later AI systems and improve their ability to resolve the same problems.

Even with these caveats, the empirical pattern is consistent across the question-level and network analyses: LLMs change the mix of problems that reach humans. The main distinction is not

simply between more and less difficult questions. As established problems are increasingly handled outside the community, human knowledge demand shifts toward problems that combine technical domains in previously unseen ways. Mapping this shift through question tags, specializations, user histories, answer outcomes, and the co-occurrence network provides a tractable way to study how the boundary between machine assistance and human knowledge production evolves when problem demand itself is endogenous to the technology available for solving earlier problems.

References

Daron Acemoglu and David Autor. Skills, tasks and technologies: Implications for employment and earnings. In *Handbook of Labor Economics*, volume 4B, pages 1043–1171. Elsevier, 2011. doi: 10.1016/S0169-7218(11)02410-5.

Daron Acemoglu and Pascual Restrepo. Automation and new tasks: How technology displaces and reinstates labor. *Journal of Economic Perspectives*, 33(2):3–30, 2019. doi: 10.1257/jep.33.2.3.

Daron Acemoglu, Dingwen Kong, and Asuman Ozdaglar. Ai, human cognition and knowledge collapse. Technical report, National Bureau of Economic Research, 2026a.

Daron Acemoglu, Tianyi Lin, Asuman Ozdaglar, and James Siderius. How ai aggregation affects knowledge. Technical report, National Bureau of Economic Research, 2026b.

Ajay Agrawal, Joshua S Gans, Avi Goldfarb, and Luis Garicano. 1. genius on demand: The value of transformative artificial intelligence. In *The economics of transformative AI*, pages 15–36. University of Chicago Press, 2026.

Nesreen K Ahmed, Jennifer Neville, and Ramana Kompella. Network sampling: From static to streaming graphs. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(2):1–56, 2013.

David H. Autor. Why are there still so many jobs? the history and future of workplace automation. *Journal of Economic Perspectives*, 29(3):3–30, 2015. doi: 10.1257/jep.29.3.3.

David H. Autor, Frank Levy, and Richard J. Murnane. The skill content of recent technological change: An empirical exploration. *The Quarterly Journal of Economics*, 118(4):1279–1333, 2003. doi: 10.1093/qje/118.4.1279.

Kevin J. Boudreau, Eva C. Guinan, Karim R. Lakhani, and Christoph Riedl. Looking across and looking beyond the knowledge frontier: Intellectual distance, novelty, and resource allocation in science. *Management Science*, 62(10):2765–2783, 2016. doi: 10.1287/mnsc.2015.2285.

Léonard Boussioux, Jacqueline N. Lane, Miaomiao Zhang, Vladimir Jacimovic, and Karim R. Lakhani. The crowdless future? generative AI and creative problem-solving. *Organization Science*, 35(5):1589–1607, 2024. doi: 10.1287/orsc.2023.18430.

Erik Brynjolfsson, Bharat Chandar, and Ruyu Chen. Canaries in the coal mine? six facts about the recent employment effects of artificial intelligence. *Working Paper*, 2025a.

Erik Brynjolfsson, Danielle Li, and Lindsey Raymond. Generative AI at work. *The Quarterly Journal of Economics*, 140(2):889–942, 2025b. doi: 10.1093/qje/qjae044.

Gordon Burtch, Dokyun Lee, and Zhichen Chen. The consequences of generative AI for online knowledge communities. *Scientific Reports*, 14:10413, 2024. doi: 10.1038/s41598-024-61221-0.

Christoph Carnehl and Johannes Schneider. A quest for knowledge. *Econometrica*, 93(2):623–659, 2025. doi: 10.3982/ECTA22144.

R. Maria del Rio-Chanona, Nadzeya Laurentsyeva, and Johannes Wachs. Large language models

reduce public knowledge sharing on online Q&A platforms. *PNAS Nexus*, 3(9):pgae400, 2024. doi: 10.1093/pnasnexus/pgae400.

Fabrizio Dell'Acqua, Edward McFowland III, Ethan R Mollick, Hila Lifshitz-Assaf, Katherine Kellogg, Saran Rajendran, Lisa Krayer, François Candelon, and Karim R Lakhani. 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*, (24-013), 2023.

Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. Gpts are gpts: An early look at the labor market impact potential of large language models. *arXiv preprint arXiv:2303.10130*, 10, 2023.

Samer Faraj, Georg von Krogh, Eric Monteiro, and Karim R. Lakhani. Online community as space for knowledge flows. *Information Systems Research*, 27(4):668–684, 2016. doi: 10.1287/isre.2016.0682.

Lee Fleming. Recombinant uncertainty in technological search. *Management Science*, 47(1):117–132, 2001. doi: 10.1287/mnsc.47.1.117.10671.

Santo Fortunato, Carl T Bergstrom, Katy Börner, James A Evans, Dirk Helbing, Staša Milojević, Alexander M Petersen, Filippo Radicchi, Roberta Sinatra, Brian Uzzi, et al. Science of science. *Science*, 359(6379):eaao0185, 2018.

Luis Garicano. Hierarchies and the organization of knowledge in production. *Journal of Political Economy*, 108(5):874–904, 2000. doi: 10.1086/317671.

Luis Garicano and Esteban Rossi-Hansberg. Organizing growth. *Journal of Economic Theory*, 147 (2):623–656, 2012. doi: 10.1016/j.jet.2009.11.007.

Enrique Ide and Eduard Talamàs. Artificial intelligence in the knowledge economy. *Journal of Political Economy*, 133(12):3762–3800, 2025. doi: 10.1086/737233.

Lars Bo Jeppesen and Karim R. Lakhani. Marginality and problem-solving effectiveness in broadcast search. *Organization Science*, 21(5):1016–1033, 2010. doi: 10.1287/orsc.1090.0491.

Benjamin F. Jones. The burden of knowledge and the "death of the renaissance man": Is innovation getting harder? *Review of Economic Studies*, 76(1):283–317, 2009. doi: 10.1111/j.1467-937X.2008.00531.x.

Samia Kabir, David N. Udo-Imeh, Bonan Kou, and Tianyi Zhang. Is stack overflow obsolete? an empirical study of the characteristics of chatgpt answers to stack overflow questions. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*, pages 1–17, 2024. doi: 10.1145/3613904.3642596.

David McDowall, Richard McCleary, and Bradley J. Bartos. *Interrupted Time Series Analysis*. Oxford University Press, 2019.

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

Bruno Ribeiro and Don Towsley. Estimating and sampling graphs with multidimensional random walks. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 390–403, 2010.

Hani Safadi, Steven L. Johnson, and Samer Faraj. Who contributes knowledge? core–periphery tension in online innovation communities. *Organization Science*, 32(3):752–775, 2021. doi: 10.1287/orsc.2020.1364.

J Schneider, T Shah, and JC Abraham. Navigating the generative ai disruption in software. *McKinsey & Company*, 2024.

Neha Sharma, Gad Allon, and Achal Bassamboo. Structuring online communities. SSRN Working Paper 3959181, revised August 2025, 2021.

Brian Uzzi, Satyam Mukherjee, Michael Stringer, and Ben Jones. Atypical combinations and

scientific impact. *Science*, 342(6157):468–472, 2013. doi: 10.1126/science.1240474.

Caroline S Wagner, J David Roessner, Kamau Bobb, Julie Thompson Klein, Kevin W Boyack, Joann Keyton, Ismael Rafols, and Katy Börner. Approaches to understanding and measuring interdisciplinary scientific research (idr): A review of the literature. *Journal of informetrics*, 5 (1):14–26, 2011.

