Innovative Collaboration: Generalists, Specialists and Teams

Lena Greska*

March 17, 2025

Abstract

In innovative teamwork, two forces collide: Higher productivity from specialization of team members and the need to coordinate among them. Do the benefits from specialization outweigh its costs? Using data from online machine learning competitions, in which small teams of computer scientists develop solutions for machine learning problems, I show that generalist teams outperform specialist teams. This pattern is consistent with predictions from a theoretical framework in which teams trade-off the higher productivity of specialists against increased coordination costs. I present several sets of evidence pointing towards coordination costs as a channel: First, complex problems, which require more coordination, induce larger generalist-specialist performance differences. Second, a reduction in coordination costs through the availability of ChatGPT decreases generalist-specialist performance differences. Finally, I do not find support for alternative mechanisms such as management or social skills. I make a methodological contribution by developing a novel measure of specialization based on the semantic diversity of a computer scientist's code.

*Ludwig-Maximilians-Universität München, lena.greska@econ.lmu.de

This project would have been impossible without the support and guidance of my advisors, Fabian Waldinger and Claudia Steinwender. I thank Ran Abramitzky, Daron Acemolglu, David Autor, Pierre Azoulay, Svenja Frieß, Ingrid Hägele, John Horton, Robert Johannes, Paul-Henri Moisson, Friederike Reichel, Laura Rosendahl Huber (discussant), Carlo Schwarz and seminar audiences at LMU Munich, MIT and Stanford University, as well as conference participants at RISE7 for insightful comments and suggestions. I am grateful to Philipp Eller, Igor Gilitschenski, Christof Henkel, Addison Howard, Daniel Muthukrishna, and Maxim Tabachnyk for sharing valuable insights into collaboration in computer science, machine learning competitions and their experiences on Kaggle. I am particularly indebted to Valentin Hoffman, whose advice in computational linguistics was pivotal to the design of the specialization index. Krisha Agrawal provided excellent research assistance. I gratefully acknowledge financial support from Joachim Herz Foundation and the German Research Foundation (DFG) through CRC-TRR 190.

1 Introduction

Economists have long recognized specialization as fundamental to prosperity. In the first chapter of *The Wealth of Nations*, "Of the division of labour", Adam Smith argued that "The greatest improvements in the productive powers of labour [...] seem to have been the effects of the division of labour." (Smith, 1776). Similar arguments for specialization have been made in science and innovation. Here, the need to specialize is particularly pressing, since innovation builds on prior knowledge, and a single person cannot know everything (Jones, 2009). The well-documented rise in scientific specialization over the last century (Jones, 2009) has coincided with an increase in scientific teamwork (Wuchty et al., 2007).

Yet, teamwork and specialization are hard to reconcile in practice: Efficient teamwork requires coordination. Coordination is more demanding when individual team members are very specialized. For example, it becomes harder to understand and adapt to others' work, integrate individual contributions into a cohesive whole, and provide feedback. In this paper, I investigate the trade-off between specialization and coordination in scientific teamwork. I show theoretically that a team of generalists – scientists with low levels of specialization – produces the highest quality innovation when coordination costs are high. I then test the predictions of the model in the context of online machine learning competitions. Employing various econometric strategies to isolate the team type effect from the effect of individual team members, I find that generalist teams develop higher quality innovation than specialist teams. I present a range of evidence which points to coordination costs as the mechanism that drives generalist-specialist quality differences rather than other factors.

In my theoretical framework, scientists with heterogeneous levels of specialization work alone or collaborate to solve problems with varying complexity. Similar to Becker and Murphy (1992), on which my model builds, each problem requires a set of tasks to be executed in order to solve it. The quality of the problem's solution depends on the quality of execution of these sub-tasks. In a blend of Deming (2017) and Garicano (2000), scientists' specialization is modeled as different ex-ante task-specific productivities. A generalist, with zero specialization, can execute all tasks but each with lower productivity. The more specialized a scientist, the fewer tasks they can solve, but the higher their productivity for these tasks. Here, I depart from Becker and Murphy (1992), where all team members are equally specialized ex-post as a result of dividing tasks in a team. When collaborating, scientists can split the tasks necessary to solve a problem, and produce quality more efficiently. However, they incur coordination costs, which reduce the quality of their solutions. Importantly, coordination costs increase (1) in each team member's degree of specialization, and (2) in the complexity of the problem they are trying to solve. In this aspect, I deviate from other models of scientific teamwork

and specialization, which often assume costless aggregation of knowledge in teams (Jones, 2009).

I derive two predictions from this framework: First, if coordination costs are sufficiently high, a team of generalists solves a problem with the highest quality. Thus, a decrease in coordination costs will lower or eliminate quality differences between generalists and specialists. Second, complexity increases quality differences between generalist and specialist teams if coordination costs are sufficiently high. These predictions allow me to develop empirical tests to pin down coordination costs as the mechanism behind observed quality differences in teams' innovative output.

Testing these predictions requires an empirical setting that allows me to observe how well different types of teams solve problems *holding the problem constant*, and how well different problems get solved, *holding team types constant*. To compare solutions, I need to be able to construct an objective measure of solution quality. To accurately categorize teams, I need to quantify each team member's specialization. Crucially, the setting needs to allow me to isolate the effect of individual team members' ability from the effect of team type on quality. Online machine learning competitions provide an optimal environment for studying specialized scientific teamwork as they satisfy all these criteria. With a long tradition in computer science, specifically in machine learning and artificial intelligence (Koch and Peterson, 2024), the objective of these competitions is to develop an algorithm that solves a given, well-specified problem. The resulting algorithm's quality is evaluated according to a pre-specified, objective performance metric. I use data from Kaggle.com (Kaggle), a large online machine learning platform which provides infrastructure for (research) institutions and corporations to host machine learning competitions.[1] Kaggle users are computer science professionals like artificial intelligence researchers, machine learning engineers and data scientists. Kaggle's features are particularly suitable for my study: First, problems – competitions – are well-defined, and have a large number of teams and solo competitors trying to solve them. Second, the quality of solutions is measured in an objective way for all teams attempting to solve the problem: The host of the competition specifies a quality metric, and all submitted solutions are ranked according to this criterion. Third, Kaggle users also publish code on the platform, from which I construct my measure of specialization. Fourth, users frequently work alone. This enables me to observe individual performance independently of team performance, forming the foundation of my three-pronged empirical strategy: Explicitly controlling for observable heterogeneity between specialists and generalists, employing user-fixed effects to account for time-invariant unobservables, and conducting an event study based on team formation timing.

[1] As of November 2024, Kaggle has more than 21 million global users and hosted 407 competitions awarding a total of USD 22.3 million in prizes.

I develop a new measure of specialization for the computer sciences based on the breadth of a computer scientist's code portfolio. Intuitively, to solve a computer science problem, a computer scientist needs to write code. Broader coding skills reflect an ability to tackle a broader range of problems, and hence, less specialization. To quantify the breadth of coding skills, I use CodeT5+ (Wang et al., 2023), an open-source large language model (LLM), trained on a substantial corpus of code and comments from GitHub. With CodeT5+, I transform code into "embeddings" – vector representations capturing the semantic meaning of code. Embeddings allow me to quantitatively compare different code snippets by calculating cosine similarity, a standard metric from natural language processing used to assess the conceptual similarity of text. My specialization measure then is the average cosine similarity between all pairs of code instances written by a given scientist. A high average similarity indicates that the scientist's code tends to cluster within a narrow area, suggesting specialization. Conversely, a low average similarity implies the scientist works across a broader range of coding domains. Scientists with above-median similarity scores (i.e., lower breadth) are classified as specialists, while those below the median are classified as generalists.

Using my measure of specialization and quality measures from Kaggle, I show that generalist teams achieve higher quality solutions than specialist or mixed teams. In my preferred specification, a solution by a team comprised solely of generalists is ranked 4 percentiles higher than a solution by a specialist team, corresponding to placing on average 50 absolute ranks higher. In addition, generalist teams are 8 percentage points more likely to win a medal, a prestigious labor market signal, and 5 percentage points more likely to reach the coveted top three positions in a competition associated with large monetary prices. Mixed teams place between generalist and specialist teams.

I employ a range of empirical strategies that support the interpretation of these quality differences as fundamental differences between specialist and generalist teams, over and above differences between individual generalists and specialists. First, I control for a large set of individual characteristics to capture any observable heterogeneities other than specialization that might influence solution quality, like experience or ability. Second, to account for time-invariant unobservable scientist characteristics that might differ between generalists and specialists, I estimate a regression with user fixed effects. Third, to address concerns that performance in other competitions is not a good counterfactual for performance at a given task, I make use of a unique feature of Kaggle: Competitors can submit preliminary solutions during a competition to get a signal of the quality of their solution. Since some teams only form during the competition, I implement an event study design similar to Lemus and Marshall (2024) to assess how the quality of preliminary solutions changes before and after team formation. Generalist teams outperform specialist teams in all empirical approaches.

After establishing solution quality differences between generalist and specialist

teams, I turn to testing the mechanism that creates these differences in the model, coordination costs. Since I cannot measure coordination costs directly, I use two shifters of coordination costs to assess their impact on generalist and specialist teams. First, I split the sample into high and low complexity competitions. I can only detect quality differences in high complexity competitions, indicating that complexity indeed mediates coordination costs as suggested by my theoretical model. Second, I use the introduction of ChatGPT as an exogenous shock to coordination costs. ChatGPT can smooth some of the frictions in specialized collaboration, for instance by explaining unfamiliar concepts and the function of unfamiliar code in particular, translating one coding language into another or writing code to integrate disjoined components of an algorithm. Again, I split the sample into competitions before and after the introduction of ChatGPT, and only find quality differences in the period prior to the introduction of ChatGPT.

I also examine a range of alternative mechanisms suggested by the literature (e.g., Deming, 2017; Weidmann and Deming, 2021; Ahmadpoor and Jones, 2019; Adhvaryu et al., 2023; Freund, 2024; Minni, 2024; Weidmann et al., 2024): higher effort by generalist teams, better management in generalist teams, higher social skills in generalist teams, and better collaborator matching in generalist teams. I do not find any evidence in support of these alternative mechanisms. Taken together, my findings indicate that coordination costs indeed reduce the productivity of specialist collaboration in innovation.

This paper contributes to the literature on specialization and teamwork in science and innovation. A large body of research has documented an increasing dominance of teams in innovation (Wuchty et al., 2007; Jones, 2009; Ahmadpoor and Jones, 2019; Pearce, 2023) as well as increasing specialization (Jones, 2009; Agrawal et al., 2016). The implications of these two trends for scientific productivity have often been examined through the lens of "ideas being harder to find" (Bloom et al., 2020), specifically, that implementing ideas increasingly requires differentiated expertise. Allocca (2024) suggests that teams with more diverse educational backgrounds are more likely to successfully complete research projects. Pearce (2023) finds that patents by research teams with more diverse expertise receive more citations. While these studies consider the diversity of expertise embodied by the team, I focus on the diversity of expertise embodied by one individual, their *specialization*, and how this impacts their productivity in scientific teamwork. This dimension has been less examined by the literature. Notable exceptions are Teodoridis (2018), who documents changes in the rate of generalist-specialist collaboration after a technology shock that substituted for specialist skills, and Teodoridis et al. (2019), who provide evidence for individual generalists outperforming specialists when the overall rate of innovation is slower. To the best of my knowledge, there is no paper that empirically examines the effect of generalist-specialist team composition on innovative productivity.

My paper also contributes to the largely theoretical literature on the trade-off between specialization and coordination in teamwork in organizational economics. Starting with Becker and Murphy (1992), multiple studies show how decreases in coordination costs, sometimes modeled as communication costs (e.g. Bolton and Dewatripont, 1994), allow for higher levels of profitable specialization in teams and hierarchies (e.g. Garicano, 2000; Garicano and Rossi-Hansberg, 2006). Conversely, Dessein and Santos (2006) show how decreases in communication costs can lead to *lower* levels of specialization within an organization by allowing the organization to become more adaptive to new information. However, none of these incorporate heterogeneous levels of specialization. Recent work by Freund (2024) models specialization as ex-ante differences in the dispersion of task-specific productivity, similar to my framework, but there are no coordination frictions within the team. Perhaps closest in spirit to my theoretical framework is Deming (2017), where workers with different task-specific skills coordinate on "trading" tasks to produce a final good. The efficiency of this trade is determined by social skills, which are orthogonal to task-specific skills, i.e. specialization. I endogenize coordination costs to a team member's specialization to more explicitly capture the trade-off between specialization and coordination.

I also contribute to the literature on what makes teams succeed. Recent contributions have highlighted social skills (Deming, 2017; Weidmann and Deming, 2021; Adhvaryu et al., 2023), collaboration experience and team-specific human capital (Jaravel et al., 2018; Chen, 2021), and leadership (Minni, 2024). I add generalists to this list. Finally, a small body of research has explored teamwork and innovation on Kaggle and similar platforms, with a specific focus on tournament effects (Boudreau et al., 2011, 2016; Lemus and Marshall, 2021, 2024). For example, Lemus and Marshall (2024) find that teamwork improves performance compared to solo competitors. I take this finding further: How can we shape teams to generate maximum innovation?

2 A Model of Collaboration between Generalists and Specialists

In this section, I develop a model of scientific collaboration between specialized scientists. Scientists with heterogeneous levels of specialization solve problems alone or in teams. Specialization affects scientists' problem solving in three ways: Which problems a scientist can solve, how well scientists can solve a problem, and how well scientists collaborate. I derive two testable predictions from this framework. To keep notation and exposition as simple as possible, the focus of my analysis is on teams of two scientists.

2.1 Set-Up

Problems Scientists solve problems p. To solve a problem, scientists have to execute a continuum of tasks Θ_p. Like Becker and Murphy (1992), I assume tasks are perfect complements, where the problem's overall solution quality is determined by the quality of the worst-executed task. While stringent, this assumption keeps the model tractable and is not too far from reality. Consider, for instance, conducting empirical research as consisting of two tasks, data cleaning and regression analysis. If the data are not cleaned properly, no amount of well-specified regressions can lead to sound conclusions. Conversely, even perfect data will not save a misspecified regression. Problems differ along two dimensions, (1) the amount $\Theta_p \in (0, 1]$ of different tasks that need to be executed to solve the problem and (2) complexity γ_p. I formalize complexity as a factor that makes teamwork more difficult.

Scientists Scientists i have different levels of specialization σ_i. Specialization has three effects on a scientist's problem solving ability. First, it determines how many different tasks a scientist can execute, where $b_i \in (0, 1]$ is the range of tasks the scientist can execute. The range of tasks a scientist can solve is inversely proportional to their specialization, i.e. the higher a scientist's level of specialization, the fewer tasks the scientist can execute ($\sigma_i = \frac{1}{b_i}$). For expositional simplicity, I consider a scientist who can execute all tasks ($\sigma_i = b_i$ =1) a generalist. Specialization also affects how well a scientist can execute a task. Conditional on being able to execute a given task, a scientist with a higher level of specialization executes the task with higher quality. These first two effects are similar to the most prominent ways of modelling specialization in the literature – heterogeneous task-specific productivity (Deming, 2017; Freund, 2024) and whether or not a task can be executed (Garicano, 2000; Jones, 2009). The third effect is novel and where the main contribution of my model lies. Specialization also determines how easy it is for a scientist to collaborate. Teamwork is more difficult if scientists are very specialized. In the empirical research example, imagine two scientists are collaborating on a study, but one scientist only knows how to clean data, and the other only how to run regressions. Coordinating on how to conduct the study would be extremely difficult in this case.

Collaboration Scientists can choose whether to try and solve a problem on their own or whether to collaborate. For simplicity, I abstract from collaborator search. Instead, I consider a world in which, for each problem, a scientist is matched at random with another scientist, and their only choice is whether to collaborate or not. When collaborating, the two scientists have to split the Θ_p tasks among themselves. Let w_i denote the share of tasks executed by scientist i. When collaborating, the team of scientists incurs coordination costs C. Coordination costs capture all the frictions

that arise in team production but not when working alone, for example deciding and negotiating how to split tasks, which tasks to bundle, or determining how to aggregate the output from sub-tasks into the final product.

Problem Solving and Payoffs A problem p is solved by scientist i with quality Q_p:

$$Q_p = \min_{0 \leq \theta \leq \Theta} q(\theta) \qquad \text{where} \quad q(\theta) = \begin{cases} 0 & \text{if } \theta \notin b_i \\ q(w_i, \sigma_i, e_i) & \text{if } \theta \in b_i \end{cases} \tag{1}$$

e_i is the effort spent by scientist i on the θ^{th} task. The quality of execution of each task thus depends on the amount of tasks that one scientist needs to execute w_i, a scientist's specialization σ_i, and the effort spent on each task e_i. Intuitively, $\partial q(w_i, \sigma_i, e_i)/\partial e_i \partial \sigma_i > 0$, meaning that conditional on being able to execute a task, a more specialized scientist has a higher marginal productivity of effort. The marginal productivity of effort however decreases in the amount of tasks a scientist has to execute, i.e. $\partial q(w_i, \sigma_i, e_i)/\partial e_i \partial w_i < 0$. Similar to assumptions made, for example, by Deming (2017) and Becker and Murphy (1992), this allows scientists to become more productive as they focus on fewer tasks, and is the main mechanism through which teamwork can be beneficial. An intuitive interpretation is that when scientists have to execute many task, they have to split their attention and get distracted more easily, or cannot learn as much about the individual task.