Jing Wang, Gen Li, and Kai-Lung Hui. Monetary incentives and knowledge spillover: Evidence from a natural experiment. *Management Science*, 68(5):3549–3572, 2022. doi: 10.1287/mnsc.2021.4048.

Molly McLure Wasko and Samer Faraj. Why should i share? examining social capital and knowledge contribution in electronic networks of practice. *MIS Quarterly*, 29(1):35–57, 2005.

Martin L. Weitzman. Recombinant growth. *The Quarterly Journal of Economics*, 113(2):331–360, 1998. doi: 10.1162/003355398555595.

Lei Xu, Tingting Nian, and Luís Cabral. What makes geeks tick? a study of stack overflow careers. *Management Science*, 66(2):587–604, 2020. doi: 10.1287/mnsc.2018.3264.

Xiaoquan Michael Zhang and Feng Zhu. Group size and incentives to contribute: A natural experiment at chinese wikipedia. *American Economic Review*, 101(4):1601–1615, 2011. doi: 10.1257/aer.101.4.1601.

Kai Zhu and Dylan Walker. The AI democratization paradox: Evidence from decentralized knowledge communities. *Management Science*, 2026. doi: 10.1287/mnsc.2024.04717. Articles in Advance.

Appendix A

1 Proofs for Section 2

Proof of Proposition 1. Consider individuals with knowledge e and hold the data available for each problem type fixed. Write $p_s = p_{own}(D_s, e, \phi)$ and $p_r = p_{own}(D_r, e, \phi)$.

Since $p_h v \geq c$, the probability of resolving s in a period is

$$\sigma_s = p_s + (1 - p_s)p_h.$$

All individuals initially encounter s. Thus, the contribution of individuals with knowledge e to questions about s in period 0 is

$$Q_{s,0}(e) = Mf(e)(1 - p_s),$$

which is weakly decreasing in ϕ, and strictly decreasing whenever $\partial_\phi p_s > 0$.

Individuals encounter r in period 1 only if they resolved s in period 0 and transitioned to r. Their question demand is therefore

$$Q_{r,1}(e) = Mf(e)\pi_{sr}\sigma_s(1 - p_r).$$

For $f(e) > 0$, $\pi_{sr} > 0$, and $p_r < 1$, logarithmic differentiation gives

$$\partial_\phi \log Q_{r,1}(e) = \frac{(1 - p_h)\partial_\phi p_s}{p_h + (1 - p_h)p_s} - \frac{\partial_\phi p_r}{1 - p_r}.$$

Hence $\partial_\phi Q_{r,1}(e) > 0$ if and only if condition (5) holds. ■

Proof of Proposition 2. Fix an individual with knowledge e and consider technology $i \in \{0, 1\}$. Let

$$p_r^i(e) = p_{own}(D_r, e, \phi^i), \qquad \sigma_r^i(e) = p_r^i(e) + [1 - p_r^i(e)]p_h.$$

An individual who does not resolve a problem remains at that problem and tries again in the next period. Each attempt at type r leads to successful resolution with probability $\sigma_r^i(e)$ and to a question with probability $1 - p_r^i(e)$. The expected number of questions asked before resolving one encounter with r is therefore

$$\sum_{k=0}^{\infty}[1 - \sigma_r^i(e)]^k[1 - p_r^i(e)] = \frac{1 - p_r^i(e)}{p_r^i(e) + [1 - p_r^i(e)]p_h}.$$

Upon resolving r, the next problem type is determined by π_{rs}, which does not depend on the technology. Thus, an improvement in private resolution changes how long individuals spend at each problem, but not the sequence of problem types they encounter upon successful resolution.

Let N_r be the expected number of times a project encounters type r before completion. Since projects eventually reach completion and the set of problem types is finite, N_r is finite and inde-

pendent of the technology.

The expected total number of questions asked by an individual with knowledge e before completing the project is

$$C^i(e) = \sum_{r \in \mathcal{R}} N_r \frac{1 - p_r^i(e)}{p_r^i(e) + [1 - p_r^i(e)]p_h}.$$

For $p \in [0, 1]$, define

$$g(p) = \frac{1 - p}{p + (1 - p)p_h}.$$

Then

$$g'(p) = -\frac{1}{[p + (1 - p)p_h]^2} < 0.$$

By assumption, $p_r^1(e) \geq p_r^0(e)$ for every r and e. Therefore $C^1(e) \leq C^0(e)$ for every e.

Integrating against $Mf(e)$ establishes that the expected total number of questions asked before project completion weakly decreases. The inequality is strict if private resolution improves for a positive mass of individuals at a problem type they encounter with positive probability.

With a constant inflow of new individuals and unchanged data and technology, the same comparison holds for long-run question volume per period. ■

Data generated through human answers. Holding the distribution of individuals across problem types fixed at the beginning of period t, an increase in $D_{s,t}$ weakly increases the probability of resolving problem s without human assistance. It therefore weakly reduces questions about s in period t and increases the number of individuals who reach its direct successors in period $t + 1$. This does not imply that questions about every subsequent problem increase at all later dates.

First appearance of a specialization. Suppose the realized number of questions of type r in each period is Poisson, with mean $Q_{r,t}$, independently across periods conditional on the model's question demand. The probability that type r appears in the community for the first time in period t is

$$\exp\left(-\sum_{\tau<t} Q_{r,\tau}\right)\left(1 - e^{-Q_{r,t}}\right).$$

The expected number of questions in period t carrying a specialization that has not appeared in an earlier period is

$$Q_{r,t} \exp\left(-\sum_{\tau<t} Q_{r,\tau}\right).$$

Holding earlier periods fixed, both expressions increase with $Q_{r,t}$. Summing over problem types gives the expected number of newly observed specializations and the expected number of questions involving those specializations, respectively.

2 Data and Preprocessing

Data Source and Collection

We draw our sample from the official Stack Exchange public data dumps[2], which provides daily logs of user posts on all public Stack Exchange sites. Parsing all `Posts.xml` files dated between 1 January 2018 and 31 December 2023, we retain only rows with the user questions, i.e., `PostTypeId = 1`. Other posts, such as answers, comments, edits, and community-wiki contributions, are discarded to focus on primary knowledge-seeking interactions. For each question, we keep its unique identifier, UTC creation time, and tags associated with the question (the maximum number of tags associated with a question is 5), full question text, community score for the question, and its answer count.

We remove the version number that appears at the end of the tags. For instance `python-3.x, python-4.0, python` are normalized to `python`. Lastly, for each specialization, we order the associated tags alphabetically. The resulting panel comprises roughly 9.33 million questions and is described in more detail in Section 2.

StackOverflow serves as a natural test bed for many reasons. First, StackOverflow is the main source of knowledge for programming enthusiasts and developers. Second, these questions provide rich and quality data to train the GenAI models. Third, after the launch of GPT 3.5 in November 2022[3] StackOverflow saw a substantial decline in its question volumes, unlike general interest discussion forums like Reddit and Quora Burtch et al. [2024], del Rio-Chanona et al. [2024].

Empirical Setting: GPT 3.5 Release

We focus on a multi-year period from 2018 to 2023 to analyze longitudinal shifts in questions asked on StackOverflow. This extensive timeframe was chosen to ensure a robust pre-intervention ("control") period spanning 2018–2021, capturing baseline platform dynamics prior to the introduction of large-scale generative AI tools. The years 2022–2023 serve as the "treatment" period, marked by the widespread public deployment of prominent Large Language Models (LLMs).

Key milestones in this period include the release of GitHub Copilot in May 2022 and the public launch of OpenAI's ChatGPT 3.5 in November 2022. While GitHub Copilot represented an early step toward integrating generative models into software development workflows, the introduction of ChatGPT 3.5 marked a qualitatively different shift. Unlike earlier LLMs, which were often limited to specialized users and narrow applications, GPT-3.5 marked the first large-scale, general-purpose LLM accessible to the public via a conversational interface.