Scientists maximize payoffs, equal to a problem's solution quality minus the total cost of effort $w_i c(e_i)$, where $c(e_i)$ is the effort spent on one task θ. Since tasks are symmetric and perfect complements, the total cost of effort is the cost of effort for each task multiplied by the share of tasks executed by one scientist, w_i. That is, when working alone, scientist's payoffs Π_p^a from solving problem p are given by:

$$\Pi_p^a = \begin{cases} 0 & \text{if } \Theta_p > b_i \\ \min\limits_{0 \leq \theta \leq \Theta_p} q(\Theta_p, \sigma_i, e_i) - \Theta_p c(e_i) & \text{if } \Theta_p \leq b_i \end{cases} \tag{2}$$

where $\Theta_p c(e_i)$ is the cost of effort of executing all Θ_p tasks required to solve a problem. Scientists receive utility from solving a problem well, such that the quality of a problem's solution maps directly into scientists' payoffs. When collaborating, scientists' payoffs are equal to *half* the problem's solution quality. However, scientists can achieve quality more efficiently because they can split tasks to increase their marginal productivity of effort. They can also potentially collaborate with a more productive, i.e. more specialized, scientist then themselves. Splitting tasks also means that scientists have lower total effort costs for the same task-specific effort e_i, since they execute fewer tasks. Yet, teams also incur coordination costs. A scientist i's payoffs Π_p^t from solving problem

p in a team with scientist j with specialization σ_j are given by:

$$\Pi_p^t = \frac{1}{2}\{\min_{0\leq\theta\leq\Theta_p} q(w_i, \Theta_p - w_i, \sigma_i, \sigma_j, e_i, e_j) - C(\gamma_p, \sigma_i, \sigma_j)\} - w_i c(e_i) \tag{3}$$

where e_j is the effort executed by scientist j. That is, payoffs crucially depend on the other team member. Coordination costs $C(\gamma_p, \sigma_i, \sigma_j)$ reduce a problem's solution quality and hence payoffs for both team members, and are an increasing function of the problem's complexity γ_p, and both team members' degree of specialization, i.e. $\partial C/\partial\gamma_p > 0$ and $\partial C/\partial\sigma_i > 0$. Importantly, complexity has a stronger impact on more specialized scientists, i.e. $\partial C/\partial\gamma_p\partial\sigma_i > 0$. Why is that? Going back to the previous example, imagine the two scientists now have to implement an econometric strategy that requires careful data preparation, for example in matching control observations on a complex range of characteristics. If the team member who cleans the data has no understanding of the identification challenges, they might focus on the wrong characteristics and produce a matched sample that exacerbates bias. The same complexity is less harmful if the team member who cleans the data also understands identification.

2.1.1 Choices

Scientists make three main choices. First, they choose whether to collaborate or not. Second, only if collaborating, they choose how to allocate tasks. Finally, they choose the optimal level of effort.

Collaboration Choice When choosing to collaborate, scientists trade off the gains from the division of tasks with collaboration costs. Two scientists form a team when the payoffs to teamwork are larger than the payoffs to working alone for both scientists in the team.

$$\frac{1}{2}\underbrace{\{Q_p^t - C(\gamma_p, \sigma_i, \sigma_j)\}}_{\text{team solution quality}} - \underbrace{w_i c(e_i^{t*})}_{\text{total cost of effort}} \geq \underbrace{Q_p^a}_{\text{individual solution quality}} - \underbrace{\Theta_p c(e_i^{a*})}_{\text{total cost of effort}} \tag{4}$$

If scientists choose not to collaborate, they either solve the problem on their own if $\Theta_p \leq b_i$ or do not participate if $\Theta_p > b_i$.

Task Division and Effort Choice Scientists choose the division of task to maximize solution quality minus total effort costs. Optimal task division and efforts are given by:

$$w_i^* = \begin{cases} \Theta_p & \text{if } n = 1, \\ \arg\max_{w_i} \quad \frac{1}{2}\min q(w_i, \Theta_p - w_i, \sigma_i, \sigma_j, e_i^*, e_j^*) - w_i c(e_i^*) & \text{if } n = 2. \end{cases} \tag{5}$$

and

$$e_i^* = \arg\max_{e_i} \quad \frac{1}{n} \min q(e_i, \sigma_i, w_i) - w_i c(e_i) \tag{6}$$

where n is 1 when working alone and 2 when collaborating. When working alone, w_i is Θ_p by definition. In a team, the optimal division of tasks will be pinned down by the fact that for both scientists, marginal productivity has to equal the marginal change to total effort costs for both scientists, and that each scientist has to produce the exact same quality in each sub-task.

2.2 Predictions

In the model, scientists trade off higher productivity in specialized team work and coordination costs. I derive two predictions from this framework to test whether coordination costs really do limit the returns from specialization. For more detailed derivations, see Appendix A.

Prediction 1. *If coordination costs are sufficiently high, a team of generalists produces the highest quality solution. Conversely, a decrease in coordination costs will reduce quality differences between generalist and specialist teams.*

Proof: This follows directly from applying the envelope theorem to the function of team solution quality.

Prediction 2. *An increase in complexity will increase the quality differences between generalist and specialist teams.*

Proof: Since $\partial C / \partial \gamma_p \partial \sigma > 0$ quality differences between teams will be larger when Prediction 1 holds.

3 Machine Learning Competitions and Kaggle

3.1 Why machine learning competitions?

In machine learning, competitions are widely used both to accelerate innovation and to evaluate the quality of innovation. Competitions were first introduced by the Defense Advanced Research Projects Agency (DARPA) in the 1980s to allocate grants for risky artificial intelligence projects (Koch and Peterson, 2024) and are still widely used (Pavão, 2023).[2] A prominent example is the 2006 Netflix Prize, in which the streaming platform

[2]Theory and empirical evidence also indicate that competitions are an efficient method to generate innovation (Lazear and Rosen, 1981; Boudreau et al., 2011, 2016; Kireyev, 2020).

Netflix awarded USD 1 million to the team that would achieve 10% higher prediction accuracy than Netflix's proprietary movie recommendation algorithm (Bennet and Lanning, 2007).[3] Competitions are also an integral part of academic computer science. For example, prestigious computer science conferences like Knowledge Discovery and Data Mining (KDD) and Neural Information Processing Systems (NeurIPS) include a dedicated competition track (Albrecht et al., 2024; Association for Computing Machinery, 2025). The basic structure of any machine learning competition has remained the same since their introduction: The host of the competition describes the problem at hand, provides the dataset as input for the desired algorithm, and sets timeline, evaluation metric and prizes. Crucially, only a subset of the dataset (*training data*) is made available to competitors to develop their algorithm. Once the competition ends, all competitors submit an algorithm. The performance of the submitted algorithm is evaluated against a dataset that was not previously known to competitors (*test data*). Whosever algorithm achieves the best performance in the test data wins the competition.

Machine learning competitions are an attractive setting for the study of innovative teamwork for several reasons. First, problems are well-defined. Second, performance is measured in a unified, objective way: ranks by performance of the algorithm.[4] This enables me to compare the performance of different team types at the exact same problem. Third, online competition platforms like Kaggle provide rich data that facilitate econometric analysis. In any team study it is necessary to isolate the effect of the team type from the characteristics of the team members (Weidmann and Deming, 2021; Bonhomme, 2022). This requires observing performance of team members when they are working alone. For many science and innovation settings this poses a challenge, since both patents and papers are increasingly produced in teams (Wuchty et al., 2007). On Kaggle, only 1% of all users never work alone, such that I can observe individual performance for almost all team members. Additionally, several features of the platform allow me to construct measures for concepts like effort and social skills to explore mechanisms. Most importantly, I can measure specialization.

3.2 Kaggle

Kaggle is a platform for data science and machine learning. Founded in 2012, its core business is providing infrastructure for companies and research institutions to host machine learning competitions, but it also serves as a learning, networking, as well as data, models and code sharing platform for computer science professionals.

[3]Note that a competition is distinct from a hackathon. While hackathons are more open-ended ideation challenges within a thematic complex, competitions are directed innovation focusing on a single computation problem.

[4]This compares to much of the academic literature in computer science: Papers benchmark their proposed algorithms to the state of the art.

As of November 2024, Kaggle has more than 21 million global users and hosted 407 competitions awarding a total of USD 22.3 million in prizes. Kaggle makes all data from their users' public activities available on their website as a downloadable dataset, Meta Kaggle (Risdal and Bozsolik, 2022). The dataset contains granular information on users, competitions, teams, discussions, and code meta data. The main focus of this paper are competitions. However, I utilize users' activities in the other areas to explore mechanisms and construct main variables.

Competitions My analysis focuses on the two most prominent types of competitions on Kaggle, *Featured* and *Research* competitions. Of all users who ever participated in any competition, 46.8% participated in a featured competition and 15.2% in a research competition.[5] Though mostly similar, featured and research competitions differ slightly in terms of the type of problems they cover – featured competitions tend to focus more on commercially applicable problems, whereas research competitions focus on scholarly and scientific problems.[6] The term competitions will refer to *Featured* and *Research* competitions from now on. Kaggle competitions proceed like typical machine learning competitions. A competition host – a company or a research institute – sets up a competition. The host describes the goal of the competition (e.g. detecting cancer from mammograms, recognizing bird voices from audio), provides labeled data, determines the evaluation metric[7], the prizes, and the competition timeline. Competitions run for three months on average, and there is an entry and team formation deadline. Appendix Table A1 lists example competitions.

Once a competition begins, users are invited to work on the problem and submit their solutions. A feature particular to Kaggle is that users can obtain preliminary feedback on the performance of their algorithm in real time. Kaggle splits the data provided by the competition host into three datasets: The training set is available to users to train their algorithms, the *public* test set is used to evaluate submissions during the competition, and the *private* test set is used to evaluate submissions once the competition ends. This split is meant to ensure that competitors get feedback on their algorithms during the competition, but that solutions are actually useful by reducing the incentive to overfit to the available data. Competitors' ranks in terms of the score of their

[5] All numbers refer to the version of Meta Kaggle used in this paper, downloaded on February 7, 2024 (Risdal and Bozsolik, 2022).

[6] I exclude the other types of competitions, *Community*, *Playground*, and *Getting Started*, from the analysis since the problems posed in these competitions are more akin to classroom problems, and performance is not incentivized. I also exclude *Recruitment* competitions, used by companies like AirBnB or Facebook as assessment centers for data scientists, in which teamwork is prohibited.

[7] Usually, these are prediction accuracy metrics commonly used in machine learning, like Root Mean Square Error or Log Loss. Some competitions additionally set infrastructure constraints, like run-time, maximum central processing unit (CPU) and graphics processing unit (GPU) usage or internet access.

solution are displayed publicly on the *leaderboard* in real time during the competition.[8] At the end of the competition, users can select two submissions to be judged against the private test set. Submission scores at this state determine the winners of the competition. I observe public and private ranks of all submissions made to any competition.

Teams Competition participants can choose to work alone or to collaborate with up to four other competitors. Competitors first enter a competition on their own and can then form or join a team at any point during the competition up to a predetermined team formation deadline. Teams can only be dissolved if the team has not made any submission. If a team wins a competition prize, the prize is divided equally among all team members. Team formation is voluntary and self-organized. For example, Kaggle users can use public message boards attached to each competition to announce they are looking for teammates[9], or use Kaggle's messaging features to contact potential collaborators. Since Kaggle displays a range of information on each user's profile, such as in which competitions a user participated in the past, and how they performed, team members' abilities are public knowledge. Anecdotal evidence from interviews I conducted with active Kaggle users suggests that these observable performance signals determine users' team formation decisions. I observe each team member's past performance, as well as team sizes and when the team was formed.

Incentives Do teams actually try to develop the best algorithm? A common worry in the contest design literature is that a paucity of monetary prizes decreases incentives of all contestants, since each contestant has a low chance of winning (e.g., Taylor, 1995). Kaggle counteracts this with providing additional reputation incentives. Each user receives a platform-wide ranking, which is a function of competition performance, and belongs to a status group. Akin to chess, the most prestigious groups are masters and grandmasters. These can be achieved by performing extremely well, but not necessarily winning a monetary prize, in competitions. Both platform ranks and status groups are displayed on a user's Kaggle profile page and are valuable signals in the labor market for computer scientists. Users frequently include this information in their résumés (Reich, 2023), and several companies explicitly hire based on Kaggle rankings and status groups. For example, GPU manufacturer NVIDIA employs a team of Kaggle grandmasters solely to participate in Kaggle competitions (NVIDIA, 2025). Finally, competitors are also intrinsically motivated to contribute to open science. Although winners are not required to do so, many share their approaches and solutions after the competition. Kaggle competitions have resulted in research papers (e.g. Bulten et al.,

[8] For a discussion of the impact of intermediate feedback in tournaments and specifically Kaggle, see Lemus and Marshall (2021). They find that intermediate feedback generates better performing algorithms.

[9] In August 2022 Kaggle additionally introduced a waving-hand icon that users and teams who are looking for teammates can display next to their name on the leaderboard.

2022; He et al., 2024), with competition winners as co-authors. Kaggle users also stress the innovation aspect of their work on Kaggle: "Normally, Kaggle is always one step ahead of science. [...] To perform really well in the competitions, one has to go an extra mile. Anyone can read what already exists. You have to come up with something new to beat 1000 other people."[10]

3.3 Measuring Specialization in Kaggle

3.3.1 Conceptual Considerations

A scientist's degree of specialization is equal to their breadth of knowledge. I thus need to find a representation of the knowledge held by one scientist. In computer science, this is what sort of code someone is able to write. For example, a computer scientist who writes code for both image data and text data has broader knowledge than a scientist who only ever writes code for text data. The first scientist is less specialized than the second. The main challenge is to quantify this distinction. Recent advances in computational linguistics make this possible through embedding models for code. These models, which also power large language models like ChatGPT, transform code into high-dimensional vectors that capture the semantic meaning of code. By learning the context in which instances of codes occur, embedding models are able to represent the relationship between two sets of code, as well as between code and plain text in an ordinal way. Two related concepts will be represented by two closely located vectors. Conversely, the distance between two vectors measures how different two concepts are. The totality of embeddings thus effectively maps the "knowledge space" of coding expertise. This feature makes embedding models ideally suited for the study of specialization.

To illustrate why, it is helpful to contrast this approach with existing measures in the literature. For example, Jones (2009) measures specialization of inventors as the probability of changing patent categories between to successive inventions. Teodoridis (2018) and Teodoridis et al. (2019) construct a diversification index based on the Herfindhal index of a scientist's paper categories. Discrete categories however fail to capture both the relationship of knowledge representations between and within categories. Imagine there are three types of code: `areg`, `xtreg` and `hpfilter`. A category-based index would assign the same level of specialization to a user that has written one `xtreg` and `areg` code each, and one that has written one `xtreg` and one `hpfilter` code, although `xtreg` and `areg` are more closely related than `xtreg` and `hpfilter`. An embedding model resolves this issue – the distance between `areg` and `xtreg` will be smaller than the distance between `hpfilter` and `areg` or `xtreg`,

[10]Christof Henkel (alias Dieter), current highest-ranking Kaggle user and a member of the Kaggle Grandmasters of NVIDIA (Kaggle.com, 2024c), in zoom interview with the author, April 2024.

respectively. An additional advantage of an embedding-based index is that it does not rely on labeled data, which is often costly or impossible to obtain. It is particularly infeasible in this context where code files are rarely labeled, although Kaggle enables authors to self-assign category tags to code files. Tag coverage is sparse, endogenous, and tags are often not meaningful (e.g., "advanced").

To quantify the "distance" between two instances of code, I use cosine similarity, a standard measure of textual similarity in computational linguistics. For each scientist, I then define specialization as the average cosine similarity between their individual code files and their core coding discipline, represented by the centroid of a scientist's code files:

$$Specialization_i = \frac{1}{K_i} \sum_{1}^{K_i} cos(v_{ki}, c_i)$$

where K_i is the total number of code files k_i written by a user i, v_{ki} is the embedding of code file k_i, and c_i is the centroid (mean) embedding of user i, constructed as the average of all of user i's code file embeddings: $c_i = \frac{1}{K_i} \sum_{1}^{K_i} v_{ki}$. In less technical terms, this measure is analogous to the variance of knowledge around an individual's core discipline, represented by the centroid embedding.[11]

3.3.2 Construction of the Specialization Index in Kaggle

In addition to competitions, Kaggle provides infrastructure for running, publishing and sharing code. Similar to competitions, users are incentivized to publish code by receiving medals, titles and a platform-wide ranking.[12] Kaggle makes the entirety of shared code available as a corpus (Plotts and Risdal, 2023). In the corpus, code has several formats and languages, either unstructured files containing the entire code with comments, or a notebook-style format in which code, comments and other exhibits are logically separated into cell blocks[13], which makes an additional processing step necessary. Before I transform each instance of code into an embedding, I remove all code from the data that has multiple authors, that is an edited version of code written by another user, has less than 10 lines, or is a duplicate. When the original author of a duplicate code file is not credited by the author of the copy, I manually search for the original author, assign the code file to only them and drop all duplicates. For each code

[11] Taking the centroid as a base is necessary for an intuitive reason: Imagine two scientists starting at the same point in the knowledge space. Each scientist takes ten steps of equal length in this knowledge space. However, one scientist walks in a circle around the starting point, while the other walks ten steps away from the starting point. We would probably all consider the first scientist more specialized than the second.

[12] Although being a competition grandmaster is considered the more prestigious title, code grandmasters also provide this information on their LinkedIn, and are hired by leading AI companies, see e.g. h2o.ai (2025).

[13] See Appendix Figure A1 for a visualization

file, I only keep the most current version. I also drop all code written by users who have published only one code file on the platform to ensure I have enough observations per user to reliably classify specialization.

I extract code embeddings from CodeT5+ (Wang et al., 2023). CodeT5+ is an open source large language model trained for several tasks, such as suggesting code to solve a specific problem, explain code, or complete code. CodeT5+ is particularly well-suited for my study because it is trained on both code and natural language comments, as well as entire programs rather than isolated code snippets. These features make it more effective for capturing broader programming contexts. Although I can technically separate code and comment data in some notebooks, it is non-trivial to remove all natural language from code files.[14] Including comments in the measure of specialization is also beneficial because it adds information on the task at hand. While CodeT5+ is able to process relatively large sequences of code and comments, many Kaggle code files exceed CodeT5+'s maximum sequence length[15]. To handle longer files, I split the code file into logical units, equivalent to paragraphs (see Figure A1 for an illustration), which I transform into embeddings and then aggregate.[16]

Once I have obtained embeddings for each code file, I construct the centroid (mean) embedding for each user, calculate the pairwise cosine similarity of each individual code embedding to the centroid, and take the average. This yields a continuous measure of specialization for 65.211 users, see Figure 1, Panel (a). A value closer to one indicates that all code written by a user is very similar, i.e. a higher level of specialization. However, mechanically, the distribution of the specialization index is shifted to the right when it is constructed from fewer code files, see Figure 1, Panel (b). To not confound the number of code files written with specialization, I adjust the index by residualizing it to the number of kernels, as shown in Panels (c) and (d). I then define *Generalists* as users with below median levels of specialization, and *Specialists* as users with above median levels of specialization.

3.4 Variable Definitions

3.4.1 Key Concepts

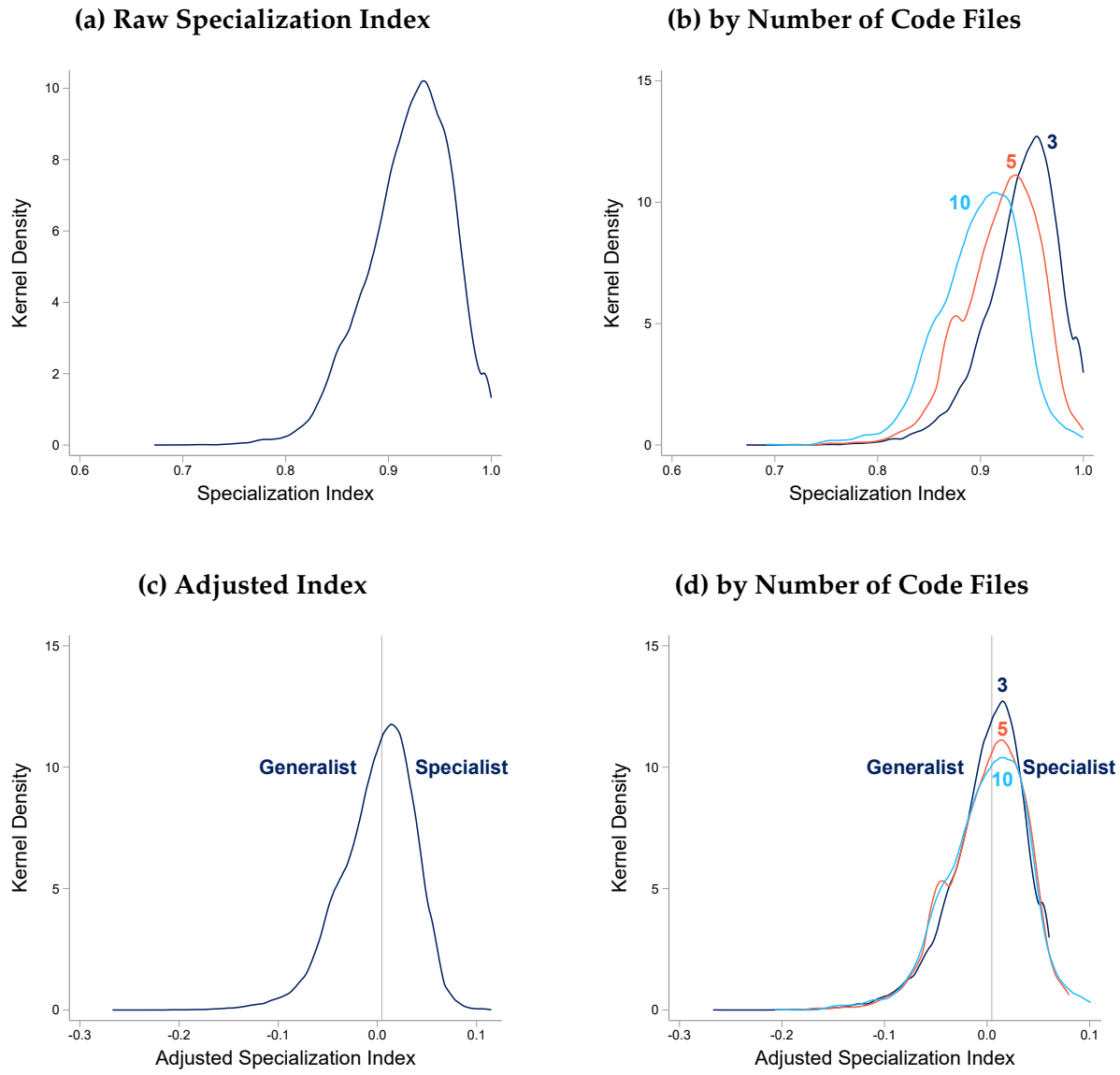
Quality I measure solution quality in three different ways, natural to the setting: *Rank*, *Medal Win*, and *Top 3*. These three metrics roughly target three different objects: average quality, high quality, and top quality. First, all competitions on Kaggle specify an objective, numerical target to assess a submission's quality, for example minimizing root mean square error. All submissions are then ranked according to this metric. This allows

[14]For example, code might be commented out, or a comment might follow a line of code. Additionally, comments are marked differently in different coding languages.

[15]768 tokens ("words" of code) to be exact.

[16]If these sub-units are still too large to be processed by Code T5+, I split them further into lines.

Figure 1: Specialization Index



Notes: The figure illustrates the construction of the specialization index. Panel (a) displays the distribution of the raw specialization index, which is heavily skewed towards high similarities. Panel (b) illustrates how the number of code files affects the distribution of the raw specialization index with the examples of users who have written 3, 5 or 10 code files. More code files mechanically lead to lower specialization. Panel (c) shows the distribution of the number-of-code-files adjusted specialization, removing the impact of the number of code files on the distribution of specialization (Panel (d)).

me to measure quality differences in a continuous, granular manner. *Rank* transforms a team's absolute rank in a competition into their percentile rank within that competition. *Rank* is equal to 100 for the best, and 0 for the worst performing team. This allows me to compare teams' relative positions across competitions with different numbers of competitors. Second, a share of high-ranking teams receive virtual gold, silver or bronze medals. I generate a binary indicator to measure differences in the probability of

attaining a medal, *Medal Win*.[17] Third, I create an indicator for placing among the top three competitors. The top three competitors are typically awarded monetary prizes, required to license their solution code, and publish detailed, often paper-like solutions on Kaggle. We can think of winning a medal as providing a high-quality solution, and placing among the top three as an excellent solution.

Complexity I assess a competition's (problem's) complexity by measuring the ex-post likelihood of overfitting. To uncover complex relationships, a machine learning model needs to be able to incorporate this complexity. The higher a model's capacity to reflect complex relationships, the higher the complexity of the model itself. However, model complexity comes at the cost of a higher risk of overfitting (Mohri et al., 2018). The algorithm might pick up spurious correlations in the test data, but fail to capture true underlying relationships (Mullainathan and Spiess, 2017). A competition is doubly complex if it is hard to spot this risk of overfitting. I measure a competition's complexity as the aggregate realized amount of overfitting. To measure overfitting risk, I use "Shake-Up", a concept coined by Kaggle users to assess overfitting in competitions (Trotman, 2019), which has been validated as a measure of overfitting in machine learning (Roelofs et al., 2019):

$$\text{Complexity} = \frac{1}{\text{Number of Competitors}} \sum \frac{|\text{Private Rank} - \text{Public Rank}|}{\text{Number of Competitors}} \tag{7}$$

In other words, the average change in relative performance of the algorithm from public to private test set. Large changes are mostly driven by many competitors overfitting on the training data and achieving high scores on the public dataset, but dropping in ranks once the algorithm is evaluated against the test data. Those that did not overfit will subsequently shoot up in ranks, even if the score of their algorithm remains stable – as it should when properly designed.

3.4.2 Additional Variables

User Characteristics To control for factors that may influence solution quality other than specialization, I define a range of time-varying user-level covariates. To proxy a user's current ability, I generate two measures of performance: *Previous Rank* and *Previous Medal Wins*. *Previous Rank* is the average percentile rank achieved by a user in prior competitions in which they participated as a solo competitor. *Previous Medal Wins* is the share of previous competitions in which a user won a medal among all competitions in which they participated as a solo competitor. I also measure *Competition Experience*, the number of competitions a user competed in, whether alone or in a team,

[17] Kaggle users refer to this as placing "in the metal". The number of medals awarded in a competition is determined by the number of competitors. For details see Kaggle.com (2025).

before entering a given competition. I collect an additional set of user demographics from users' profiles, on which users can report their location, occupation and preferred pronouns, as well as write short biographical notes (*user bio*).[18] I use pronouns and user names to classify a user's gender in a multi-step procedure, for details, see Appendix B.2. I hand-code occupation strings and occupational information reported in the user's bio into 17 occupational categories. Appendix Table A2 provides details on chosen occupational categories, and included occupations.

Code Characteristics Since specialization is measured from code, I include several characteristics of code files: The number of lines per code file, the number of lines of comment, and the votes received by a code file as a measure of code quality.[19] I aggregate code characteristics at the author level.

Team Characteristics Team characteristics are simply all team members' user-level characteristics aggregated to the team level as well as team size. This mirrors the identification strategy in Weidmann and Deming (2021), where individual performance at the same task of all team members is aggregated to isolate the effect of team members' ability from team effects.

Submission Characteristics During a competition, participants can submit preliminary solutions to the competition, which are then evaluated against the public test set. For all submissions, I observe which user made the submission, submission dates, algorithm performance, and which submissions were chosen for the final submission. I use submission data in two ways: First, since I am able to construct counterfactual ranks for each preliminary submission – information that is not available to competition participants – I investigate the change in submission quality in response to teaming up in an event study approach similar to Lemus and Marshall (2024). Second, I use the amount and timing of submissions as a measure for motivation and effort to rule out alternative mechanisms.

3.5 Sample Construction and Summary Statistics

I start with the 7.19 million team member observations who participate in any competition. Note that teams are defined at the competition level, i.e. team member refers

[18] Many users report CV-like information in their bio, some share their motivation to participate in Kaggle, others only include pictures or emojis.

[19] If another user finds a code file useful, well-written, or creative, they can vote for the code file. Since Kaggle status groups are based on votes, voting behavior is strictly monitored. Self-promotion or vote collusion can result in bans (see Kaggle.com (2024b)), so votes are a relatively noise-free signal of code quality.

to a user on a specific team in a specific competition. I then drop teams who participate in competitions in which performance is not incentivized, as well as recruitment competitions, in which there naturally is no teamwork. This leaves me with 4.35 million observations. I then drop teams that sign up for a competition but never make a submission, which is the majority of teams (3.83 million). A possible explanation for this strikingly large number is that, to access competition data, Kaggle users have to formally sign up for a competition, even if they never intended to compete.[20] I also drop all benchmark submissions by competition hosts and the Kaggle team. At the end, 518,240 team member observations remain. In a final step, I drop all teams for which I cannot measure specialization for one or more team members, to ensure I accurately measure team types. This leaves me with 76,776 user-competition observations, or 74,433 teams, from 357 distinct competitions. These are comprised of 13,646 distinct users. Table 1 presents summary statistics for competitions.

Table 1: Summary Statistics on Competitions

Variable	Mean	SD	Observations
Prizes	53152.26	140325.19	353
Competitors	1450.75	1493.91	357
Competitors with Specialization	215.06	223.78	357
Duration	86.49	48.85	357
Teams	0.02	0.02	357
Research	0.34		357
Post ChatGPT	0.11		357

Notes: The table reports summary statistics for competitions. Data come from Meta Kaggle (Risdal and Bozsolik, 2022). Prizes are total competition prize pools in U.S. dollars. Competitors reflect the total number of competitors submitting to a competition, and Competitors with Specialization are the subset of competitors for which I can measure specialization. Duration is the duration of a competition in days. Teams indicates what share of competitors enters the competition as a team. Research is an indicator whether the competition is a research competition (as defined by Kaggle). Post ChatGPT is an indicator whether a competition ended after the introduction of ChatGPT.

Table 2 presents summary statistics for users, split into generalists and specialists. Kaggle users are mostly male, have been active on the platform for about 2.5 years, and enter many competitions, but compete mostly on their own. Observing many solo competitions per user allows me to measure individual ability independent of team performance. Generalists have been active on the platform slightly longer than specialists, are more likely to report their location and to be from the USA, to report their occupation and to identify as data scientists, write shorter code, are less likely to collaborate, and to do so in mixed teams. The empirical analysis take these underlying differences into account.

[20]Of these, 49% are teams that sign up for a competition after the competition deadline, which is possible since Kaggle competitions stay active after they ended to enable other users to access the competition data.

Table 2: Summary Statistics on Users

Variable	All Users	Generalists	Specialists	p-value Gen = Spec
No. Users	13646	7077	6569	
Female	0.068	0.065	0.071	0.18
User Platform Age	940.001	1006.982	868.054	0.00
Reports Location	0.856	0.876	0.835	0.00
India	0.220	0.210	0.231	0.00
USA	0.134	0.161	0.105	0.00
Reports Occupation	0.680	0.692	0.667	0.00
Data Scientist	0.164	0.175	0.151	0.00
Code Files	10.628	10.318	10.962	0.25
Lines	219.740	188.315	253.595	0.00
Votes	6.308	6.572	6.025	0.15
Competitions	5.626	6.143	5.070	0.00
Collaborations	0.048	0.046	0.052	0.05
Mixed Team	0.571	0.547	0.600	0.02
Team Lead	0.423	0.433	0.412	0.31

Notes: The table reports summary statistics for users. Data come from Meta Kaggle (Risdal and Bozsolik, 2022) and own hand-coding. Female is an indicator whether the user is female, user platform age is the time (in days) a user has been active on Kaggle. Reports location is an indicator whether the user reports a location in their profile, with India (dummy) and USA (dummy) being the most two most common locations. Reports Occupation is an indicator whether the user includes information on their occupation in their profile, with Data Scientist (dummy) being the most frequent occupation. Code Files is the number of code files solo-written by a user, lines the average length of a code file, and votes the average number of votes received by a code file. Competitions is the number of competitions a user in the data enters, whether as a solo competitor or in a team. Collaborations indicates the share of competitions a user enters as a team. Mixed Team measures the share of a user's teams that are mixed teams, and team lead is the share of teams in which a user is the team lead.

Table 3 presents summary statistics for stable teams. I define a team composed of the exact same users across competitions as a stable team. Note that I also consider a user's observations from solo competitions as stemming from a stable "team". Overall, I observe 327 generalist teams who compete 1.47 times on average, 704 mixed teams competing 1.31 times on average and 211 specialist teams competing on average 1.46 times. Teams on Kaggle are small. While most competition entrants are solo competitors, teams are mostly comprised of two people. Users who compete in teams have been active on the platform for a longer time, and users tend to enter many competitions on their own before ever competing as a team. Teams also make more submissions than solo competitors and outperform solo competitors on all dimensions. Among teams, generalist teams achieve the highest ranks, win most medals, and place among the top 3 most often. Reflecting the higher experience of generalists, generalist teams also are comprised of the most experienced and able users. Isolating the effect of the ingredients – team members – and the recipe – team type – on algorithm quality is the focus of the next section.

Table 3: Summary Statistics on Stable Teams

Variable	All Teams	Solo Competitor	Generalist Team	Mixed Team	Specialist Team
No. Team-Competition Observations	74433	72721	482	922	308
No. Stable Teams	14568	13326	327	704	211
No. Competitions as Stable Team	5.11	5.46	1.47	1.31	1.46
Rank	46.45	43.75	77.21	76.83	67.86
Medal Win	0.11	0.07	0.49	0.48	0.34
Top 3	0.00	0.00	0.06	0.04	0.02
Team Size	1.12	1.00	2.21	2.63	2.21
Female Share	0.06	0.07	0.05	0.05	0.06
Data Scientist	0.17	0.16	0.28	0.24	0.22
At 1st Competition as Stable Team					
User Platform Age	504.88	467.77	995.60	902.37	735.08
Experience	1.23	0.19	13.21	13.19	8.44
Previous Individual Rank	61.20		62.58	61.73	56.69
Previous Medal Wins	0.22		0.24	0.22	0.20
Total Submissions	18.46	12.68	77.73	85.60	67.54
High Complexity Competition	0.29	0.30	0.28	0.28	0.29
Post ChatGPT Competition	0.17	0.18	0.11	0.16	0.17

Notes: The table reports summary statistics for stable teams, i.e. teams with the exact same members across competitions. One user can be a member of multiple stable teams, and a solo competitor is also considered a stable team whenever they compete alone. Data come from Meta Kaggle (Risdal and Bozsolik, 2022). No. Competitions as Stable Team measures how frequently I observe a team composed of the exact same users across competitions. Rank is the percentile rank achieved by a team in a competition, Medal Win an indicator for whether a team won a medal, and Top 3 an indicator for whether a team placed among the top 3 in a competition. Team Size measures the size of a team, female share is the share of team members who are female, and Data Scientist the share of team members whose occupation is Data Scientist. User Platform Age is the team member's average time active on the platform when they first enter a competition (for solo competitors) or first join a team. Experience measures the team members's average number of previous competitions. For solo competitors, this is the number of competitions participated in as a team before the first solo competition, for teams this is the average number of competitions participated in as a solo competitor or in a team before joining the stable team. By construction, Previous Individual Rank and Previous Medal Win are not defined for a solo competitor's first solo competition. For teams, these reflect the average percentile rank achieved by team members' in solo competitions, and the team-member-average of Previous Medal Wins, i.e. the share of previous solo competitions in which a team member has won a medal. Total Submissions is the count of submissions made to a competition by a team. High Complexity Competitions is an indicator that is one whenever a team enters a high complexity competition (for details, see Section 3.4.1). Post ChatGPT is an indicator that is one whenever a team enters a competition that ended place after the introduction of ChatGPT.

4 The Impact of Team Type on Solution Quality

4.1 Empirical Strategy

The quality of a team's algorithm naturally depends both on the characteristics of individual team members as well as any synergies created by the team. Therefore, it is crucial to isolate the effects of individual team members from the overall team effect. For instance, if generalists tend to be more experienced, the superior performance of generalist teams might reflect experience rather than any inherent benefit of the team structure itself. I employ several strategies to disentangle the effect of individual team members from the composition effect. This is econometrically challenging, as outcomes are realized at the team level, but potential confounds arise at the individual level

(Constantine and Correia, 2021; Bonhomme, 2022).

A particular concern is the ability of individual team members, which I first account for explicitly by constructing a measure of individual ability from team members' past solo competition performance. Adhvaryu et al. (2023) use a similar approach to isolate the effect of individual scientists on co-authored papers or joint patents. To address unobservable team member heterogeneity, I include a specification with team member fixed effects. This is similar in spirit to the AKM literature (Abowd et al., 1999), where worker fixed effects are identified from firm switchers. I identify scientist fixed effects from team switchers as well as from switches from solo work to teamwork. Finally, I utilize the fact that I can observe individual contributions to team output in the form of submissions. For a small subset of teams, I can track individual team member's submissions over the course of the competition and assess the impact of joining a team of a specific type on submission quality, a strategy similar to the one implemented by Lemus and Marshall (2024).