The mass adoption of ChatGPT 3.5 made it a natural intervention point for studying the impact of LLMs on human knowledge production systems like StackOverflow. Hence, we use the November 2022 release of ChatGPT 3.5 as the focal point of our empirical setting, defining the surrounding year-months as pre- and post-intervention periods for comparative analysis.

[2] `https://data.stackexchange.com/`

[3] See the annual developer survey conducted in 2024, `https://survey.stackoverflow.co/2024/ai`

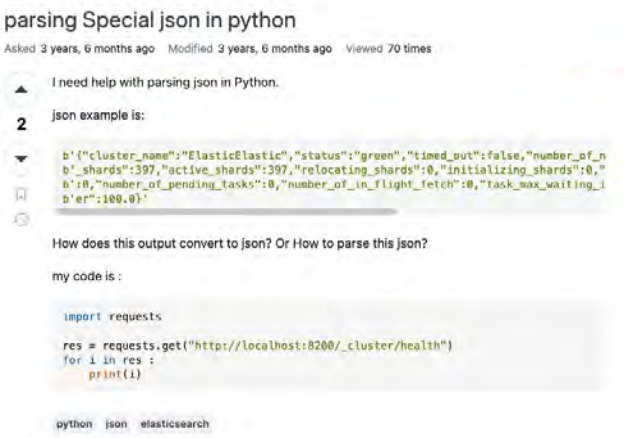


Figure 6: Sample Question on StackOverflow

Knowledge Domains and Specialization In the StackOverflow community, users who ask questions manually assign *tags* that best describe the *knowledge domains* a user must possess to answer their questions. These tags serve as a human-curated, community-moderated taxonomy that plays a central role in searchability, moderation, and knowledge organization. For example, consider the StackOverflow post illustrated in Figure 6. The question seeks help with parsing JSON in Python and contains the tags `python`, `json`, and `elasticsearch`. Thus, answering the question requires fluency in these three knowledge domains. The combination of tags used to label a question reflects a *specialization*—a specific intersection of knowledge domains required to solve that problem. For instance, a combination of {`python, pandas`} suggests a specialization in data wrangling in Python, whereas {`react, graphql, tailwind`} would likely require a specialization in modern front-end architecture. Hence, we refer to this set of domains as a *specialization*, i.e., a unique combination of tags. Throughout this study, we treat individual tags as atomic *knowledge domains*, and the combination of tags attached to a question as its *specialization*.

We leverage these tags and their combinations on questions to structure the knowledge in the community. While there are other methods, such as latent topic models, which infer themes from textual co-occurrence patterns from the full question *text*, we use question tags for the following key reasons. First, tags are moderated, semantically meaningful, and stable over time—making them especially valuable for measuring the impact of external shocks, such as the introduction of large language models (LLMs). Second, the tagging system itself is unaffected by these shocks: users continue to assign tags through the same interface and under the same community moderation mechanisms as before[4][5]. Last, these tags are more amenable to interpretation as they represent the current technical knowledge domains used by the users, and their combinations signal the specializations and the applications of these knowledge domains.

[4]The latent topic distributions change may not be independent of the LLM shock.

[5]Auditing and editing tags to questions are exogenous to the GPT release under the assumption that the moderators' behavior is not affected by the GPT release.

Summary Statistics

We provide a high-level overview of the knowledge domains and specializations in our dataset. This summary highlights the general evolution of the community's knowledge structure before we apply the formal analysis presented in the main text.

We summarize how the knowledge domains and the specializations users require have evolved over time. Tables 5–10. Table 5 illustrates the most frequently used and the least frequently used tags in the community. As expected, tags such as `python`, `javascript, java, c#` are among the most popular tags across all years from 2018-2024 as they are the most common programming languages that support various applications. In contrast, the least popular tags show significant year-to-year turnover, suggesting that these are niche domains that do not persist in the community's knowledge landscape.

On the other hand, Table 6 demonstrates the percentile of the frequency of tags occurring in the community. Notably, the contribution of tags to the question volumes in the community is intensely skewed, wherein just 10% of the tags appear only once, 25% of the tags appear at most 2 times in the whole year, and 1% of tags appear more than 1000 times. Finally, Table 7 shows that across years, there is a similar proportion of tags that are discarded and retained over the years, suggesting that the underlying base of knowledge domains is not rapidly expanding or contracting.

Year	**Most Popular Knowledge Domains**	**Least Popular Knowledge Domains**
2018	`javascript, python, java, c#, android, php`	`shinythemes, interbase, alertifyjs, rx-binding, haskell-turtle`
2019	`python, javascript, java, c#, android, html`	`class-structure, normals, automic, graphql.net, draw`
2020	`python, javascript, java, c#, html`	`guice-persist, msr, yampa, sidetiq, onclick`
2021	`python, javascript, java, reactjs, html`	`micromamba, choclat.js, pythonplotter, tinymapper, phplot`
2022	`python, java, javascript, reactjs, c#`	`droidscript, sqlite-odbc, mongodb-tools, grako, bigsql`
2023	`python, java, javascript, reactjs, c#`	`jsonml, dreamfactory, language-evolution, tortoisegitmerge, gmail.js`

Table 5: Most popular and least popular knowledge domains on StackOverflow over the years

In a similar vein, Table 8 lists the most popular and the least popular specializations observed in the community. While one would expect `python, r, java` to be among the most popular specializations, it is interesting to observe that `dart|flutter` is one of the most popular specializations in 2023. This specialization relates to questions on Google's app development platform, and interestingly, `dart` or `flutter` were not popular tags in any year. As in Tables 6 and 7,

Year	% of Knowledge domains					
	10%	25%	50%	75%	90%	99%
2018	1	2	7	27	108	1691
2019	1	2	6	24	100	1538
2020	1	2	7	25	104	1654
2021	1	2	6	22	89	1398
2022	1	2	6	20	80	1244
2023	1	2	5	18	66	945

Table 6: Question count with % of knowledge domains

Year	# of Knowledge Domains			
	Unique	Retained	New	Discarded
2018	39656			
2019	39456	34055	5401	5601
2020	41179	34616	6563	4840
2021	40017	34996	5021	6183
2022	39777	34193	5584	5824
2023	40344	34092	6252	5685

Table 7: Evolution of knowledge domains

Year	Most Popular Specializations	Least Popular Specializations
2018	javascript, python, excel|vba, css|html, java	java|key-bindings|keyboard|macos|swing, dispatch-async|gran-central-dispatch|ios|swift, .net|msbuild|nuget|visual-studio, akka|future|playframework|scala, module|ocaml|signature
2019	python, javascript, excel|vba, pandas|python, r	laravel|laravel-envoy|laravel-horizon|php|server, android|android-bottomappbar|material-design, css|database|ms-access, html|iframe|java|javascript|selenium, http|json|mysql|node.js
2020	python, javascript, pandas|python, r, css|html	google-authentication|php|service-accounts, javascript|react-component|react-hooks|react-props|reactjs, keras|nueral-network|oop|python|tensorflow, api|json|normalize|python|python-requests, handlebars.js|html|javascript|tags
2021	python, javascript, pandas|python, r, css|html	.net|c#|coordinates, backend|dart|flutter|frameworks|server, angular|jestjs|ts-jest|tsconfig, face-recognition|pip|python, c#|func|generics|unity-game-engine|unmanaged
2022	python, javascript, pandas|python, r, css|html	arduino|i2c|lego-mindstorms-e, amazon-emr|amazon-s|apache-spark|pyspark, database|express|mongodb|node.js|realm, apache-kafka|apache-kafka-connect|debezium|mysql|postgresql, audio|cross-correlation|fft|java|signal-processing
2023	python, javascript, r, css|html, dart|flutter	autofac|c#|wcf, elasticsearch|pip|python, java|ssl, django|payment|stripe-payments, azure|azure-active-directory|azure-sdk-.net|c#|virtual-machine

Table 8: Most popular and least popular specializations over the years

Tables 9 and 10 outline the percentile of the specializations in the community and the evolution

Year	% of Specializations					
	10%	25%	50%	75%	90%	99%
2018	1	1	1	1	2	15
2019	1	1	1	1	2	15
2020	1	1	1	1	2	14
2021	1	1	1	1	2	14
2022	1	1	1	1	2	13
2023	1	1	1	1	2	11

Table 9: Question count with % of specializations

Year	# of Specializations			
	Unique	**Retained**	**New**	**Discarded**
2018	841694			
2019	769157	144104	625053	697590
2020	856449	142880	713569	626277
2021	724067	135329	588738	721120
2022	657921	116825	541096	607242
2023	545951	92711	453240	565210

Table 10: Evolution of specializations

of specializations over time, respectively. Table 9 illustrates the skewed distribution of questions across specializations, wherein 90% of the specializations contribute merely two questions or fewer in a year, and most of the question volumes are generated by the top 1% of the specializations. Further, we observe that the number of specializations dropped and added year over year in the Table 10.

3 Additional Analyses for Question Demand

3.1 Changes in Question Type Concentrations: Lorenz Curves

We show that the disproportionate change in question traffic is evident not only between new and existing specializations, but also in the concentration of activity across specializations. Each curve in Figure 7 is a Lorenz curve built on data from one month. For each month, specializations are ordered according to the number of questions they carry. The x-axis represents the cumulative share of specializations, from least to most frequent, and the y-axis shows the cumulative share of total questions accounted for by those specializations.

The dashed line marks the 45-degree line; the further a Lorenz curve lies from this line, the more concentrated platform activity is in the more popular specializations. The comparison shows that, while the overall distribution of questions across specializations remains skewed, the degree of concentration decreases post-intervention: a smaller share of questions is concentrated on the more popular specializations, and more activity shifts toward the long tail of less frequently mentioned specializations.

3.2 Falsification Test

Falsification Tests. We conduct falsification tests on the main results reported in Table 1 to ensure that our main effect is not purely driven by random data variations. Specifically, we focus on the pre-treatment period, replicate the length of the actual post-treatment window, and des-

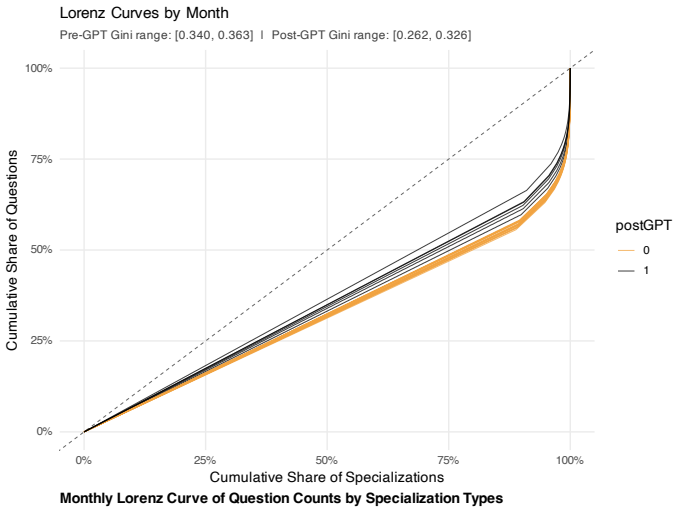


Figure 7: Monthly Lorenz Curves of Question Counts by Specialization

ignate six months prior to the LLM shock (November 2022) as a pseudo-treatment date. In this specification, May 2022 through October 2022 serve as the pseudo post-treatment period, while months prior to May 2022 serve as the pseudo pre-treatment period. The corresponding estimates are reported in Table 11. None of the coefficients are statistically significant, indicating that the pseudo-LLM shock does not affect the distribution of question types (new versus existing). This falsification test supports that our main results are not driven by spurious data variations.

3.3 Alternative Time Trends and Model Comparison

Alternative Polynomial Time Trend $f(\tau_t)$ and Model Comparisons. We conduct a robustness check on our main results by varying the functional form of the polynomial time trend $f(\tau_t)$. Table 12 reports estimates from models with linear (columns 1 and 3) and quadratic (columns 2 and 4) time trends. The last two rows of the table summarize the GPT-3.5 treatment effect on the baseline and the heterogeneous effect across specialization types. Given that only seven post-treatment periods are available, we do not consider higher-order polynomials.

The results show that the estimated effects are consistent across specifications: the treatment effects in the last two rows are comparable under both linear and quadratic trends. Comparing model fit, the BIC values indicate that the linear trend provides a better fit for both outcome variables. Accordingly, we report results based on the linear time trend $f(\tau_t)$ in the main text. Note that columns 1 and 3 in Table 12 reproduce the main results (Table 1) and are included here solely for model comparison.

4 Additional Analyses for Novel Question Composition

Falsification Tests. Analogous to the placebo test in Table 11, we conduct falsification tests for the main results in Table 2 and report the estimates in Table 13.

	Total Questions (1)	New Specialization (2)
Pseudo GPT-3.5	1,361.7	0.005
	(2,324.7)	(0.003)
Pseudo GPT-3.5 $\times t$	-223.6	0.0002
	(305.4)	(0.0005)
Pseudo GPT-3.5 $\times$ New Spec.	1,078.0	
	(804.2)	
Pseudo GPT-3.5 $\times$ t $\times$ New Spec.	171.2	
	(166.5)	
Log-Likelihood	-519.2	-2,462,112.1
Observations	56	3,478,944

Table 11: Placebo Test: Question Traffic by Specialization Types

This table presents the placebo test of Table 1. We focus on pre-LLM shock time and treat the 6 months before the LLM shock as the pseudo treatment time and run the same specifications in both columns. ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, †: $p < 0.1$.

The results show no significant coefficients for the relative log-odds model concerning new-domain questions. For niche-containing domains, the coefficient on the pseudo GPT-3.5 indicator is negative and marginally significant. Importantly, our main analysis finds a positive effect of GPT-3.5 on niche-containing domains—opposite in sign to the placebo result reported here. Overall, this evidence supports that our main findings are not driven by random variations in the data.

5 Answer Outcomes for Novel Questions

The main text asks whether the novel questions that remain in the community continue to receive human answers. Here we report the underlying answer-outcome specifications, including the number of answers, the probability of receiving an answer or accepted answer, and the timing of the first and accepted answer.

We assess how quickly the community responds by estimating (7) with dependent variable $\Delta^{\text{ans}}_{s(q),t}$ and $\Delta^{\text{accept ans}}_{s(q),t}$. Table 15 reports OLS models for the time to first answer and the time to accepted answer (in days). For all questions, the introduction of GPT-3.5 is associated with a reduction of about 1.3 days in time to first answer and 0.75 days in time to accepted answer. The interaction terms with new specializations are negative and sizable for time to first answer, indicating that novel specialization questions experience an even larger reduction in waiting time. In other words, conditional on being asked, novel questions are not only as likely to be answered as before; they tend to be answered faster.