4.2 Results Isolating Team Member Ability

I first estimate a regression of solution quality on team type to test Prediction 1:

$$\text{Y}_{tc} = \alpha + \tau_1 \text{Generalist Team}_t + \tau_2 \text{Mixed Team}_t + \tau_3 \text{Specialist Team}_t + X'_{tc}\beta + \gamma_c + \epsilon_{tc} \quad (8)$$

Where Y_{tc} is a team t's solution quality (Rank, Medal Win, Top 3) in competition c, α is the intercept, equivalent to the average performance of a solo competitor. Generalist Team$_t$ is an indicator that equals one if team t is comprised of only generalists, Mixed Team$_t$ is an indicator that equals one if team t consists of both generalists and specialists and Specialist Team$_t$ is an indicator that equals one if team t consists of only specialists. The coefficients τ_1, τ_2 and τ_3 measure the difference in solution quality of each type of team relative to the solution quality of a solo competitor. X_{tc} is a vector of time-varying team-level controls including average team member experience, proxies for average team member abilities, team size, and demographic controls, including occupation shares for all occupations reported by at least 100 users, country shares for all countries with at least 100 users, share female and the average time team members have been active on the platform. γ_c are competition fixed effects.

Table 4 displays results for Equation (8) on the percentile rank achieved by each team's solution. Column 1 shows gross quality differences: all teams produce higher quality than individual competitors, with generalist teams placing 21 percentile ranks higher than solo competitors, mixed teams placing roughly 18 percentile ranks above solo competitors, and specialist teams placing about 12.5 percentile ranks above solo competitors. These differences are significant at the 1% level. In addition, the quality

differences between team types are also highly significant, with generalists achieving higher quality than mixed and specialist teams.

Table 4: Team Type and Solution Quality: Rank

Dependent Variable:	*Rank*				
	(1)	(2)	(3)	(4)	(5)
Generalist Team	21.19***	18.45***	18.77***	18.66***	20.03***
	(1.44)	(1.11)	(1.08)	(1.07)	(1.07)
Mixed Team	17.85***	16.16***	16.46***	16.20***	16.65***
	(1.12)	(0.94)	(0.93)	(0.92)	(0.90)
Specialist Team	12.50***	15.73***	16.26***	15.92***	15.80***
	(2.09)	(1.99)	(2.02)	(2.04)	(1.99)
Competition Experience		0.03***	0.05***	0.05***	0.08***
		(0.01)	(0.01)	(0.01)	(0.01)
Previous Rank		0.52***	0.50***	0.49***	0.49***
		(0.01)	(0.01)	(0.01)	(0.01)
3-Person Team	8.11***	6.71***	6.38***	6.11***	6.07***
	(2.00)	(1.63)	(1.62)	(1.60)	(1.58)
4-Person Team	17.18***	12.98***	12.49***	12.30***	12.47***
	(2.26)	(2.34)	(2.35)	(2.31)	(2.19)
5-Person Team	17.15***	13.74***	13.58***	13.48***	14.69***
	(2.13)	(1.69)	(1.69)	(1.67)	(1.74)
R^2	0.01	0.15	0.16	0.16	0.20
Observations	74433	60897	60204	60090	60090
Demographic Controls			Yes	Yes	Yes
Code Quality Controls				Yes	Yes
Competition FEs					Yes
p-value (Generalist Team = Mixed Team)	0.04	0.09	0.08	0.06	0.01
p-value (Generalist Team = Specialist Team)	0.00	0.22	0.26	0.22	0.06
p-value (Specialist Team = Mixed Team)	0.02	0.84	0.92	0.89	0.68

Notes: The table reports the estimates of equation (8). The dependent variable measures the percentile rank of a team's solution in competition c. The main explanatory variables are indicators for the team's type: Generalist Team (a team of only generalists), Mixed Team (a team of both generalists and specialists) and Specialist Team (a team of only specialists). Solo competitors are the omitted category. Competition experience measures the average of the number of competitions each team member participated in before competition c. Previous Rank measures the average of the percentile rank achieved by each team member when competing alone in a previous competition. Demographic controls include share female, average user platform age, country shares for all countries with at least 100 users, and occupation shares for all occupations reported by at least 100 users (c.f. Table A2). Code quality controls include the average of each team member's lines of code, lines of comment, number of code files, and votes received by code files. Standard errors are clustered at the stable team level. Significance levels: $***$ p<0.01, $**$ p<0.05, and $*$ p<0.1.

As previously discussed, quality differences might simply be a reflection of pre-existing ability differences and positive selection into collaborating rather than any effect of teamwork or the specific type of team. To account for these differences, column 2 adds controls for team members' previous competition experience and previous percentile ranks achieved by team members when they were competing on their own. This reduces sample size by circa 14,000 observations, since I can only observe experience and previous ranks for teams that have at least one competitor that has competed once before. While the effect of competition experience is significant but small, the effect of previous rank is large in size: Placing 1 percentile higher in previous competitions

is associated with placing 0.5 percentiles higher in the current competition's ranking on average. Again, teams consistently achieve higher quality than solo competitors. The differences between team types however are smaller, reducing performance gaps to 3 percentile ranks between generalist and specialist teams, and 2 percentile ranks between generalist and mixed teams. This reflects the fact that generalist teams seem to be positively selected (c.f. Table 2 and Table 3). In column 3, I add the full set of demographic controls with minimal effects on coefficient sizes. Since specialization is measured from code, my index might also capture subtle differences in code quality, which could be correlated with team performance. Column 4 thus adds code quality controls, such as the length of code, amount of comments, the number of code files and votes received by code files as a proxy for code quality. The results are quantitatively similar to columns 2 and 3.

In the model, a crucial determinant of quality gaps between generalist and specialist teams is a problem's type. Thus, I add competition fixed effects in column 5. When accounting for problem type, the quality difference between generalist teams and specialist teams are large and significant: generalist teams place around 4 percentile ranks higher than specialist teams, this difference is significant at the 10% level. Generalist teams also place about 3.5 percentile ranks higher than mixed teams (significant at the 1% level).

Are generalist teams also more likely to provide high quality solutions? Table 5 displays results for the probability of winning a medal. As in Table 4, column 1, all team types are significantly more likely to win a medal than solo competitors. Generalist teams are 31 percentage points more likely to win a medal, while mixed and specialist teams are 25 and 15 percentage points more likely. These coefficients are all significantly different from each other (at the 5 or 1 % level). When controlling for experience and previous medal wins in column 2, the coefficient for specialist teams increases to 20 percentage points. However, it remains significantly lower than the effect for generalist teams at 29 percentage points. Competitors' medal track records are highly predictive of winning a medal: If all team members have won a medal before, their probability of winning a medal as a team is 48 percentage points higher than that of a team in which no member has ever won a medal. Controlling for demographics (column 3), code quality (column 4) and including competition fixed effects (column 5) barely affects coefficient sizes. When accounting for all covariates and fixed effects, Generalist teams are 8 percentage points more likely to win a medal than specialist teams.

Turning now to the probability of providing an excellent solution, Table 6 reports results for Equation (8) on the likelihood of a team's solution being among the top three solutions in a competition. Column 1 displays gross quality differences. While generalist teams are 5 percentage points more likely to find a solution that is among the top three, and mixed teams are 1 percentage point more likely, specialist teams

Table 5: Team Type and Solution Quality: Medal Win

Dependent Variable:	*Medal Win*				
	(1)	(2)	(3)	(4)	(5)
Generalist Team	0.31***	0.29***	0.30***	0.30***	0.30***
	(0.03)	(0.03)	(0.03)	(0.03)	(0.03)
Mixed Team	0.25***	0.24***	0.25***	0.25***	0.25***
	(0.02)	(0.02)	(0.02)	(0.02)	(0.02)
Specialist Team	0.15***	0.20***	0.22***	0.21***	0.22***
	(0.04)	(0.03)	(0.03)	(0.03)	(0.03)
Competition Experience		-0.00**	-0.00***	-0.00***	-0.00***
		(0.00)	(0.00)	(0.00)	(0.00)
Previous Medal Wins		0.48***	0.45***	0.43***	0.39***
		(0.01)	(0.01)	(0.01)	(0.01)
3-Person Team	0.19***	0.18***	0.18***	0.17***	0.18***
	(0.04)	(0.04)	(0.04)	(0.04)	(0.04)
4-Person Team	0.32***	0.29***	0.28***	0.28***	0.30***
	(0.06)	(0.06)	(0.06)	(0.06)	(0.06)
5-Person Team	0.29***	0.28***	0.27***	0.27***	0.29***
	(0.06)	(0.06)	(0.06)	(0.06)	(0.06)
R^2	0.02	0.11	0.12	0.12	0.15
Observations	74433	60897	60204	60090	60090
Demographic Controls			Yes	Yes	Yes
Code Quality Controls				Yes	Yes
Competition FEs					Yes
p-value (Generalist Team = Mixed Team)	0.07	0.10	0.12	0.11	0.08
p-value (Generalist Team = Specialist Team)	0.00	0.03	0.05	0.04	0.04
p-value (Specialist Team = Mixed Team)	0.03	0.28	0.37	0.36	0.37

Notes: The table reports the estimates of equation (8). The dependent variable is an indicator whether a team's solution has won a medal in competition c. The main explanatory variables are indicators for the team's type: Generalist Team (a team of only generalists), Mixed Team (a team of both generalists and specialists) and Specialist Team (a team of only specialists). Solo competitors are the omitted category. Competition experience measures the average of the number of competitions each team member participated in before competition c. Previous Medal Wins measures the average of the share of previous competitions for each team member in which a team member won a medal when competing alone. Demographic controls include share female, average user platform age, country shares for all countries with at least 100 users, and occupation shares for all occupations reported by at least 100 users (c.f. Table A2). Code quality controls include the average of each team member's lines of code, lines of comment, number of code files, and votes received by code files. Standard errors are clustered at the stable team level. Significance levels: $***$ p<0.01, $**$ p<0.05, and $*$ p<0.1.

are indistinguishable from solo competitors. Controlling for experience and previous performance (column 2), demographics (column 3), code quality (column 4) and including competition fixed effects (column 5) has minimal effect on coefficient sizes and significance. In fact, generalist and mixed teams' advantage over solo competitors increases by one percentage point compared to column 1, while specialist teams are not differentially likely to place among the top three than solo competitors. Generalist teams are significantly more likely to place among the top three than mixed teams and specialist teams throughout all specifications. These differences are significant at the 1% level. Taken together, the patterns documented in Tables 4-6 provide strong support for Prediction 1.

Table 6: Team Type and Solution Quality: Top 3

Dependent Variable:	*Top 3*				
	(1)	(2)	(3)	(4)	(5)
Generalist Team	0.05***	0.06***	0.06***	0.06***	0.06***
	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)
Mixed Team	0.01***	0.02**	0.02***	0.02**	0.02**
	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)
Specialist Team	0.00	0.01	0.01	0.01	0.01
	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)
Competition Experience		-0.00*	-0.00***	-0.00***	-0.00***
		(0.00)	(0.00)	(0.00)	(0.00)
Previous Rank		0.00***	0.00***	0.00***	0.00***
		(0.00)	(0.00)	(0.00)	(0.00)
3-Person Team	0.04**	0.04**	0.04**	0.04**	0.04**
	(0.02)	(0.02)	(0.02)	(0.02)	(0.02)
4-Person Team	0.05*	0.05	0.05	0.05	0.05
	(0.03)	(0.03)	(0.03)	(0.03)	(0.03)
5-Person Team	-0.00	-0.01	-0.01	-0.01	-0.01
	(0.02)	(0.02)	(0.02)	(0.02)	(0.02)
R^2	0.01	0.02	0.02	0.03	0.05
Observations	74433	60897	60204	60090	60090
Demographic Controls			Yes	Yes	Yes
Code Quality Controls				Yes	Yes
Competition FEs					Yes
p-value (Generalist Team = Mixed Team)	0.00	0.00	0.00	0.00	0.00
p-value (Generalist Team = Specialist Team)	0.00	0.00	0.00	0.00	0.00
p-value (Specialist Team = Mixed Team)	0.24	0.50	0.54	0.52	0.52

Notes: The table reports the estimates of equation (8). The dependent variable is an indicator whether a team's solution was among the top three solutions in competition c. The main explanatory variables are indicators for the team's type: Generalist Team (a team of only generalists), Mixed Team (a team of both generalists and specialists) and Specialist Team (a team of only specialists). Solo competitors are the omitted category. Competition experience measures the average of the number of competitions each team member participated in before competition c. Previous Rank measures the average of the percentile ranks achieved by each team member when competing alone in a previous competition. Demographic controls include share female, average user platform age, country shares for all countries with at least 100 users, and occupation shares for all occupations reported by at least 100 users (c.f. Table A2). Code quality controls include the average of each team member's lines of code, lines of comment, number of code files, and votes received by code files. Standard errors are clustered at the stable team level. Significance levels: $***$ p<0.01, $**$ p<0.05, and $*$ p<0.1.

4.3 Results Isolating Team Member Unobservables

A potential concern is that previous solution quality achieved by team members as solo competitors and code quality do not sufficiently capture individual ability. Rather, there may be time-invariant unobservable user characteristics that influence team solution quality. To address this concern, I estimate the following regression:

$$\mathrm{Y}_{tc} = \alpha + \tau_1 \text{Generalist Team}_t + \tau_2 \text{Mixed Team}_t + \tau_3 \text{Specialist Team}_t + \gamma_c + \sum_{i \in G(t)} \upsilon_i + \epsilon_{tc} \tag{9}$$

Y_{tc} and Generalist Team$_t$, Mixed Team$_t$ and Specialist Team$_t$ are defined as above. $\sum_{i \in G(t)} \upsilon_i$ is a function which aggregates all $G(t)$ user fixed effects υ_i to the team level, γ_c are competition fixed effects, and ϵ_{tc} are team-competition specific shocks. To efficiently estimate individual fixed effects with group level outcomes, I use the estimator introduced with the reghdfe package (Constantine and Correia, 2021), which aggregates individual fixed effects at the group level and provides accurate inference.

Table 7: Team Type and Solution Quality: Team Member Fixed Effects

Dependent Variable:	*Rank*		*Medal Win*		*Top 3*	
	(1)	(2)	(3)	(4)	(5)	(6)
Generalist Team	17.98***	19.25***	0.28***	0.28***	0.05***	0.05***
	(1.00)	(1.05)	(0.02)	(0.02)	(0.01)	(0.01)
Mixed Team	17.72***	18.17***	0.26***	0.27***	0.02***	0.02***
	(0.90)	(0.88)	(0.02)	(0.02)	(0.01)	(0.01)
Specialist Team	16.23***	15.82***	0.22***	0.23***	0.02	0.02
	(1.56)	(1.51)	(0.03)	(0.03)	(0.01)	(0.01)
3-Person Team	5.05***	5.14***	0.16***	0.16***	0.04**	0.04**
	(1.32)	(1.31)	(0.03)	(0.03)	(0.02)	(0.02)
4-Person Team	9.43***	8.99***	0.26***	0.27***	0.04	0.04
	(1.89)	(1.93)	(0.05)	(0.05)	(0.03)	(0.03)
5-Person Team	10.06***	10.97***	0.28***	0.29***	-0.00	-0.00
	(1.74)	(1.87)	(0.06)	(0.06)	(0.02)	(0.02)
R^2	0.36	0.40	0.26	0.29	0.14	0.16
Observations	69381	69372	69381	69372	69381	69372
User FEs	Yes	Yes	Yes	Yes	Yes	Yes
Competition FEs		Yes		Yes		Yes
p-value (Generalist Team = Mixed Team)	0.84	0.39	0.56	0.51	0.05	0.04
p-value (Generalist Team = Specialist Team)	0.34	0.06	0.17	0.21	0.04	0.04
p-value (Specialist Team = Mixed Team)	0.40	0.17	0.30	0.40	0.69	0.70

Notes: The table reports the estimates of equation (9). The dependent variable is the percentile rank of a team's solution quality, an indicator whether a team's solution has won a medal or an indicator whether a team's solution was among the top three solutions in competition c. The main explanatory variables are indicators for the team's type: Generalist Team (a team of only generalists), Mixed Team (a team of both generalists and specialists) and Specialist Team (a team of only specialists). Significance levels: $***$ p<0.01, $**$ p<0.05, and $*$ p<0.1.

Table 7 displays results for Equation (9). Columns 1 and 2 report coefficients for the team's percentile ranks, once with (1) and once without (2) including competition fixed effects. The coefficients on the different team types are close to the coefficients reported in Table 4. This pattern is repeated in columns 3 and 4, which report results for the probability of a team's solution winning a medal, and columns 5 and 6 which report results for the probability of a team's solution placing among the top three solutions in a competition. In all specifications, generalist teams provide higher quality solutions than specialist teams. The difference between generalist and specialist teams is significant at the 10% level for Rank, and at the 5% level for Top 3, but not for Medal Win, despite similar coefficient sizes as in Table 5. These results indicate that quality differences

between generalist and specialist teams are not driven by time-invariant unobservable user characteristics.

4.4 Results Isolating the Effect of Joining a Team

In the previous sections, I have shown that generalist teams achieve higher quality solutions than specialist teams, and that this difference is driven neither by time-varying observables nor time-constant unobservables. Both empirical approaches relied exclusively on between-competition variation. Still, one might be concerned that performance in different competitions insufficiently captures task-specific ability. Using performance in other competitions as a proxy for ability in a given competition implicitly assumes that computer scientists only choose to enter competitions that match their skill sets. If, for instance, computer scientists are not fully aware of which skills a problem requires before entering the competition, generalists would in expectation work on problems that are within their skill set more often, since their skill set is broader. This is not an unlikely scenario. Furthermore, generalists' broader skills might enable them to better recognize which skills are required by a competition.

To address this concern, I make use of two unique features on Kaggle, preliminary submissions and in-competition team formation. As described in Section 3.4.2, competitors are able to make preliminary submissions to a competition to receive feedback on how the proposed algorithm performs on the public test set. While only performance on the public, but not the private (final) test set is known to competitors during the competition, Kaggle also calculates and records how an algorithm would have scored on the final test set. Using these scores and information on the scoring algorithm, I can construct counterfactual percentile ranks for each submission.

Users do not have to enter a competition as a team, but can also first enter a competition as a solo competitor, work independently, and later form a team. Importantly, once they join a team, which team member made a submission is still recorded by Kaggle. Similarly to Lemus and Marshall (2024), I track how submission quality changes before and after team formation. However, I deviate from their analysis in two core dimensions. First, my analysis is at the team *member* level rather than at the team level, allowing me to isolate individual contributions to team performance. Second, since the timing of team formation is likely endogenous, I restrict my analysis to teams forming at the exogenously set team formation deadline.[21] The team formation deadline is the modal date of team formation for all team types, see Figure A2. Since team members can join a team in a sequential way, "treatment" timing in larger teams is ambiguous, such that I additionally drop all teams with more than two members.