	Total Questions		New Specialization	
	(1)	(2)	(3)	(4)
	OLS	OLS	Logit	Logit
GPT-3.5	-666.5	-838.9	0.079***	0.067***
	(1,976.2)	(1,893.1)	(0.010)	(0.014)
GPT-3.5 × t	-2,598.6***	-4,960.0***	0.036***	0.003
	(476.7)	(1,197.6)	(0.006)	(0.012)
GPT-3.5 × New Spec.	4,002.1***	2,987.1**		
	(728.0)	(1,034.5)		
GPT-3.5 × t × New Spec.	2,070.1***	1,517.7*		
	(258.2)	(669.1)		
GPT-3.5 × t^2		400.0†		0.005*
		(224.8)		(0.002)
GPT-3.5 × t^2 × New Spec.		53.2		
		(122.5)		
GPT-3.5 at $t = 4$	-10,394	-14,279	0.223	0.159
GPT-3.5 × New Spec. at $t = 4$	12,283	9,909		
BIC	1292.3	1299.2	5,465,101	5,465,002
Observations	68	68	4,039,806	4,039,806

Table 12: Heterogeneous Impact of LLM on Question Traffic by Specialization Types: Quadratic $f(\tau_t)$

This table presents the robustness check of Table 1. Columns (1) and (3) are our main results as presented in Table 1. Columns (2) and (4) keep all other model parameters the same, and we alter the linear trend into a quadratic trend. The BIC across the linear trend and quadratic trend models is comparable in both specifications. The bottom rows also report GPT effects evaluated at $t = 4$ for both baseline and the moderating effect by specialization types. The effects are comparable across models. ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, †: $p < 0.1$.

Specialization Type (baseline: Popular only)	New-Domain	Niche-Containing
Pseudo-GPT-3.5	-0.002	-0.026^{*}
	(0.059)	(0.011)
Pseudo-GPT-3.5 $\times$ t	-0.017	-0.004
	(0.013)	(0.002)
Change in $\mathbb{P}(d_{s(q)=\cdot,t=4})$	-0.03	-0.90^{*}
Change in $\mathbb{P}(d_{s(q)=\cdot,t=7})$	-0.06	$-1.14^{\dagger}$
Observations	1,412,580	1,412,580

Table 13: Falsification Test: Impact of Pseudo-GPT on Share of Questions by Types of New Specializations

This table presents the placebo test to Table 2. We focus on pre-LLM shock time and treat the 6 months before the LLM shock as the pseudo treatment time and run the same specifications in both columns of Table 2. The estimation results from the multinomial logit model using a pseudo-GPT placebo treatment. All specifications also contain the linear time trend t. Standard errors are clustered at the year-month level. ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, †: $p < 0.1$.

	No. of answers (1) OLS	Prob. of getting an answer (2) Logit
GPT-3.5	$0.0503^{\dagger}$	0.2185^{*}
	(0.029)	(0.098)
GPT-3.5 $\times t$	-0.0375^{***}	-0.1256^{***}
	(0.009)	(0.029)
GPT-3.5 $\times$ New Spec.	0.0119^{**}	-0.0024
	(0.004)	(0.008)
GPT-3.5 $\times$ $t\times$ New Spec.	0.001	-0.0010^{**}
	(0.001)	(0.000)
Change in $\mathbb{P}((a_{q(s),t}=1)\|g_{s(q),t}=\text{new}, t=4)$		-6.47 pp
Observations	3,972,547	3,972,547

Table 14: Number of Answers and Answer Probability

This table reports estimation results from an OLS model for the number of answers (column 1) and a logit model for the probability of getting an answer within a week (column 2). All specifications include an intercept and a linear time trend t. "New Spec." denotes questions in new specializations. ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, †: $p < 0.1$.

6 Knowledge Network: Construction and Robustness

6.1 Construction of the Knowledge Structure

We represent the set of tag co-occurrences as a graph $\mathcal{G} = (\mathcal{T}, E)$, where $\mathcal{T}$ denotes the set of all tags and E the set of undirected edges connecting tags that co-occur in at least one question. The weight of an edge captures the number of questions in which the two tags appear together.

	Time to answer (days)	Time to accepted answer (days)
	(1)	(2)
	OLS	OLS
GPT-3.5	-1.3173***	-0.7533***
	(0.397)	(0.205)
GPT-3.5 $\times t$	0.0263	0.0131
	(0.035)	(0.024)
GPT-3.5 $\times$ New Spec.	-1.2366***	-0.2116
	(0.335)	(0.283)
GPT-3.5 $\times$ $t\times$ New Spec.	-0.1441***	-0.0467
	(0.018)	(0.011)
Observations	3,110,128	1,627,756

Table 15: Effects of GPT-3.5 on Answer Timing

Notes: This table reports OLS estimates of the effect of GPT-3.5 availability on answer timing outcomes. Column (1) uses the time to the first answer (in days) as the dependent variable. Column (2) uses the time to the accepted answer (in days) as the dependent variable. All specifications include an intercept and a linear time trend t. "New Spec." is an indicator for questions in new specializations, and interaction terms capture heterogeneous effects over time and by specialization status. Robust standard errors are reported in parentheses. ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, †: $p < 0.1$.

	Prob. of having an accepted answer	With New Specialization
	(1)	(2)
	Logit	Logit
GPT-3.5	0.0950†	0.0978†
	(0.055)	(0.052)
GPT-3.5 $\times t$	-0.0676***	-0.0592***
	(0.018)	(0.017)
GPT-3.5 $\times$ New Spec.		0.0038
		(0.743)
GPT-3.5 $\times$ $t\times$ New Spec.		-0.0171***
		(0.004)
Change in $\mathbb{P}((a_{s(q),t} = \text{accepted}), t = 4)$	-4.00	
Change in $\mathbb{P}((a_{s(q),t} = \text{accepted}), t = 7)$	-8.24	
Change in $\mathbb{P}((a_{s(q),t} = \text{accepted}) \vert g_{s(q)} = \text{new}, t = 4)$		-4.35
Change in $\mathbb{P}((a_{s(q),t} = \text{accepted}) \vert g_{s(q)} = \text{new}, t = 7)$		-8.68
Observations	3,972,547	3,972,547

Table 16: Probability of answer being accepted

This table reports estimation results from logit models for the probability of getting an accepted answer within a week (column 2). All specifications include an intercept and a linear time trend t. "New Spec." denotes questions in new specializations. ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, †: $p < 0.1$.

Formally, let $A \in \{0,1\}^{|\mathcal{T}|\times|\mathcal{T}|}$ denote the adjacency matrix of $\mathcal{G}$. An element $a_{ij} = 1$ if tags t_i and t_j co-occur in at least one question and $a_{ij} = 0$ otherwise. We construct such co-occurrence

	No. of answers	Time to answer (days)	Time to accepted answer (days)
	(1)	(2)	(3)
	OLS	OLS	OLS
GPT-3.5	0.0503†	-1.3173***	-0.7533***
	(0.029)	(0.397)	(0.205)
GPT-3.5 $\times t$	-0.0375***	0.0263	0.0131
	(0.009)	(0.035)	(0.024)
GPT-3.5 $\times$ New Spec.	0.0119**	-1.2366***	-0.2116
	(0.004)	(0.335)	(0.283)
GPT-3.5 $\times$ $t\times$ New Spec.	0.001	-0.1441***	-0.0467
	(0.001)	(0.018)	(0.011)
Observations	3,972,547	3,110,128	1,627,756

Table 17: Number of Answers, and Time to Answers

This table reports estimation results from an OLS model for the number of answers (column 1) and a logit model for the probability of getting an answer within a week (column 2). All specifications include an intercept and a linear time trend t. "New Spec." denotes questions in new specializations. ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, †: $p < 0.1$.

networks for each month. Each node represents a unique tag, each edge signifies co-occurrence within a question, and the edge weight reflects how often this co-occurrence appears in that month.

For each month and knowledge domain, we compute degree centrality, closeness centrality, degree assortativity, clustering coefficient, and betweenness centrality. To study heterogeneity by domain popularity, we order domains according to total pre-treatment question counts; a higher rank corresponds to a less frequently used domain.