[21]This specification choice also allows me to circumvent issues created by staggered treatment as discussed in Sun and Abraham (2021), as all competitors in my analysis either form a team at exactly the same time or not at all.

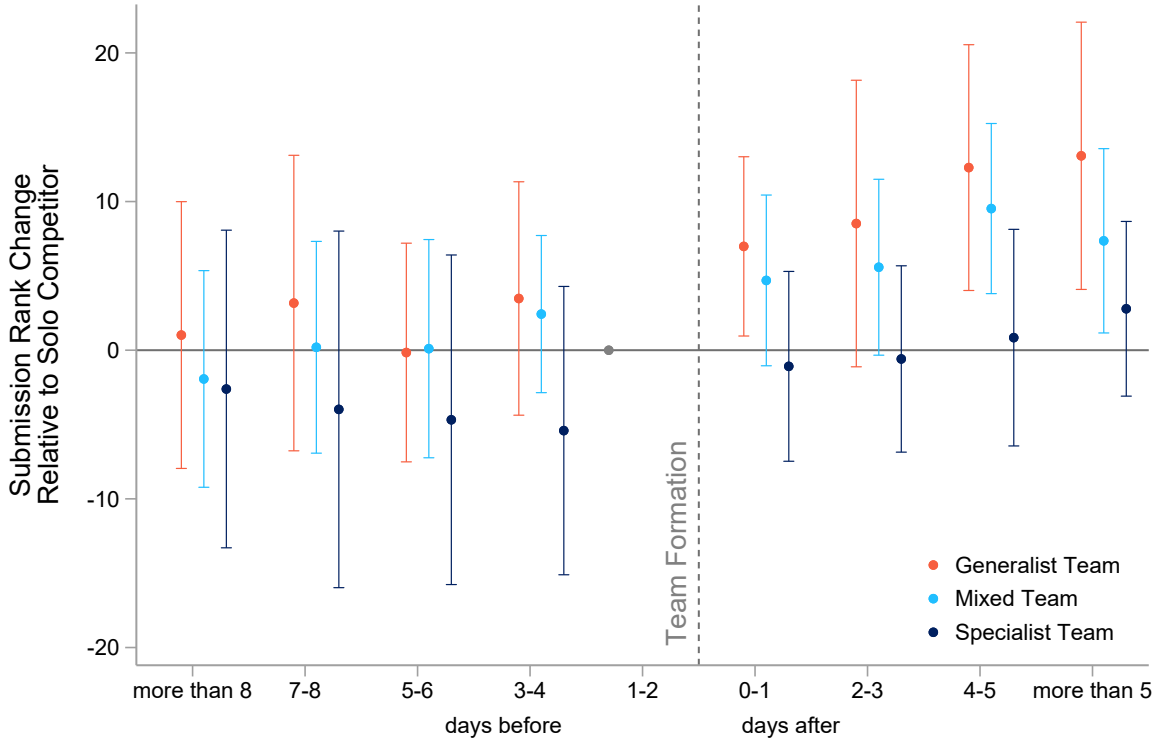
I then estimate the following event study:

$$\begin{aligned}\text{Submission Rank}_{ict} =&\text{Generalist Team}_{ic} \sum_{d\neq -1} \delta_d^G D_{T+d} + \text{Mixed Team}_{ic} \sum_{d\neq -1} \delta_d^M D_{T+d} \\ &+\text{Specialist Team}_{ic} \sum_{d\neq -1} \delta_d^S D_{T+d} + \sum_{d\neq -1} \delta_d D_{T+d} + \upsilon_{ic} + \phi_t + \epsilon_{ict}\end{aligned} \tag{10}$$

where d indexes periods of two days relative to team formation, and D_{T+d} denote the event-study indicators for the periods leading up to and following team formation. Generalist Team$_{ic}$ is an indicator equal to one if user i joins a generalist team, Mixed Team$_{ic}$ is an indicator equal to one if user i joins a mixed team, and Specialist Team$_{ic}$ is an indicator equal to one if user i joins a specialist team team in competition c. Coefficients δ_d capture the effect of time relative to the team formation deadline at T for users who never form a team. δ_d^G measure the differential effect of time relative team formation for generalist teams, δ_d^M for mixed teams, and δ_d^S for specialist teams. υ_i are team member fixed effects, i.e. a user $\times$ competition fixed effect, and ϕ_t are competition day fixed effects.

Figure 2 plots results from Equation (10), and Table A3 displays coefficients. Since the team formation deadline most commonly coincides with the week before the competition ends, I include three lags of two days, and one lag pooling all submission dates more than five days after the team formation deadline. As Figure A3 illustrates, few teams make submissions after day seven. I also include two-day leads only for up to eight days before the team formation deadline and one lead capturing all earlier submission dates, since submissions are comparatively sparse in this time. The small number of daily submissions also informs my choice to bin submission dates into two-day windows, since estimation of the multitude of fixed effects is demanding. Before forming a team, submission ranks of eventual team members develop no differently than those of solo competitors, indicating no anticipation. After forming a team, the quality of submissions made by the members of generalist teams increases immediately by about 7 percentile ranks relative to users who do not form a team. This effect is significant at the 5% level, and significantly different from mixed and specialist teams, for whom I cannot detect any effects of team formation on users working in mixed or specialist teams. In the week after joining a team, submission quality by team members in generalist teams steadily increases relative to solo competitors, with members of mixed teams catching up slightly. The quality of submissions by members in specialist teams does not change significantly in response to team formation relative to those by solo competitors. After five days, submissions by team members of generalist teams place circa 10 percentile ranks higher than those of members of specialist teams.

Figure 2: Team Type and Solution Quality: Submission Event Study



Notes: The figure plots the estimated coefficients δ_d^G, δ_d^M and δ_d^S from Equation (10). Each coefficient is equivalent to a two-day period d before or after team formation at the exogenously set team formation deadline. The two-day period immediately before team formation is the omitted category. The coefficients indicate the percentile difference in submission ranks for each team type relative to the team formation deadline and to solo competitors who do not form a team at $d = 0$. The corresponding regression results are reported in Table A3.

4.5 Robustness

Robustness to Ability Proxies In Section 4.2 I use the average of past performance in all solo competitions to control for potential ability differences between team types. This could potentially over- or understate current ability if computer scientists' skills change dynamically, for instance because they learn from past competitions. To address this concern, I include only recent performance as an ability control in Table A4. Specifically, I average the percentile ranks achieved in the last (respectively second to last and third to last) solo competition for all team members. For medals, I take the share of team members who achieved a medal in their last (second/third to last) competition. We might also be worried that rather than the absolute past performance, trajectories matter for team performance. Table A5 controls for the team-level average change in percentile ranks between the first and second to last, second and third to last, and the average of first and second and second and third to last solo competitions. Coefficient sizes remain remarkably stable across specifications, as does significance. The only exception are

results for medal win with three performance lags (Table A4, Column 8) where I cannot detect significant differences between generalist and specialist teams anymore, likely due to substantially reduced sample size.

Robustness to Specialization Measure To ensure results are not driven by how I chose to classify specialists and generalists, I conduct several robustness checks. First, because I create generalist and specialist categories by binarizing the continuous specialization index, I conduct a sensitivity analysis to determine how different binarization cutoff points affect my results. Figure 3 presents results of this sensitivity analysis. It plots coefficients from Equation (8), including all controls and competition fixed effects as in Column (5) of Tables 4-6 with alternative team type classifications. Each category on the x-axis corresponds to a different cutoff point in the specialization index, e.g. 10% reflects team type classifications in which only individuals with very generalist skills, i.e. only those whose specialization is in the lowest decile of the specialization index were classified as generalists. Results at 50% correspond to the main results presented in this section. For all cutoff points below 50%, generalist teams achieve significantly higher ranks (Figure 3a), are more likely to win a medal (Figure 3b), and to place among the top three (Figure 3c) than specialist teams. In fact, the stricter I classify generalists, the larger the advantage of generalist teams over specialist and mixed teams, suggesting the results presented in this section might be a conservative estimate of the differences between generalists and specialists.

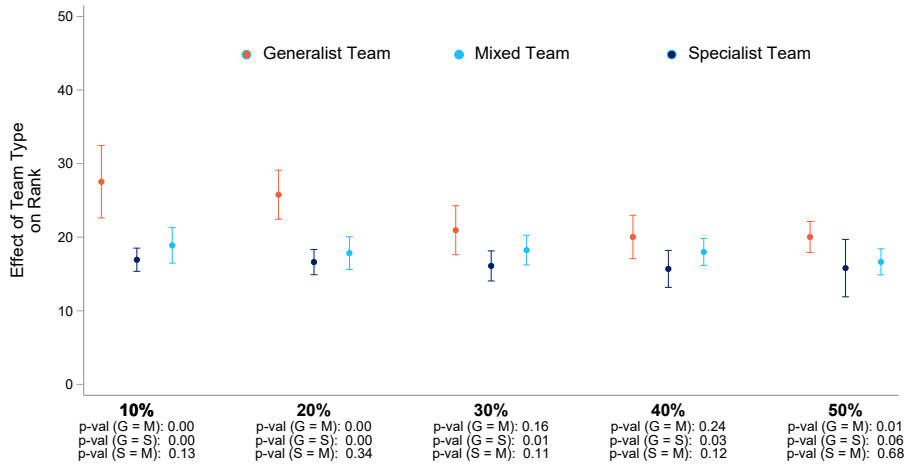
As a further robustness check, I re-calculate the specialization index excluding all code files published after November 30$^{\text{th}}$ 2022, when ChatGPT was introduced. This reduces the sample size by about 4000 teams. I also construct an alternative measure of specialization based on the python packages included in each code file. To use the functionality of a given python package, each python code file needs to explicitly import a package at the beginning of the file. Imported packages correspond to the goal of the code, for instance, package cv2 provides functionality for computer vision problems such as image processing, whereas package nltk is used for natural language processing. I construct *Python Package Diversity* (PPD) as:

$$PPD_i = \frac{\sum_{p=1}^{P_i} \text{Share of user i's Code Files using Package p}}{\text{Number of Distinct Packages used by user i}}$$

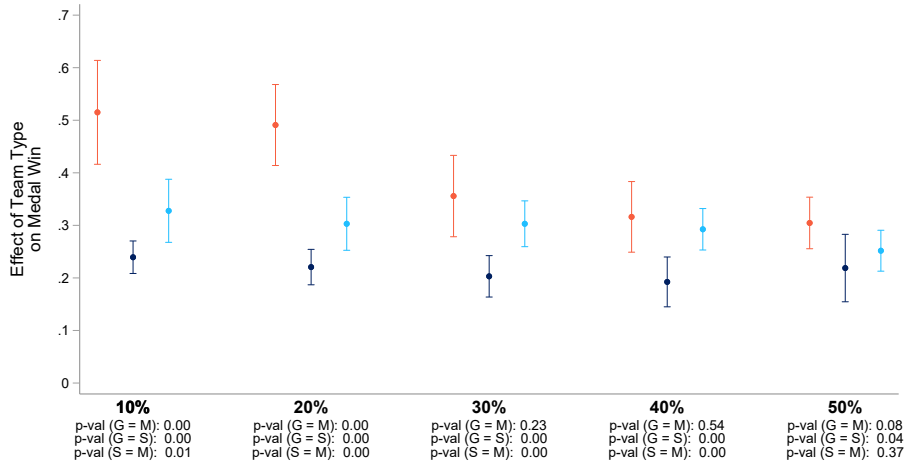
which is equal to one if all code files written by user i include the same combination of packages. I classify users with below median PPD as generalists. Table A6 presents results using these new specialization classifications. When excluding all code files written after the introduction of ChatGPT, I do not find significant percentile rank differences between generalist and specialist teams anymore. However, generalist teams are still significantly more likely than specialist teams to win a medal or place

Figure 3: Robustness of Main Result to Different Classification Cutoffs

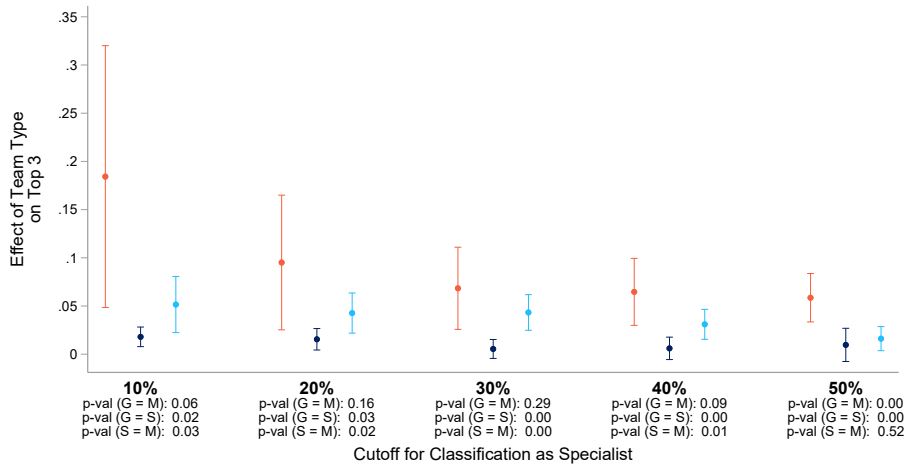
(a) Rank



(b) Medal Win



(c) Top 3



Notes: The figure plots the estimated coefficients τ_1, τ_2 and τ_3 from Equation (8) with all covariates and competition fixed effects when classifying generalists at different cutoffs of the specialization index. Panel (a) displays estimates for *Rank*, panel (b) for *Medal Win* and panel (c) for *Top 3*.

among the top three. When using PPD as a measure of specialization, I find that generalist teams place about 8.5 percentile ranks higher than specialist teams and are 12 percentage points more likely to win a medal than specialist teams. I cannot detect significant differences for the probability of placing among the top three.

Overall, the results from Section 4.2, Section 4.3 and Section 4.4 support Prediction 1 as well as an interpretation of these differences as reflecting fundamental differences in the way generalist teams solve problems, rather than a different ability composition of generalist teams. The next sections turn towards mechanisms.

5 Mechanisms

My theoretical framework attributes quality differences between specialist and generalist teams to coordination costs. As I cannot measure coordination costs directly, I present several sets of diagnostic evidence: First, solution quality differences between generalist and specialist teams are driven by high complexity competitions. Second, the introduction of ChatGPT, a coordination-cost reducing technology, eliminates quality differences between generalist and specialist teams. Finally, I rule out four alternative explanations that could plausibly generate quality differences between team types: Generalists being more motivated and exerting more effort, generalists as better managers, generalists having higher social skills and generalists being able to find better team members.

5.1 Investigating Complexity

A more complex problem requires more coordination, for example because parts of the algorithm may interact in an unforeseen way, or because it may be less clear how to split the problem in sub-tasks. Prediction 2 echoes this intuition. As complexity increases coordination costs, quality differences between generalists and specialists should be larger for higher complexity problems. I use Prediction 2 as a first diagnostic to test whether solution quality differences between generalists and specialist arise because of higher coordination costs for generalist teams. To do so, I split the sample into high and low complexity competitions, and estimate Equation 8 separately for both types of competitions. I define a high-complexity competition as a competition with an overfitting risk in the top quartile of the overfitting distribution (c.f. Section 3.4.1) Figure 4 displays the main coefficients and Table A7 full regression results. Generalist and specialist teams achieve similar ranks (Figure 4a) and are similarly likely to win a medal (Figure 4b) in low-complexity competitions. In high-complexity competitions however, generalist teams achieve higher quality than both mixed and specialist teams. For instance, in low-complexity competitions, generalist teams place only 2 percentile ranks

higher than specialist teams on average, a difference that is not statistically significant, in high-complexity competitions however, the gap between generalist and specialist teams is equivalent to almost 10 percentile ranks and highly significant. In fact, all teams perform worse in high complexity competitions, but specialist teams are most affected by problem complexity. With the exception of *Top 3*, where generalist teams are best throughout, generalist teams only outperform specialist teams in high-complexity competitions.

Robustness To ensure my conclusions are not driven by how I classify complexity, I conduct several robustness checks. First, I change the complexity cut-off for a "high-complexity" competitions to the 50^{th} (Table A8) and to the 90^{th} percentile of competition complexity (Table A9). In both exercises, I only detect significant rank and medal differences between generalist and specialist teams in high-complexity competitions. Second, I use another measure of overfitting risk in a competition, competition *difficulty*, which is defined as $1-$ the share of teams whose best performing submission on the training dataset was also their best performing submission on the test dataset (Trotman, 2019). Again, I can only detect significant differences between generalist and specialist teams' solution quality in high-difficulty competitions. Finally, I assess problem complexity from two different angles, the length of instruction and whether the competition additionally specifies compute constraints. I extract competition descriptions from each competition's overview site on Kaggle. I consider a competition with above average instruction length a high complexity competition. Many competitions with the goal of detecting diseases from biomedical images fall under this category, like segmenting and grading biopsy images (c.f. Bulten et al., 2022), whereas competitions using tabular data with the goal to, for example, predict financial transactions, often have shorter instructions. Table A11 reports results. Here again, quality differences between generalist and specialist teams only emerge in high-complexity competitions. Last, a subset of Kaggle competitions specify additional compute constraints. To be eligible to win, competitors' algorithms have to run in a given time on given hardware. Competitors thus have to optimize along two dimensions, maximizing prediction accuracy while minimizing runtime, a more complex task. Table A12 displays results when splitting the sample into unconstrained and constrained competitions. As before, I only detect significant quality differences between generalist and specialist teams for more complex competitions, that is in competitions *with* compute constraints.