6.2 Overall Impact of LLMs on Domain Centralities

Table 18 reports the estimated impact of LLMs on degree centrality, closeness centrality, and degree assortativity on average across all domains. We find that average domain-level degree and closeness centralities decline, whereas average assortativity increases.

Overall, these estimates are consistent with the main-text result that the centrality of popular tags weakens while the network becomes more fragmented and assortative after the GPT-3.5 release.

6.3 Other Centrality Measures

We present estimators from alternative centrality measures as the outcome variables in Table 19. The results suggest that the treatment effect on the clustering coefficient is insignificant across all ranks when evaluated at $t = 4$. The treatment effect on betweenness centrality becomes insignificant for nodes with ranks greater than approximately 20, indicating that less popular nodes experience limited changes in betweenness centrality. Extending the analysis to a longer timeline, the treatment effects on both the baseline and heterogeneity of betweenness centrality and the clustering coefficient

$\times 10^{-3}$	Degree Centrality (1)	Closeness Centrality (2)	Assortativity Degree (3)
GPT-3.5	0.027***	0.029	1.909
	(0.006)	(0.557)	(1.246)
GPT-3.5 $\times$ t	-0.010***	-0.598***	0.828**
	(0.0015)	(0.143)	(0.239)
GPT-3.5 at $t = 4$	-0.014**	-2.36***	5.22***
Observations	649,983	649,983	649,983

Table 18: Impact of LLM on Centrality.

This table presents estimation results from centrality analysis. All specifications also contain the linear time trend t. Standard errors clustered at domain and year-month level. ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, †: $p < 0.1$.

	Betweenness Centrality (1)	Clustering Coefficient (2)
GPT-3.5	4,925.8	-0.0165
	(59,867.0)	(0.0122)
GPT-3.5 $\times$ t	30,097.8*	0.0074**
	(16,286.9)	(0.0028)
GPT-3.5 $\times$ log(Rank)	-597.5	0.0020
	(7,712.3)	(0.0022)
GPT-3.5 $\times$ t $\times$ log(Rank)	-3,875.9*	-0.0015**
	(2,098.1)	(0.0005)
Observations	649,983	649,983

Table 19: Robustness Check: Other Centrality Measures.

This table presents estimation results from centrality analysis with logged rank as the moderator. Standard errors clustered at the domain and year-month level. ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, †: $p < 0.1$.

eventually become marginal or insignificant.

6.4 Falsification Tests

Analogous to the placebo test in Table 11, we conduct falsification tests for the main results in Table 4 and report the estimates in Table 20. The results show no significant coefficients for degree or closeness centrality. For assortativity, the coefficients on Pseudo GPT-3.5 and Pseudo GPT-3.5 $\times$ t are statistically significant but opposite in sign to our main effects. This contrast reinforces that our main findings are not driven by random variation in the data.

	Degree Centrality (1)	Closeness Centrality (2)	Assortativity Degree (3)
Pseudo GPT-3.5	0.0001	0.0002	-0.006***
	(9.21×10^{-5})	(0.001)	(0.001)
Pseudo GPT-3.5 $\times t$	1.35×10^{-5}	8.13×10^{-5}	-0.0009***
	(1.44×10^{-5})	(6.28×10^{-5})	(0.0002)
Pseudo GPT-3.5 $\times$ log(rank)	-5.21×10^{-8}	-4.6×10^{-7}	1.03×10^{-9}
	(1×10^{-5})	(1×10^{-5})	(1×10^{-5})
Pseudo GPT-3.5 $\times$ t $\times$ log(rank)	-5.15×10^{-9}	-1.26×10^{-7}	2.73×10^{-9}
	(1×10^{-5})	(1×10^{-5})	(1×10^{-5})
Observations	519,957	519,957	519,957

Table 20: Falsification Test: Heterogeneous Impact of LLM on Centrality by Pre-GPT-3.5 Rank
This table presents the placebo test for Table 4. The estimation results from centrality analysis with logged static rank as the moderator. Standard errors clustered at domain and year-month level. ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, *dag*: $p < 0.1$.

Network Visualization

To assess the robustness of the visual patterns, we reproduce the period comparison using two standard graph-sampling views that emphasize complementary parts of the network. In all cases, we construct the same underlying tag co-occurrence graph by aggregating monthly co-appearances of tags on questions within each period; we color nodes by period-specific type (Popular/Niche/New), place nodes on k-core rings with the same radii in both periods, and draw only strong ties using the Pre-anchored weight cutoff and a shared width scale. These choices ensure visual commensurability across panels.

To create legible visualizations from the dense full graphs, we sampled each period's graph to a target size of k=600 nodes using three distinct algorithms. Each algorithm leverages the pre-computed average Weighted degree centrality as its guiding metric, denoted as $m(v)$ for a node v.

Metric-Stratified Neighbor Expansion

This method first selects a set of seed nodes in a stratified manner to ensure broad coverage of node types and importance levels, and then expands the sample to include their neighbors. We form a seed set that stratifies by tag type (Popular/Niche/New) and by importance deciles, and then include each seed's immediate neighbors (RN+N). The process is detailed in Algorithms 1 and 2 Ahmed et al. [2013].

This view preserves local neighborhoods around a heterogeneous set of reference points and is commonly used to capture both central and peripheral fabric in one sample. Post-treatment panel in Figure 11 (b) illustrates that the inner shells occupy a larger radius (i.e., lose their coreness to the network) and a more populated outer annulus, highlighting the fragmentation towards more

specialized domains. The qualitative implications match the observations in Figure 11.

However, since this sampling technique tries to sample from all domain types, namely, Popular, Niche, and New, we observe that the niche domains in both the pre-and post-LLM periods are on the outer rings of the network. More interestingly, we find that there are no domains at the center of the network.

Stratified Seed Selection

Require: Full graph $G = (V, E)$, metric map $m(v)$, type map $t(v)$, number of seeds N_{seeds}

Ensure: Set of seed nodes S

1. Initialize $S \leftarrow \emptyset$
2. Partition nodes V into bins $B_{type,decile}$ based on node type and metric decile.
3. Create a list of all non-empty bins $\mathcal{B}$.
4. Shuffle the order of bins in $\mathcal{B}$.
5. $N_{per_bin} \leftarrow \lceil N_{seeds}/|\mathcal{B}| \rceil$
6. **for** each bin $B \in \mathcal{B}$ **do**
   (a) Sort nodes $v \in B$ descending by $m(v)$.
   (b) $S \leftarrow S \cup \{\text{top } N_{per_bin} \text{ nodes from sorted } B\}$
   (c) **if** $|S| \geq N_{seeds}$ **then break**
7. **end for**
8. **return** first N_{seeds} nodes from S.

Figure 8: Stratified Seed Selection

Metric-Ranked Core

This is a straightforward importance sampling technique that selects the top-k nodes as ranked by the guiding metric. This method is designed to capture the subgraph induced by the most central nodes. Its implementation is described in Algorithm 3.

Multi-seeded Random Walk with Restart (RWR)

This method explores the graph topology by simulating multiple short random walks that can restart at a set of pre-selected start nodes. To ensure a balanced exploration of different graph

Algorithm 2: Neighbor Expansion Sampling

Require: Full graph $G = (V, E)$, target size k, seed nodes $S_{initial}$

Ensure: Set of sampled nodes V_{sample}

1. Initialize $V_{sample} \leftarrow S_{initial}$
2. **for** each seed node $s \in S_{initial}$ **do**
   (a) Add s and its neighbors $\mathcal{N}(s)$ to V_{sample}.
   (b) **if** $|V_{sample}| \geq k$ **then break**
3. **end for**
4. **if** $|V_{sample}| < k$ **then**
   (a) Let $V_{rem} \leftarrow V \setminus V_{sample}$
   (b) Add $k - |V_{sample}|$ nodes chosen uniformly at random from V_{rem} to V_{sample}.
5. **end if**
6. **return** first k nodes from V_{sample}.