5.2 ChatGPT as a Shock to Coordination Costs

In the last section, I tested for coordination costs as a mechanism through an indirect shifter of coordination costs, problem complexity. Here, I will examine a direct shock

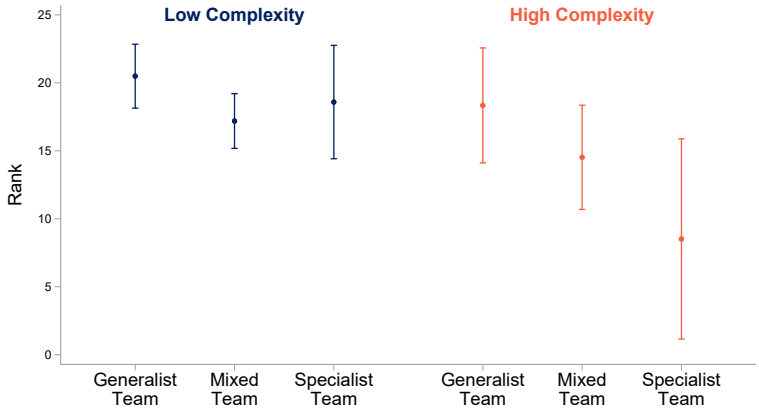
to coordination costs: The introduction of ChatGPT. Technologies like ChatGPT can reduce coordination costs in teams in multiple ways, for example by writing quick code to integrate components, by explaining code written by another team member or by narrowing down the search space (Toner-Rodgers, 2025). At Google, large language model coding assistants were found to significantly speed up the code review process, one of the key elements of teamwork, by translating verbal code improvement suggestions made by other team members to code suggestions (Frömmgen et al., 2024). OpenAI introduced ChatGPT on November 30, 2022 as a way for users to interact with their large language model GPT-3. The introduction of ChatGPT was unexpected, both to the broader public (Figure A4a) and the Kaggle community, but attracted more attention on Kaggle than other large language models, like ChatGPT's predecessor GPT-2 or Google's BERT (Devlin et al., 2019) (Figure A4b and Figure A4c).

Similarly to the complexity analysis in the last section, I split the sample into competitions completed before November 30, 2022, and after. I assign competitions active during the introduction of ChatGPT to the post period, since all active competitions remained active for at least three weeks after the introduction of ChatGPT, enough time for competitors to use ChatGPT. I also truncate the pre-period to the exact same length as the post-period – my sample ends in February 2024 – to ensure that I am not attributing disappearing quality gaps a to lack of power.

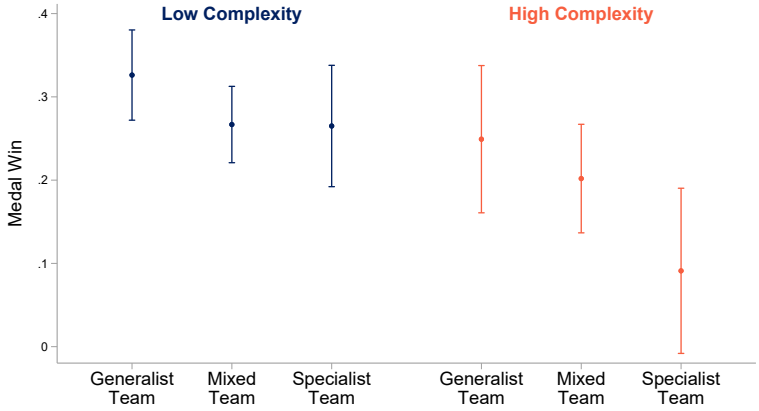
Figure 5 displays main coefficients and Table A13 full split-sample regression results. Before the introduction of ChatGPT, generalist teams place on average 11 percentile ranks higher than specialist teams, but after ChatGPT, specialist teams' ranks are indistinguishable from generalist teams', even suggesting they might outperform generalist teams. In the period before ChatGPT, specialist teams are not more likely to win a medal than solo competitors, and about 29 percentage points less likely than generalist teams to win a medal. This difference shrinks drastically in the post-ChatGPT period, and is not statistically significant anymore. I fail to detect significant differences in the probability of placing among the top three in both periods, likely due to substantially reduced sample sizes compared to Table 6.

Figure 4: Team Type, Complexity and Solution Quality

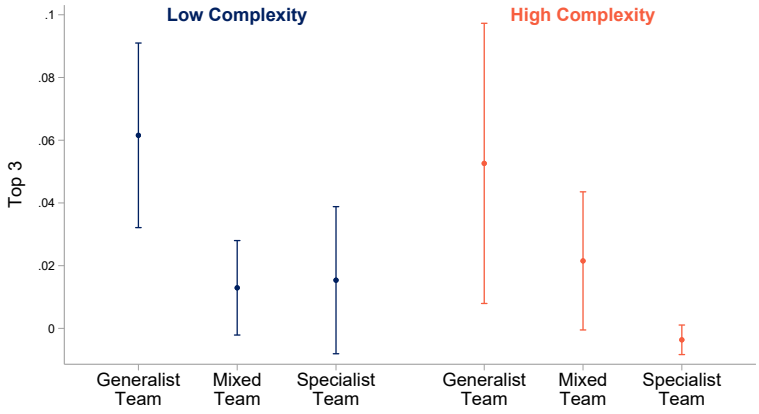
(a) Rank



(b) Medal Win



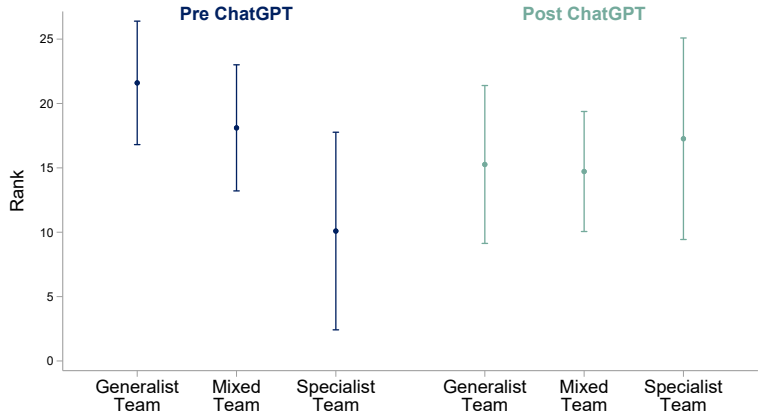
(c) Top 3



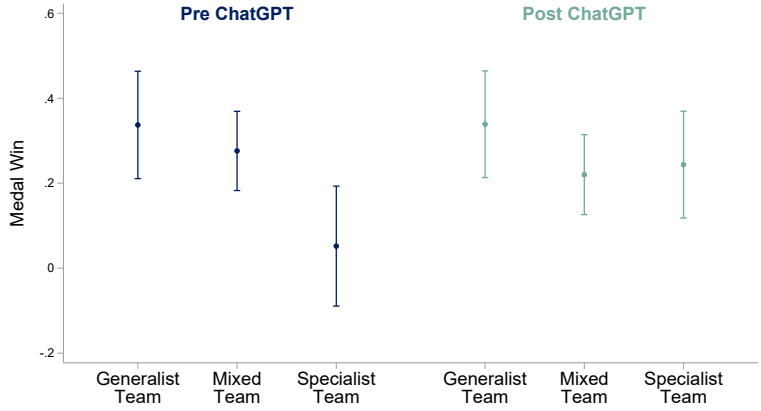
Notes: The figure plots the estimated coefficients τ_1, τ_2 and τ_3 from Equation (8). Regressions are estimated separately for the subset of low- and high-complexity competitions. A competition is classified as a high-complexity competition if its overfitting risk lies in the top quartile of the overfitting risk distribution. Panel (a) displays estimates for *Rank*, panel (b) for *Medal Win* and panel (c) for *Top 3*. I report coefficients from regressions using all covariates and competition fixed effects as in column (5) in Tables 4-6. The corresponding regression results are reported in Table A7.

Figure 5: Team Type, ChatGPT and Solution Quality

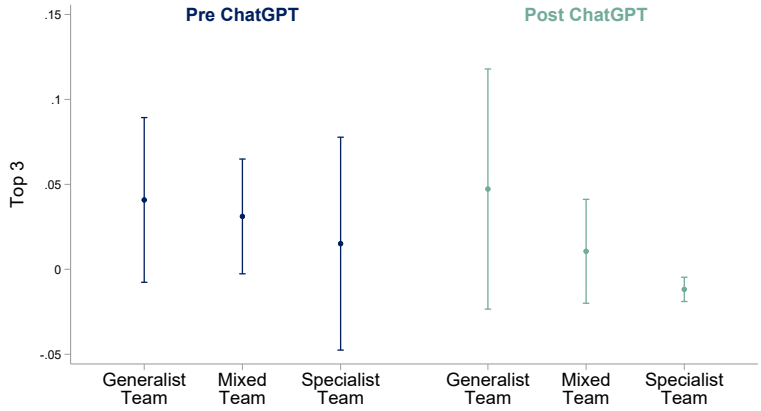
(a) Rank



(b) Medal Win



(c) Top 3



Notes: The figure plots the estimated coefficients τ_1, τ_2 and τ_3 from Equation (8). Regressions are estimated separately for competitions ending before and after the introduction of ChatGPT on November 30, 2022. Panel (a) displays estimates for *Rank*, panel (b) for *Medal Win* and panel (c) for *Top 3*. I report coefficients from regressions using all covariates and competition fixed effects as in column (5) in Tables 4-6. The corresponding regression results are reported in Table A13.

5.3 Effort and Motivation

An alternative explanation for the documented quality differences between generalist and specialist teams is that, rather than these teams having different levels of coordination costs, generalist teams are more motivated and exert more effort. For instance, Weidmann and Deming (2021) find that individuals who increase team performance do so by raising the effort levels of their team peers. Generalists might take that role in their teams, for example because they are able to keep the big picture in view and inspire their team with that vision. In Table 8, I investigate whether teams differ in terms of effort provision and motivation.

Column (1) presents results for Equation (8) with *Total Submissions* as a measure of effort. The number of submissions is highly correlated with performance in a competition. More submissions mean more feedback and more opportunities for fine-tuning the algorithm. Importantly, more submissions mean more work.[22] When controlling for ability proxies, team demographics, team size, code characteristics and competition fixed effects, I do not detect significant differences in the number of total submissions across team types. Columns (2) and (3) report estimates for a finer dimension of motivation, the care taken to select the final submission. I define *Best Public Chosen* and *Best Private Chosen* as indicators whether a team chose the best performing submission on the public (known during the competition) or private (not known to teams during the competition) test set as a final submission. Note that Kaggle selects the best public submission as the final submission unless the team makes a different choice. Coefficients can thus be interpreted as making a conscious choice which requires more effort. I do not find any significant differences in the propensity to choose the best public or private submission between generalist teams, specialist teams and mixed teams.

Since rushing has been found to both be negatively correlated with effort and to reduce the quality of innovation (Weidmann and Deming, 2021; Hill and Stein, 2025), I examine two measures of rushing: *Time to Best*, the time it took a team to reach their highest ranked submission and *Time to Final* the time to reach the submission they chose as their final submission. On average, teams take longer than individual competitors to develop their best and final submissions. The interpretation of this difference is ambiguous. In part, this may explain team's better performance, since rushing a submission could be associated with lower quality (c.f. Hill and Stein, 2025). However, a longer time to submit might also reflect teams' higher coordination costs. While I do not detect any significant differences between team types in the time that it takes a team to develop their best performing submission, generalist teams take slightly

[22]Kaggle enforces a daily submission limit, implicitly censoring effort measures. Kaggle's rationale behind this is to ensure a level playing field – larger teams might have more capacities to submit more and receive more feedback than smaller teams. However, the overwhelming majority of teams submit less than the limit specified, see Figure A5.

Table 8: Team Type and Submission Behavior

Dependent Variable:	Total Submissions	Chose Best Public	Chose Best Private	Time to Best	Time to Final
	(1)	(2)	(3)	(4)	(5)
Generalist Team	43.85***	-0.12***	-0.17***	0.13***	0.15***
	(4.32)	(0.02)	(0.02)	(0.01)	(0.01)
Mixed Team	40.18***	-0.15***	-0.20***	0.11***	0.12***
	(3.03)	(0.02)	(0.02)	(0.01)	(0.01)
Specialist Team	47.80***	-0.13***	-0.19***	0.12***	0.13***
	(5.34)	(0.03)	(0.03)	(0.02)	(0.02)
Competition Experience	-0.08***	-0.00	0.00	0.00***	0.00***
	(0.02)	(0.00)	(0.00)	(0.00)	(0.00)
Previous Rank	0.29***	-0.00***	-0.00***	0.00***	0.00***
	(0.02)	(0.00)	(0.00)	(0.00)	(0.00)
3-Person Team	41.16***	0.01	-0.01	0.04***	0.04***
	(6.22)	(0.03)	(0.03)	(0.01)	(0.01)
4-Person Team	80.34***	0.01	-0.12**	0.08***	0.10***
	(13.25)	(0.06)	(0.05)	(0.02)	(0.02)
5-Person Team	116.20***	-0.10*	-0.16***	0.10***	0.12***
	(13.83)	(0.06)	(0.06)	(0.02)	(0.02)
R^2	0.18	0.11	0.16	0.22	0.22
Observations	60090	60090	60090	60090	60090
Demographic Controls	Yes	Yes	Yes	Yes	Yes
Code Quality Controls	Yes	Yes	Yes	Yes	Yes
Competition FEs	Yes	Yes	Yes	Yes	Yes
p-value (Generalist Team = Mixed Team)	0.48	0.38	0.20	0.13	0.06
p-value (Generalist Team = Specialist Team)	0.56	0.93	0.55	0.58	0.37
p-value (Specialist Team = Mixed Team)	0.20	0.54	0.69	0.63	0.75

Notes: The table reports the estimates of equation (8). The dependent variable is a team's total number of submissions (1), an indicator whether a team chose the submission that performed best in the public test sample as their final competition submission (2), an indicator whether a team chose the submission that performed best in the private test sample as their final competition submission (3), the time the team took to develop their best submission (4), the time the team took to develop their final competition submission (5). I measure time in the fraction of days relative to total competition duration, since the value of a day is different if the competition lasts two weeks or two months. The main explanatory variables are indicators for the team's type: Generalist Team (a team of only generalists), Mixed Team (a team of both generalists and specialists) and Specialist Team (a team of only specialists). Solo competitors are the omitted category. Competition experience measures the average of the number of competitions each team member participated in before competition c. Ability controls include Previous Rank (the average percentile rank achieved by each team member when competing alone in previous competitions) and Experience (the average number of competitions that each team participated in on their own in previous competitions). Demographic controls include share female, average user platform age, country shares for all countries with at least 100 users, and occupation shares for all occupations reported by at least 100 users (c.f. Table A2). Code quality controls include the average of each team member's lines of code, lines of comment, number of code files, and votes received by code files. Standard errors are clustered at the stable team level. Significance levels: $***$ p<0.01, $**$ p<0.05, and $*$ p<0.1.

longer than mixed teams to develop their final submission. For example, in a 100-day competition, generalist teams develop the submission that they select as their final competition submission three days later than mixed teams. Overall, I find little support for differential effort as a mechanism behind generalists-specialist quality differences.

5.4 Management Skills

Recent empirical work has highlighted the importance of managers for allocating workers to specialized tasks and thereby increasing productivity (Minni, 2024; Weidmann

et al., 2024). Consistent with theoretical arguments made by Crémer et al. (2007), the role of generalists in teamwork could also be that of a manager. In Crémer et al. (2007), different units of an organization engage in solving parts of a problem. However, they do not "speak" the same language, i.e. they are specialized on different parts of the problem. A manager, who "speaks" both languages, assigns the parts of the problem to the person who can solve them. An interpretation of their theoretical model is that individuals with broader knowledge make for better managers.

Although there are no explicit managers in Kaggle teams, each team is required to select a team leader. The team leader has de facto authority: They are the primary point of contact between the platform and the team, have the sole authority over accepting new team members and merging teams, and make the final submission selections.[23] These powers and responsibilities align closely with a management role. To investigate whether generalists are better managers, I estimate the following regression in the sample of mixed teams[24]:

$$\text{Y}_{tc} = \alpha + \beta_1 \text{Generalist Team Leader}_{tc} + X'_{tc}\beta + \epsilon_{tc} \tag{11}$$

where Generalist Team Leader$_{tc}$ is an indicator that is equal to one if team t's team leader is a generalist, and Y_{tc} and X_{tc} are defined as in Equation (8).

Table 9 presents results. I do not find any evidence that teams with a generalist team leader perform any different than teams with a specialist leader. In columns 4-6, I additionally split ability controls into the ability of the team leader and that of the other members and control for both separately. The effects of team lead and team member ability are indistinguishable (F-Tests: 0.31 (Column 4), 0.34 (Column 5), 0.32 (Column 6)).

5.5 Social Skills

Social skills are crucial for teamwork (Deming, 2017; Weidmann and Deming, 2021; Adhvaryu et al., 2023). For example, Deming (2017) explicitly models social skills as a factor that reduces coordination frictions between specialized workers. Social skills could be correlated with the degree of specialization, for example if a broader set of interests is correlated with an open mind towards people. Any quality difference between generalist and specialist teams would then be a result of social skills rather than coordination costs differing by the degree of specialization. Do generalist teams have higher levels of social skills? Since I do not have access to explicit measures of social skills, I analyze users' public communication behavior on the platform to capture social skills. Kaggle users can engage in forum discussions. Aside from some platform-wide

[23] See Kaggle.com (2024a) and Reade (2024).

[24] As only generalists can be team leaders in generalist teams and analogously for specialists.

Table 9: Generalist Team Leaders in Mixed Teams

Dependent Variable:	*Rank*	*Medal Win*	*Top 3*	*Rank*	*Medal Win*	*Top 3*
	(1)	(2)	(3)	(4)	(5)	(6)
Generalist Team Leader	-0.26	-0.03	-0.01	0.14	-0.06	-0.01
	(1.69)	(0.04)	(0.01)	(1.71)	(0.05)	(0.02)
Competition Experience	0.15**	0.00**	-0.00			
	(0.07)	(0.00)	(0.00)			
Previous Rank	0.29***		0.00			
	(0.08)		(0.00)			
Previous Medal Wins		0.51***				
		(0.17)				
Team Lead's Experience				0.02	0.00	-0.00
				(0.04)	(0.00)	(0.00)
Team Members' Experience				0.08*	0.00	-0.00
				(0.04)	(0.00)	(0.00)
Team Lead's Rank				0.18***		0.00**
				(0.06)		(0.00)
Team Members' Rank				0.09		0.00
				(0.07)		(0.00)
Team Lead's Medal Wins					0.49***	
					(0.12)	
Team Members' Medal Wins					0.31**	
					(0.12)	
3-Person Team	5.06**	0.15***	0.03	7.35***	0.20***	0.05*
	(2.38)	(0.05)	(0.02)	(2.10)	(0.06)	(0.03)
4-Person Team	12.48***	0.26***	0.04	10.83***	0.25***	0.04
	(2.32)	(0.06)	(0.03)	(2.33)	(0.06)	(0.04)
5-Person Team	13.19***	0.27***	-0.02	12.02***	0.32***	-0.03
	(2.59)	(0.07)	(0.03)	(2.56)	(0.07)	(0.04)
R^2	0.56	0.52	0.40	0.62	0.57	0.45
Observations	830	830	830	655	655	655
Demographic Controls	Yes	Yes	Yes	Yes	Yes	Yes
Code Quality Controls	Yes	Yes	Yes	Yes	Yes	Yes
Competition FEs	Yes	Yes	Yes	Yes	Yes	Yes

Notes: The table reports the estimates of equation (11). The dependent variable is a team's percentile rank (1) and (4), an indicator whether a team won a medal (2) and (5), and an indicator whether a team placed among the Top 3 (3) and (6). The main explanatory variable is an indicator whether the team leader is a generalist. Competition experience measures the average of the number of competitions each team member participated in before competition c. Ability controls include Previous Rank (the average percentile rank achieved by each team member when competing alone in previous competitions) and Experience (the average number of competitions that each team participated in on their own in previous competitions). Team Lead and Team Members' variables are defined in the same way, but included disaggregated by team leader and non-team leader team members. Demographic controls include share female, average user platform age, country shares for all countries with at least 100 users, and occupation shares for all occupations reported by at least 100 users (c.f. Table A2). Code quality controls include the average of each team member's lines of code, lines of comment, number of code files, and votes received by code files. Significance levels: $***$ p<0.01, $**$ p<0.05, and $*$ p<0.1.

forums for questions or product feedback, each competition has an attached discussion forum in which users can and very frequently do interact. From these messages, I construct several metrics of social skills, and investigate whether team types differ in their levels of social skills. Note that I do not have access to private, within team communication data. A necessary assumption is thus that interactions with a larger group of peers generalize to behavior within a smaller group of peers. This is plausible in this context, since public communication is not anonymous. I estimate the following regression on the sample of stable teams:

$$Y_t = \alpha + \tau_1 \text{Generalist Team}_t + \tau_2 \text{Mixed Team}_t + \tau_3 \text{Specialist Team}_t + X_t'\beta + \epsilon_i \qquad (12)$$

where Y_t is a measure of the team's average social skills, Generalist Team$_t$, Mixed Team$_t$ and Specialist Team$_t$ are defined as before, and X_t is a vector of team-level demographics and team size indicators. I first evaluate whether the total volume of messages differs between team types. Engaging more with public message boards is an indicator for a desire and willingness to communicate with others, a prerequisite of social skills. I do not find any significant differences between team types (Table 10, Column 1). I next turn to the content of the messages: Experimental research from psychology indicates that explicit expressions of gratitude help initiate social relationships (Williams and Bartlett, 2015). I thus investigate what portion of a user's forum posts include "thank you's" and how many "thank you's" an average message contains (Table 10, Columns 2 and 3). Specialist Teams are comprised of users who thank others more often. I also examine how often other users are mentioned in messages and how often (Table 10, Columns 4 and 5)[25] as a more direct measure of social ties. I do not find any significant differences. Finally, I construct a simple measure of friendliness: whether a message includes graphical representations of emotion, i.e. smileys.[26] Table 10, Column 6 presents results. Among all teams, specialist teams include most smileys in their messages. Taken together, I do not find striking differences in various measures of social skills between team types. Since specialist teams appear at least as, if not more, sociable as generalist teams, social skill differences are unlikely to explain the latter's superior performance.

5.6 Team Member Matching

For the main part of this paper, my focus has been on differing *costs* of teamwork between generalists and specialists. However, the productivity *gains* from collaboration might also differ. Both the literature on peer effects (e.g. Hamilton et al., 2003; Mas and Moretti, 2009) and macroeconomic models of labor market inequality (Cornelissen et al., 2017; Jarosch et al., 2021; Herkenhoff et al., 2024; Freund, 2024) stress coworker complementarities in teams, finding that workers' productivity disproportionately increases when matched with higher-ability coworkers. Although it is unclear whether and to which extend these complementarities exist in science – Azoulay et al. (2010) find positive, Waldinger (2011) no, and Ahmadpoor and Jones (2019) negative productivity spillovers – we might still be worried that specialists and generalists sort into teams with different spreads in ability. For instance, generalist teams might systematically be more heterogeneous in ability, because generalists' broader knowledge might allow them to assess others' abilities better and find higher ability team members given their

[25] References to other users are clearly identifiable through HTML tags in the unprocessed message data which I have access to.

[26] Specifically, I check for the presence of the following character sequences: *:) =) ;) ;-) :-) (: (= (; (-; (-:* and the unicode strings for several graphical emoticons.

Table 10: Team Type and Social Skills

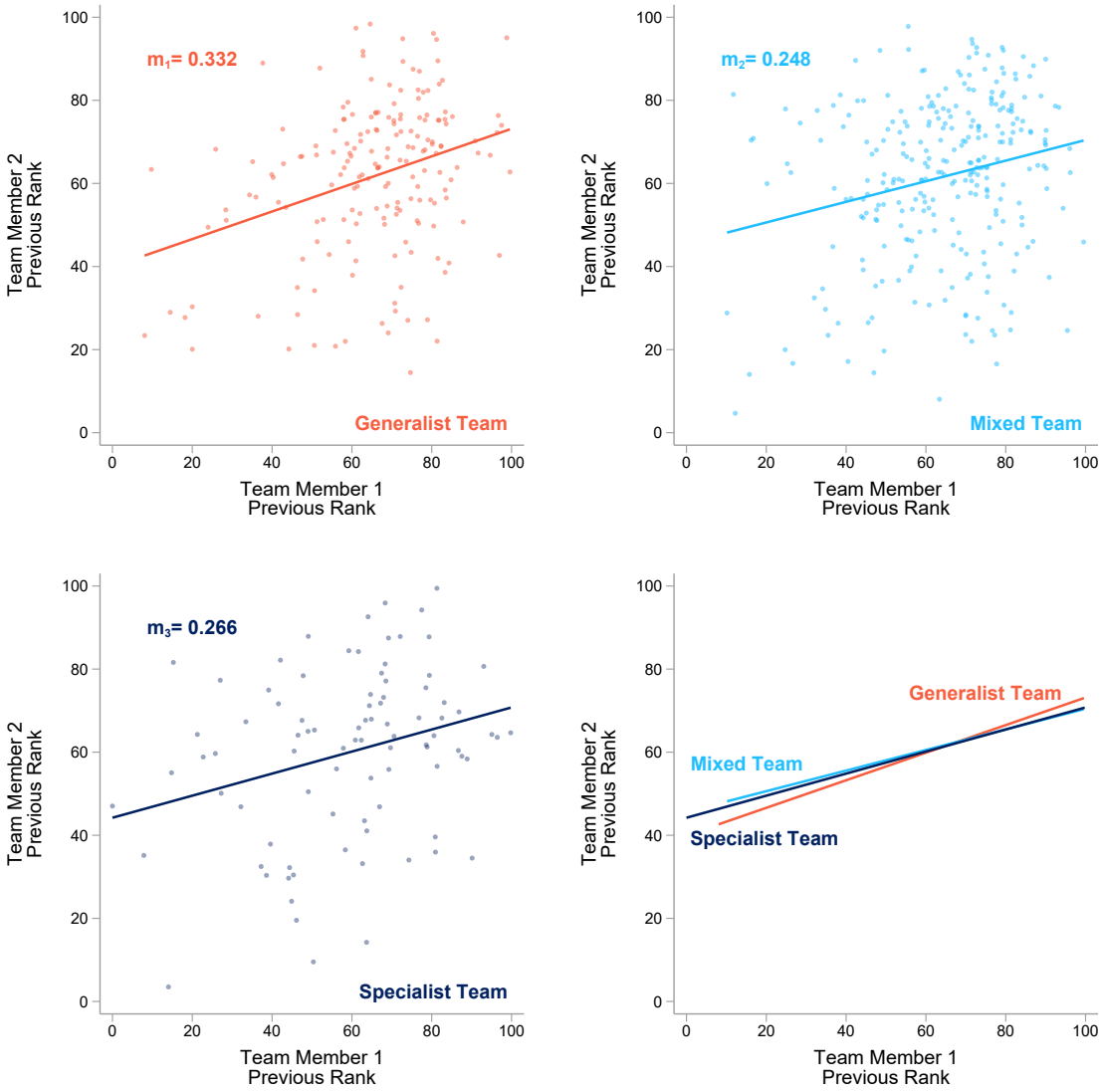
Dependent Variable:	Number Messages	Share with Thanks	Thanks per Message	Share with user mentions	User mentions per message	Smileys per message
	(1)	(2)	(3)	(4)	(5)	(6)
Generalist Team	139.58***	-0.07***	-0.07***	0.03***	0.04***	0.02***
	(16.12)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)
Mixed Team	119.59***	-0.05***	-0.05***	0.04***	0.05***	0.01***
	(11.85)	(0.01)	(0.01)	(0.01)	(0.01)	(0.00)
Specialist Team	140.73***	-0.03**	-0.03**	0.05***	0.05***	0.03***
	(30.84)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)
3-Person Team	105.48***	-0.02*	-0.02	0.02**	0.04***	0.00
	(25.07)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)
4-Person Team	283.99***	-0.04***	-0.03**	0.04**	0.06***	-0.00
	(81.83)	(0.01)	(0.01)	(0.02)	(0.02)	(0.01)
5-Person Team	203.89***	-0.03**	-0.03**	0.04***	0.06***	-0.00
	(35.67)	(0.01)	(0.01)	(0.01)	(0.02)	(0.01)
R^2	0.05	0.03	0.03	0.05	0.03	0.03
Observations	13030	13030	13030	13030	13030	13030
Demographic Controls	Yes	Yes	Yes	Yes	Yes	Yes
p-value (Generalist Team = Mixed Team)	0.30	0.03	0.02	0.22	0.30	0.62
p-value (Generalist Team = Specialist Team)	0.97	0.01	0.01	0.13	0.41	0.09
p-value (Specialist Team = Mixed Team)	0.51	0.21	0.20	0.46	0.93	0.04

Notes: The table reports the estimates of equation (12). The dependent variables are the average number of forum messages posted by a team's members (1), the average share of their messages that contain expressions of gratitude (2), as well as the share of these per message (3), the average share of messages mentioning other users (4), as well as the share of user mentions per message (5), and the average share of smileys per message (6). The main explanatory variables are indicators for the team's type: Generalist Team (a team of only generalists), Mixed Team (a team of both generalists and specialists) and Specialist Team (a team of only specialists). Solo competitors are the omitted category. Significance levels: $***$ p<0.01, $**$ p<0.05, and $*$ p<0.1.

own ability. If one high ability team member lifts the others up disproportionately, a generalist team with a larger heterogeneity in ability would produce a higher quality solution than a specialist team with the same average ability.

To explore whether different ability matching patterns could explain generalist-specialist quality differences, I examine the correlation between team members' previous performance in Figure 6. I focus on a two-person team's first collaboration, and designate the team leader as team member 1. The first three panels in Figure 6 show scatter plots of teams' ability combinations and the association between team members' ability as a linear fit, separately for each team type. I find similar correlations between the two team members' previous performance for all team types: When one team member has previously placed 10 percentile ranks higher, the other team member's previous performance increases between 2.5 to 3.3 percentile ranks. These associations are statistically indistinguishable. The p-value of the Chi2-Test comparing the differences in slopes between the sample of generalist teams and mixed teams is 0.42, 0.57 comparing generalist and specialist teams and 0.88 for specialist and mixed teams. Given their own ability, generalists do not seem to be able to find better team members than specialists, making it unlikely that differing heterogeneity of team ability drives quality differences of specialists and generalists.

Figure 6: Assortative Matching in Different Team Types



Notes: The figure shows the correlations between team member's previous percentile ranks in two-person teams. Team member 1 is the team leader. Panel (a) displays generalist teams, panel (b) mixed teams and panel (c) specialist teams. m_1, m_2, and m_3 are coefficients from a regression of team member 2 previous rank on team member 1 previous rank. Panel (d) compares the correlations across team types.

6 Conclusion

Making teamwork succeed is crucial for maintaining high rates of innovation in a time in which ideas are getting harder to find (Bloom et al., 2020) and innovation is increasingly produced in teams (Wuchty et al., 2007). In this paper I show that there is a tension between specialization and teamwork, as generalist teams produce higher quality innovation than specialist teams. I argue that this is a result of coordination costs, which are higher in specialist teams. Using data from online machine learning competitions, I employ several empirical strategies to disentangle the effect of team type

from the contributions of individual team members. I consistently find that generalist teams produce higher quality than specialist teams. I provide several pieces of evidence that point towards coordination costs as a channel. When working on a low-complexity problem, specialist teams perform at par with generalist teams. Innovation quality differences only arise in high-complexity problems, for which coordination costs are particularly high. A shift in coordination costs due to the introduction of ChatGPT also removes quality differences between generalists and specialists. I find no evidence for alternative mechanisms, such as motivation, management skills, social skills and collaborator matching, which might favor generalist teams. Overall, my findings highlight that we need to take skill *breadth* into account if we want to build innovative teams. Universities and research agencies invest considerable funds into programs designed to foster interdisciplinary research. While previous research suggests that teams from more diverse disciplinary backgrounds produce higher-impact innovation (Uzzi et al., 2013; Allocca, 2024), I find that coordination costs hamper the productivity of such teams and might lead these endeavors to fail.

References

Abowd, J. M., F. Kramarz, and D. N. Margolis (1999). High Wage Workers and High Wage Firms. *Econometrica 67*(2), 251–333.

Adhvaryu, A., N. Kala, and A. Nyshadham (2023). Returns to On-the-Job Soft Skills Training. *Journal of Political Economy 131*(8), 2165–2208.

Agrawal, A., A. Goldfarb, and F. Teodoridis (2016). Understanding the Changing Structure of Scientific Inquiry. *American Economic Journal: Applied Economics 8*(1), 100–128.

Ahmadpoor, M. and B. F. Jones (2019). Decoding team and individual impact in science and invention. *Proceedings of the National Academy of Sciences 116*(28), 13885–13890.

Albrecht, J., T. Qin, and M. Yates (2024). NeurIPS 2024 Call for Competitions. https://blog.neurips.cc/2024/03/03/neurips-2024-call-for-competitions/. last retrieved 2025/02/27.

Allocca, A. (2024). An Empirical Study of Team Formation and Performance. *mimeo University of Munich*.

Association for Computing Machinery (2025). KDD Cup Archives. https://kdd.org/kdd-cup. last retrieved 2025/02/25.

Azoulay, P., J. S. Graff Zivin, and J. Wang (2010). Superstar Extinction. *The Quarterly Journal of Economics 125*(2), 549–589.

Becker, G. S. and K. M. Murphy (1992). The Division of Labor, Coordination Costs, and Knowledge. *The Quarterly Journal of Economics 107*(4), 1137–1160.

Bennet, J. and S. Lanning (2007). The Netflix Prize. *Proceedings of the KDD Cup and Workshop 2007*.

Bloom, N., C. I. Jones, J. Van Reenen, and M. Webb (2020, April). Are Ideas Getting Harder to Find? *American Economic Review 110*(4), 1104–44.

Bolton, P. and M. Dewatripont (1994). The Firm as a Communication Network. *The Quarterly Journal of Economics 109*(4), 809–839.

Bonhomme, S. (2022). Teams: Heterogeneity, Sorting and Complementarity. *mimeo University of Chicago*.

Boudreau, K. J., N. Lacetera, and K. R. Lakhani (2011). Incentives and Problem Uncertainty in Innovation Contests: An Empirical Analysis. *Management Science 57*, 843–863.

Boudreau, K. J., K. R. Lakhani, and M. Menietti (2016). Performance responses to competition across skill levels in rank-order tournaments: field evidence and implications for tournament design. *The RAND Journal of Economics 47*(1), 140–165.

Bulten, W., K. Kartasalo, P.-H. C. Chen, P. Ström, H. Pinckaers, K. Nagpal, Y. Cai, D. F. Steiner, H. van Boven, R. Vink, C. Hulsbergen-van de Kaa, J. van der Laak, M. B. Amin, A. J. Evans, T. van der Kwast, R. Allan, P. A. Humphrey, H. Grönberg, H. Samaratunga, B. Delahunt, T. Tsuzuki, T. Häkkinen, L. Egevad, M. Demkin, S. Dane, F. Tan, M. Valkonen, G. S. Corrado, L. Peng, C. H. Mermel, P. Ruusuvuori, G. Litjens, M. Eklund, A. Brilhante, A. Çakır, X. Farré, K. Geronatsiou, V. Molinié, G. Pereira, P. Roy, G. Saile, P. G. O. Salles, E. Schaafsma, J. Tschui, J. Billoch-Lima, E. M. Pereira, M. Zhou, S. He, S. Song, Q. Sun, H. Yoshihara, T. Yamaguchi, K. Ono, T. Shen, J. Ji, A. Roussel, K. Zhou, T. Chai, N. Weng, D. Grechka, M. V. Shugaev, R. Kiminya, V. Kovalev, D. Voynov, V. Malyshev, E. Lapo, M. Campos, N. Ota, S. Yamaoka, Y. Fujimoto, K. Yoshioka, J. Juvonen, M. Tukiainen, A. Karlsson, R. Guo, C.-L. Hsieh, I. Zubarev, H. S. T. Bukhar, W. Li, J. Li, W. Speier, C. Arnold, K. Kim, B. Bae, Y. W. Kim, H.-S. Lee, J. Park, and the PANDA challenge consortium (2022). Artificial intelligence for diagnosis and gleason grading of prostate cancer: the panda challenge. *Nature Medicine 28*(1), 154–163.

Chen, Y. (2021). Team-Specific Human Capital and Team Performance: Evidence from Doctors. *American Economic Review 111*(12), 3923–62.

Constantine, N. and S. Correia (2021). Putting the "i" in Fixed Effects: Individual Fixed Effects with Group Level Outcomes. *mimeo Board of Governors of the Federal Reserve System*.

Cornelissen, T., C. Dustmann, and U. Schönberg (2017). Peer Effects in the Workplace. *The American Economic Review 107*(2), 425–456.

Crémer, J., L. Garicano, and A. Prat (2007). Language and the Theory of the Firm. *The Quarterly Journal of Economics 122*(1), 373–407.

Deming, D. J. (2017). The Growing Importance of Social Skills in the Labor Market. *The Quarterly Journal of Economics 132*(4), 1593–1640.