Figure 9: Neighbor Expansion Sampling.

Metric-Ranked Core Sampling

Require: Full graph $G = (V, E)$, metric map $m(v)$, target size k

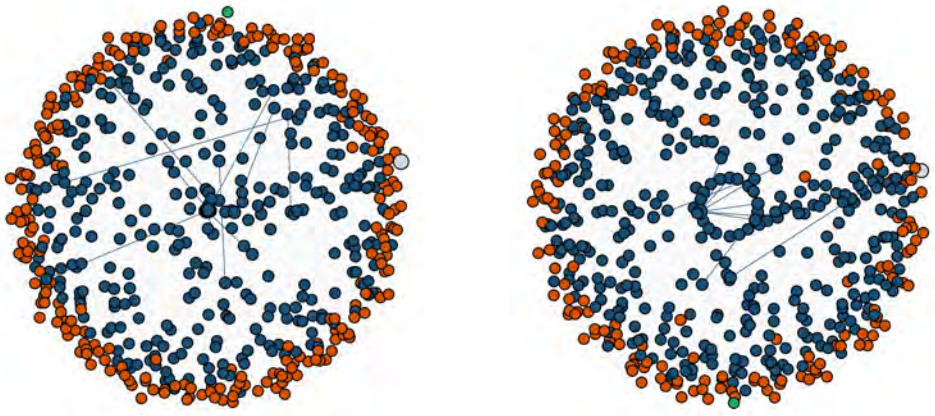
Ensure: Set of sampled nodes V_{sample}

1. Create a list of all nodes V.
2. Sort V in descending order based on metric score $m(v)$.
3. $V_{sample} \leftarrow \{\text{top } k \text{ nodes from sorted } V\}$
4. **return** V_{sample}.

Figure 10: Metric-Ranked Core Sampling

regions, start nodes were chosen from the top, middle, and lower deciles of the metric distribution. The procedure is outlined in Algorithm 4.

We select a small set of seeds from the top, middle, and bottom of the importance ranking and perform multiple RWRs (restart probability α), accumulating visited nodes until K unique nodes are obtained. This construction favors traversed, well-connected regions while still sampling the periphery through restarts as described in **?**Ribeiro and Towsley [2010].



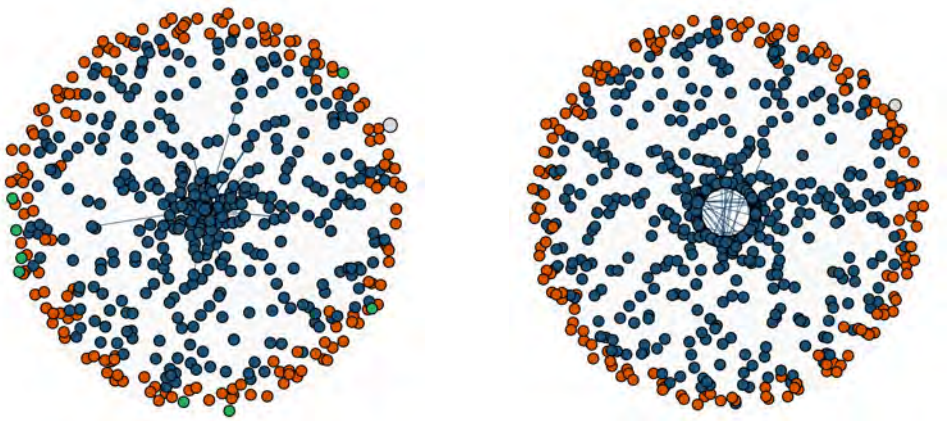
(a) Knowledge structure before Nov 2022 with blue nodes representing popular domains, orange representing niche domains, and green representing new domains.

(b) Knowledge structure after Nov 2022 with blue nodes representing popular domains, orange representing niche domains, and green representing new domains.

Figure 11: Network structure representing co-occurrence patterns of technical domains in StackOverflow before and after LLMs. We sample the original network with n = 600 using the Metric-stratified neighbor expansion sampling algorithm. The circles denote a knowledge domain and the edges denote important connections between domains.

Under identical ring radii for pre and post-LLM panels, the post-LLM network in Figure 12 (b) exhibits (i) an inner shell displaced outward relative to the pre-LLM network, core domains shift outwards, and (ii) greater occupancy of intermediate/outer shells. As with core-aware metric and metric-stratified neighbor expansion sampling techniques, the visual evidence is consistent with a redistribution of mass away from popular inner-core domains toward mid- and outer regions.

Across all three sampling views, we observe the same qualitative invariants: the highest-coreness shell is less occupied in Post; outer shells are at least as populated, with a greater visible presence of Niche tags; and strong ties are concentrated within shells rather than spanning them. Because node sets differ by construction and by period, these figures are descriptive; the claims in the main text rest on the accompanying period-level statistics (centralities, shell counts, and assortativity), with which the visual patterns are consistent.



(a) A. Knowledge structure before Nov 2022 with blue nodes representing popular domains, orange representing niche domains, and green representing new domains.

(b) B. Knowledge structure after Nov 2022, with blue nodes representing popular domains, orange representing niche domains, and green representing new domains.

Figure 12: Network structure representing co-occurrence patterns of technical domains in StackOverflow before and after LLMs. We sample the original network with n = 600 using the Metric-stratified neighbor expansion sampling algorithm. The circles denote a knowledge domain, and the edges denote important connections between domains.

Graph Layout and Visualization

We visualize each sampled subgraph using a k-core ring layout rather than a force-directed map. Node radius encodes coreness (k-shell): higher coreness lies closer to the center. For a given technique, the mapping from shell to radius is held fixed across Pre and Post, so ring indices are directly comparable; angular positions within a ring are arbitrary (small jitter prevents overlap). Node color marks type (Popular, Niche, New). To emphasize the backbone and keep panels commensurate, we draw only strong ties defined by the Pre-period 75th percentile of edge weights (computed on the technique's subgraph) and apply the same absolute cutoff to Post. Edge widths vary with weight under a winsorization cap fixed at the Pre 99th percentile; edge hues indicate tie type (e.g., Popular–Popular). Faint, equally spaced reference rings are overlaid to aid visual comparison of shell occupancy. Using three complementary sampling views (Metric-Ranked Core; Metric-Stratified Neighbor Expansion; multi-seed RWR) provides a robustness check that the qualitative patterns are not an artifact of one sampling strategy.

7 Academia Community

Table 21 illustrates the results for the change in network metrics for the Academia community in StackExchange. We do not find any significant shifts in the network metrics for this community. This indicates the distinct dynamics of technical communities compared to discussion-oriented ones like Academia. This also serves as a robustness check, suggesting that the main effects identified

Algorithm 1: Multi-seeded Random Walk with Restart Sampling

Require: Full graph $G = (V, E)$, target size k, start nodes S_{rwr}, total steps N_{steps}, restart probability α

Ensure: Set of sampled nodes V_{sample}

1. Initialize $V_{sample} \leftarrow S_{rwr}$
2. Initialize current walker positions $P \leftarrow S_{rwr}$
3. **for** $i = 1$ **to** N_{steps} **do**
   (a) Select current walker $p \leftarrow P[i \pmod{|P|}]$
   (b) **if** a random float in $[0, 1) < \alpha$ **then**
   $p \leftarrow$ a node chosen uniformly at random from S_{rwr}.
   (c) **else**
   if p has neighbors $\mathcal{N}(p)$ **then**
   $p \leftarrow$ a neighbor chosen uniformly at random from $\mathcal{N}(p)$.
   (d) Add new position of p to V_{sample}.
   (e) **if** $|V_{sample}| \geq k$ **then break**
4. **end for**
5. **return** first k nodes from V_{sample}.

Figure 13: Multi-seeded Random Walk with Restart Sampling.

width = 0.95center

	Total Questions	New Specialization	New-Domain	Niche-Containing
	(1)	(2)	(3)	(4)
	OLS	Logit	Multinomial	Multinomial
GPT-3.5	5.71	0.179†	0.927	-0.286*
	(12.4)	(0.102)	(0.596)	(0.137)
GPT-3.5 × t	2.82	0.0003	0.075	0.021
	(2.91)	(0.025)	(0.127)	(0.033)
GPT-3.5 × New Spec.	27.7*			
	(12.5)			
GPT-3.5 × t × New Spec.	1.06			
	(3.03)			
GPT-3.5 × log(Rank)				
GPT-3.5 × t × log(Rank)				
Observations	68	9,384	5,336	5,336

Table 21: Impact of LLM on the Academia Community.

This table presents estimation results from applying all analyses to the academic community. We replicate the models in Table 1 on academia community and present the results in columns (1) and (2), the multinomial model in Table 2 is replicated and presented in columns (3) and (4). ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, †: $p < 0.1$.

in StackOverflow using the RDiT method are not driven by spurious data variations.

8 Companion Validation Exercise: Comparing Answer Quality

Anecdotal Evidence

To complement our quantitative findings, we provide anecdotal evidence illustrating the qualitative differences between the types of questions that remain on StackOverflow and the capabilities of large language models (LLMs) in addressing them. These examples highlight why questions requiring the recombination of niche knowledge domains are often better suited to be asked in such online communities, whereas questions on more common domains may be well handled by LLMs.

Example 1 Figure 14a shows a user struggling with the `riverpod` package in `dart`, where they want to combine two providers so that the resulting provider always emits the most recent value from either of them. The answer provided by ChatGPT Model 4o (Figure 14b) offers a clean, idiomatic solution: it encapsulates the logic in a state-owning unit (`StateNotifier`) that listens to both inputs and updates its state accordingly. While the snippet uses the slightly older `StateNotifier` API (which can be updated to the modern `Notifier`), the structure is robust, lifecycle-safe, and easily extensible. In contrast, the StackOverflow answer (Figure 14c) attempts to avoid legacy tools

but does so by mutating `ref.state` inside a plain `Provider`, which is fragile and in fact incorrect in newer versions of Riverpod. This illustrates how, for well-documented and popular frameworks such as `dart` and `riverpod`, LLMs may provide more reliable, directly usable code than community answers.

Example 2 Figure 15a shows a user encountering import errors while working with the `calamine` crate in `rust` to read Excel files. Here, the roles are reversed: the ChatGPT Model 4o answer (Figure 15b) identifies some typos and casing issues, but its response remains generic and incomplete, leaving out key details such as the correct dependency specification and version updates. By contrast, the StackOverflow answer (Figure 15c) provides a comprehensive fix: it not only corrects identifier casing and syntax but also highlights that the dependency changed in October 2022, specifies the correct `Cargo.toml` entry, and links to documentation and examples. In this case, the community answer clearly outperforms the LLM by situating the error in its ecosystem context. This illustrates why questions involving library versioning, dependency management, and ecosystem-specific quirks remain better suited for online communities.



(a) Question asked on StackOverflow in the post-LLM period.

(b) Answer provided by ChatGPT Model 4o.

(c) Answer provided by StackOverflow users.

Figure 14: A sample question concerning a popular domain asked on StackOverflow with a response received from GPT 4o and in the StackOverflow community.

Data Construction for Evaluating Answer Quality

We constructed an evaluation dataset from a balanced sample of Stack Overflow questions posted between January 2019 and March 2024. The sample contains 25,200 questions, with 12,600 classified as general and 12,600 classified as novel. For each question, we retained the original title, body, tags, posting date, novelty label, and a human reference answer. The human reference is the accepted Stack Overflow answer when one is available; otherwise, we use the top-scored answer to that question. This design allows us to compare model-generated answers against answers that were actually provided in the Stack Overflow setting, while retaining coverage for questions that do not have an accepted answer.



(a) Question asked on StackOverflow in the post-LLM period.

(b) Answer provided by ChatGPT Model 4.0.

(c) Answer provided by StackOverflow users.

Figure 15: A sample question concerning a niche domain asked on StackOverflow with a response received from GPT 4o and in the StackOverflow community.

For each question, we generated one GPT-3.5 answer. That is, the answer-generation step was not repeatedly sampled for the same model-question pair. In the pipeline, each generated answer is indexed by question, model, and generation index; in the main evaluation, this generation index is set to one draw per question-model pair. The prompt asked the model to answer the Stack Overflow question directly, using the question title, body, and tags. The answer-generation prompt instructed the model to provide a focused Stack Overflow-style answer, prefer simple correct solutions, include runnable code when appropriate, state assumptions when needed, and avoid invented APIs or nonexistent functions. Of the planned GPT-3.5 generations, three failed because the input exceeded the model's context window; the evaluation is therefore complete, conditional on successfully generated answers.

Answer Quality Evaluation.

We then evaluated each GPT-3.5 answer against the corresponding human reference using GPT-5-mini as an LLM judge. The evaluation has two components. First, we conducted pairwise comparisons. In each comparison, the judge received the original Stack Overflow question and two blinded candidate answers: one human reference answer and one GPT-3.5 answer. The judge was asked to choose whether Answer A, Answer B, or neither answer was better for a developer reading Stack Overflow. The judge's instructions prioritized correctness first, then completeness, then clarity and conciseness, and explicitly penalized hallucinated APIs, wrong logic, and misleading advice. To reduce position bias, the order of the two answers was randomized across repetitions: in some comparisons, the human answer appeared first, and in others, the GPT-3.5 answer appeared first. This randomization was deterministic and reproducible, using the question identifier, model, generation index, repetition number, and random seed. Each question-answer pair was evaluated five times in this pairwise format. These five repetitions are repeated evaluations of the same generated answer against the same human reference, not five independently generated GPT-3.5

answers.

Second, we conducted rubric-based evaluations. For each generated GPT-3.5 answer, the judge received the original question, the human reference answer, and the GPT-3.5 answer. The judge then scored the GPT-3.5 answer on six dimensions: correctness, consistency, conciseness, comprehensiveness, efficiency, and clarity. Each dimension was scored on a five-point scale, where higher values indicate stronger performance. The rubric instructions allowed alternative correct solutions to receive high scores even when they differed from the human reference, but penalized hallucinated APIs, incorrect library use, unsafe advice, or failure to answer the question. Each rubric evaluation was repeated three times.

Answer Quality Analysis. The main outcome from the pairwise evaluation is the share of comparisons in which the GPT-3.5 answer beats the human reference, the share in which the human reference beats GPT-3.5, and the share classified as ties. Because the same generated answer is evaluated multiple times under randomized orderings, these outcomes should be interpreted as repeated judge assessments of a fixed model answer rather than as variation across multiple generated answers. In the full GPT-3.5 comparison, the human reference wins 64.1 percent of pairwise comparisons, while GPT-3.5 wins 35.1 percent, and ties account for less than one percent. The human advantage is larger when the reference answer is an accepted answer rather than a top-scored fallback answer. It is also larger for novel questions: GPT-3.5 wins 36.9 percent of comparisons on general questions but only 33.3 percent on novel questions, while the human-reference win rate rises from 62.2 percent on general questions to 65.9 percent on novel questions.