Dessein, W. and T. Santos (2006). Adaptive Organizations. *Journal of Political Economy 114*(5), 956–995.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019, June). BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, pp. 4171–4186. Association for Computational Linguistics.

Freund, L. (2024). Superstar Teams. *mimeo Columbia Business School*.

Frömmgen, A., J. Austin, P. Choy, N. Ghelani, L. Kharatyan, G. Surita, E. Khrapko, P. Lamblin, P.-A. Manzagol, M. Revaj, M. Tabachnyk, D. Tarlow, K. Villela, D. Zheng,

S. Chandra, and P. Maniatis (2024). Resolving Code Review Comments with Machine Learning. In *2024 IEEE/ACM 46th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*.

Garicano, L. (2000). Hierarchies and the Organization of Knowledge in Production. *Journal of Political Economy 108*(5), 874–904.

Garicano, L. and E. Rossi-Hansberg (2006). Organization and Inequality in a Knowledge Economy. *The Quarterly Journal of Economics 121*(4), 1383–1435.

h2o.ai (2025). Meet our Kaggle Grandmasters. https://h2o.ai/company/team/kaggle-grandmasters/. last retrieved 2025/03/07.

Hamilton, B. H., J. A. Nickerson, and H. Owan (2003). Team Incentives and Worker Heterogeneity: An Empirical Analysis of the Impact of Teams on Productivity and Participation. *Journal of Political Economy 111*(3), 465–497.

He, S., R. Huang, J. Townley, R. C. Kretsch, T. G. Karagianes, D. B. T. Cox, H. Blair, D. Penzar, V. Vyaltsev, E. Aristova, A. Zinkevich, A. Bakulin, H. Sohn, D. Krstevski, T. Fukui, F. Tatematsu, Y. Uchida, D. Jang, J. S. Lee, R. Shieh, T. Ma, E. Martynov, M. V. Shugaev, H. S. T. Bukhari, K. Fujikawa, K. Onodera, C. Henkel, S. Ron, J. Romano, J. J. Nicol, G. P. Nye, Y. Wu, C. Choe, W. Reade, E. Participants, and R. Das (2024). Ribonanza: deep learning of RNA structure through dual crowdsourcing. *bioRxiv : the preprint server for biology*.

Herkenhoff, K., J. Lise, G. Menzio, and G. M. Phillips (2024). Production and Learning in Teams. *Econometrica 92*(2), 467–504.

Hill, R. and C. Stein (2025, 02). Race to the Bottom: Competition and Quality in Science. *The Quarterly Journal of Economics*, qjaf010. forthcoming.

Howard, J. (2024). Is it a bird? Creating a model from your own data. https://www.kaggle.com/code/jhoward/is-it-a-bird-creating-a-model-from-your-own-data. last retrieved 2025/03/07.

Jaravel, X., N. Petkova, and A. Bell (2018, April). Team-Specific Capital and Innovation. *American Economic Review 108*(4-5), 1034–73.

Jarosch, G., E. Oberfield, and E. Rossi-Hansberg (2021). Learning from Coworkers. *Econometrica 89*(2), 647–676.

Jones, B. F. (2009). The Burden of Knowledge and the "Death of the Renaissance Man": Is Innovation Getting Harder? *The Review of Economic Studies 76*(1), 283–317.

Kaggle.com (2024a). How to Use Kaggle. https://www.kaggle.com/docs/competitions. last retrieved 2025/03/07.

Kaggle.com (2024b). Kaggle Community Guidelines. https://www.kaggle.com/community-guidelines. last retrieved 2025/03/07.

Kaggle.com (2024c). Kaggle Rankings. https://www.kaggle.com/rankings?group=competitions. last retrieved 2025/03/07.

Kaggle.com (2025). Kaggle Progression System. https://www.kaggle.com/progression. last retrieved 2025/03/07.

Kireyev, P. (2020). Markets for ideas: prize structure, entry limits, and the design of ideation contests. *The RAND Journal of Economics 51*(2), 563–588.

Koch, B. J. and D. Peterson (2024). From Protoscience to Epistemic Monoculture: How Benchmarking Set the Stage for the Deep Learning Revolution. *mimeo University of Chicago*.

Lazear, E. P. and S. Rosen (1981). Rank-Order Tournaments as Optimum Labor Contracts. *Journal of Political Economy 89*(5), 841–864.

Lemus, J. and G. Marshall (2021). Dynamic Tournament Design: Evidence from Prediction Contests. *Journal of Political Economy 129*(2), 383–420.

Lemus, J. and G. Marshall (2024). Teamwork in Contests. *The Review of Economics and Statistics*, 1–45.

Mas, A. and E. Moretti (2009, March). Peers at Work. *American Economic Review 99*(1), 112–45.

Minni, V. (2024). Making the Invisible Hand Visible: Managers and the Allocation of Workers to Jobs. *mimeo Chicago Booth*.

Mohri, M., A. Rostamizadeh, and A. Talwalkar (2018). *Foundations of Machine Learning* (Second Edition ed.). MIT Press.

Mullainathan, S. and J. Spiess (2017). Machine Learning: An Applied Econometric Approach. *Journal of Economic Perspectives 31*(2), 87–106.

NVIDIA (2025). Kaggle Grandmasters. https://www.nvidia.com/en-us/ai-data-science/kaggle-grandmasters/. last retrieved 2025/02/25.

Pavão, A. (2023, December). *Methodology for Design and Analysis of Machine Learning Competitions*. Theses, Université Paris-Saclay.

Pearce, J. (2023). Idea Production and Team Structure. *mimeo Federal Reserve Bank of New York*.

Plotts, J. and M. Risdal (2023). Meta Kaggle Code. https://www.kaggle.com/ds/3240808. downloaded 2024/01/08.

Reade, W. (2024). Submission selections can only be made by team leads. https://www.kaggle.com/competitions/home-credit-default-risk/discussion/64440. last retrieved 2025/03/07.

Reich, V. (2023). Signal or Noise - Signalling Skill among Data Professionals. *mimeo University of Munich*.

Risdal, M. and T. Bozsolik (2022). Meta Kaggle. https://www.kaggle.com/ds/9. downloaded 2024/2/07.

Roelofs, R., V. Shankar, B. Recht, S. Fridovich-Keil, M. Hardt, J. Miller, and L. Schmidt (2019). A Meta-Analysis of Overfitting in Machine Learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Volume 32. Curran Associates, Inc.

Smith, A. (1776). An Inquiry into the Nature and Causes of the Wealth of Nations. available through Project Gutenberg, https://www.gutenberg.org/ebooks/38194. last retrieved 2025/03/07.

Sun, L. and S. Abraham (2021). Estimating dynamic treatment effects in event studies with heterogeneous treatment effects. *Journal of Econometrics 225*(2), 175–199.

Taylor, C. R. (1995). Digging for Golden Carrots: An Analysis of Research Tournaments. *The American Economic Review 85*(4), 872–890.

Teodoridis, F. (2018). Understanding Team Knowledge Production: The Interrelated Roles of Technology and Expertise. *Management Science 64*(8), 3625–3648.

Teodoridis, F., M. Bikard, and K. Vakili (2019). Creativity at the Knowledge Frontier: The Impact of Specialization in Fast- and Slow-paced Domains. *Administrative Science Quarterly 64*(4), 894–927.

Toner-Rodgers, A. (2025). Artificial Intelligence, Scientific Discovery, and Product Innovation. *mimeo Massachusetts Institute of Technology*.

Trotman, J. (2019). Meta Kaggle: Competition Shake-up. https://www.kaggle.com/jtrotman/meta-kaggle-competition-shake-up. last retrieved 2025/03/07.

Uzzi, B., S. Mukherjee, M. Stringer, and B. Jones (2013). Atypical Combinations and Scientific Impact. *Science 342*(6157), 468–472.

Waldinger, F. (2011). Peer Effects in Science: Evidence from the Dismissal of Scientists in Nazi Germany. *The Review of Economic Studies 79*(2), 838–861.

Wang, Y., L. Hung, A. D. Gotmare, N. D. Q. Bui, J. Li, and S. C. H. Hoi (2023). CodeT5+: Open Code Large Language Models for Code Understanding and Generation. *mimeo Salesforce AI Research*.

Weidmann, B. and D. J. Deming (2021). Team Players: How Social Skills Improve Team Performance. *Econometrica 89*(6), 2637–2657.

Weidmann, B., J. Vecci, F. Said, D. J. Deming, and S. R. Bhalotra (2024). How do you find a good manager? *National Bureau of Economic Research Working Paper*.

Williams, L. A. and M. Y. Bartlett (2015). Warm thanks: gratitude expression facilitates social affiliation in new relationships via perceived warmth. *Emotion 15*(1), 1–5.

Wuchty, S., B. F. Jones, and B. Uzzi (2007). The Increasing Dominance of Teams in Production of Knowledge. *Science 316*(5827), 1036–1039.

Appendix

- Appendix A provides further details and derivations for the theoretical framework in Section 2.
- Appendix B provides further details on the construction of the data.
- Appendix C reports robustness checks and additional findings related to Section 4.
- Appendix D reports robustness checks and additional findings related to Section 5.

A Model: Derivations

To derive the predictions in Section 2.2, I solve the model outlined in Section 2 via backward induction, first deriving optimal effort, then optimal task division, and finally the conditions for collaborating.

Effort Choices and the Division of Tasks within Teams When working alone, a scientist has to execute all tasks by themselves, meaning that Q_p^a and $e_i^{a^*}$ are zero when the problem falls outside of the range of tasks a scientist can solve, i.e. $\Theta_p > b_i$.
Else, optimal efforts are given by:

$$\frac{\partial q(\Theta_p, \sigma_i, e_i^{a*})}{\partial e_i^{a*}} = \Theta_p \frac{\partial c(e_i^{a^*})}{\partial e_i^{a*}} \tag{13}$$

As $\partial q(w_i, \sigma_i, e_i)/\partial e_i \partial \sigma_i > 0$ and $\partial c(e_i)/\partial \sigma_i = 0$:
Corollary 1: Of two scientists working alone on the same problem Θ_p, the more specialized scientist (higher σ_i) will exert higher effort. Conversely, more specialized scientists will work on fewer problems alone.
Looking at team production, let us first consider the case where both scientists have $b_i \geq \Theta_p$. Optimal efforts are given by:

$$\frac{1}{2}\frac{\partial q(w_i, \sigma_i, e_i^{t*},)}{\partial e_i^{t*}} = w_i \frac{\partial c(e_i^{t*})}{\partial e_i^{t*}} \tag{14}$$

Since Equation (14) has to hold for both team members, the optimal division of labor within a team is given by:

$$\frac{\partial q(w_i, \sigma_i, e_i^{t*},)}{\partial e_i^{t*}} \Big/ \frac{\partial q(w_j, \sigma_j, e_j^{t*},)}{\partial e_j^{t*}} = \frac{w_i}{w_j} \frac{\partial c(e_i^{t*})}{\partial e_i^{t*}} \Big/ \frac{\partial c(e_j^{t*})}{\partial e_j^{t*}} \tag{15}$$

meaning, two equally specialized scientists will execute exactly half of the necessary tasks.
In an homogeneous team ($\sigma_i = \sigma_j$), optimal efforts are then given by:

$$\frac{\partial q(\Theta_p/2, \sigma_i, e_i^{t*})}{\partial e_i^{t*}} = \Theta_p \frac{\partial c(e_i^{t*})}{\partial e_i^{t*}} \tag{16}$$

Due to $\partial q(w_i, \sigma_i, e_i)/\partial e_i \partial w_i < 0$, for homogeneous teams, individual efforts per task in a team are larger than when working alone. In an mixed team ($\sigma_i \neq \sigma_j$), the more specialized scientist executes a larger share of tasks to equalize production across the team. Denote as α the share of tasks taken over by the more specialized scientist. Since $\alpha > 1/2$, the change in effort for the more specialized scientist is ambiguous. Efforts only increase if $\partial q(w_i, \sigma_i, e_i)/\partial e_i \partial w_i < 0$ is sufficiently small. However, efforts of the

less specialized scientist always increase.
Corollary 2: In a team of two equally specialized scientists, both scientists will exert more effort per tasks than when working alone. In a team of two scientists with different levels of specialization, the less specialized scientist will exert more effort, whereas the change in effort for the more specialized scientist is ambiguous. Homogeneous teams exert more effort than individuals for the same problem, mixed teams the same or more. Now, what happens when two scientists collaborate, and one of them has $b_j < \Theta_p$? Clearly, the less specialized scientist now has to execute tasks $\Theta_p - b_j$, and potentially an additional share $(1 - \varphi)$ of the tasks $\theta < b_j$. Equation (14) becomes:

$$\frac{\partial q(\Theta_p - \varphi b_j, \sigma_i, e_i^{t*})}{\partial e_i^{t*}} \Big/ \frac{\partial \varphi b_j, \sigma_j, q(e_j^{t*},)}{\partial e_j^{t*}} = \frac{\Theta_p - \varphi b_j}{\varphi b_j} \frac{\partial c(e_i^{t*})}{\partial e_i^{t*}} \Big/ \frac{\partial c(e_j^{t*})}{\partial e_j^{t*}} \tag{17}$$

Since we know that in an efficient team, the more specialized scientist executes a larger share of tasks, a team with $\Theta_p/2 \geq b_j$ will never be efficient. In this case, scientist j executes b_j tasks, and scientist i the rest.

Collaboration Choices Recall that, for a given problem Θ_p, a scientist σ_i collaborates if:

$$\frac{1}{2} \underbrace{\{Q_p^t - C(\gamma_p, \sigma_i, \sigma_j)\}}_{\text{team solution quality}} - \underbrace{w_i c(e_i^{t^*})}_{\text{total cost of effort}} \geq \underbrace{Q_p^a}_{\text{individual solution quality}} - \underbrace{\Theta_p c(e_i^{a^*})}_{\text{total cost of effort}} \tag{18}$$

where $e_i^{a^*}$ and $e_i^{t^*}$ are optimal efforts when working alone or working as a team. For a homogeneous team, ($\sigma_i = \sigma_j$) Equation (18) becomes:

$$\underbrace{\frac{1}{2} Q_p^t - \frac{\Theta_p}{2} c(e_i^{t^*}) - Q_p^a + \Theta_p c(e_i^{a^*})}_{\text{return to collaboration}} \geq \underbrace{\frac{1}{2} C(\gamma_p, \sigma_i, \sigma_i)}_{\text{collaboration costs}} \tag{19}$$

That is, the return to collaboration has to be at least as large as the collaboration costs borne by one team member.
A mixed team collaborates if:

$$\underbrace{\frac{1}{2} Q_p^t - \alpha \Theta_p c(e_i^{t^*}) - Q_p^a + \Theta_p c(e_i^{a^*})}_{\text{return to collaboration for more specialized scientist}} \geq \underbrace{\frac{1}{2} C(\gamma_p, \sigma_i, \sigma_j)}_{\text{collaboration costs}} \tag{20}$$

for the more specialized scientist. If Equation (20) holds for the more specialized scientist, it also holds for the less specialized scientist, because working alone is even less profitable for the less specialized scientist. Let's look at the case where one scientist has $b_i \geq \Theta_p$. Clearly, they will never collaborate with a scientist who has the same level of specialization, since they will not be able to execute all tasks necessary to solve the

problem. In this scenario, when will it be profitable for a less specialized scientist to collaborate with that more specialized scientist?

$$\underbrace{\frac{1}{2}Q_p^t - (\Theta_p - \varphi b_j)c(e_i^{t*}) - Q_p^a + \Theta_p c(e_i^{a*})}_{\text{return to collaborating for less specialized scientist}} \geq \underbrace{\frac{1}{2}C(\gamma_p, \sigma_i, \sigma_j)}_{\text{collaboration costs}} \tag{21}$$

Here, the relevant constraint is on the less specialized scientist: The more specialized scientist's return from working alone is zero.
Are all teams equally likely to occur? Due to lower coordination costs, collaboration between two scientists with low levels of specialization is easier to sustain than between to more specialized scientists if $\partial q(w_i, \sigma_i, e_i)/\partial w_i \partial \sigma_i = 0$, i.e. the gains from the division of labor are the same across all levels of specialization. If $\partial q(w_i, \sigma_i, e_i)/\partial w_i \partial \sigma_i > 0$, which team is easier to sustain for a given problem depends on the relative strength of the increase in coordination costs versus the increase in quality. Assuming that scientists and problems are distributed i.i.d. over the unit interval, a mixed team will be most frequent, since there is a mass of problems that some specialists can only solve in collaboration, and these specialists will only collaborate with less specialized scientists.

Corollary 3: Not all types of collaborations are equally frequent: Specialist Teams are least likely, followed by generalist teams, and mixed teams. Specialists are less likely than generalists to work on a problem on their own.

Solution Quality Consider a case in which constraints 19-21 are fulfilled for a range of σ_i, σ_j pairs. Which team will produce the highest quality solution? Recall that solution quality for teams is:

$$Q = \min_{0 \leq \Theta_p} q(\theta) - C(\gamma_p, \sigma_i, \sigma_j) \tag{22}$$

Since efforts are chosen to produce the exact same quality across all sub-tasks, we can re-write team solution quality as:

$$Q^t = q(w_i^*, \sigma_i, e_i^{t*}) - C(\gamma_p, \sigma_i, \sigma_j) \tag{23}$$

Applying the envelope theorem to this expression:

$$\frac{\partial Q^t}{\partial \sigma_i} = \frac{\partial q}{\partial \sigma_i} - \frac{\partial C}{\partial \sigma_i} \tag{24}$$

yields Prediction 1: A generalist team is best when coordination costs are more elastic to specialization than task quality. Conversely, a reduction in the slope of coordination costs will shrink performance gaps. For Prediction 2, we can differentiate Equation (24)

with respect to γ_p:

$$\frac{\partial^2 Q}{\partial \sigma_i \, \partial \gamma_p} = -\frac{\partial^2 C}{\partial \sigma_i \, \partial \gamma_p} \tag{25}$$

Since coordination costs increase in complexity, i.e. $\partial C / \partial \gamma_p \partial \sigma > 0$ the difference in solution quality between more and less specialized teams also increases in complexity.

B Further Details on Data Construction

B.1 Competitions

Table A1: Kaggle Competiton Examples

Name	Hosted By	Description	Start Date	Duration (Days)	Total Prizes (1000 USD)	Competitors
Panel A: Featured Competitions						
Passenger Screening Algorithm Challenge	Department of Homeland Security	detection and classification of threats from TSA body scans	Jun 22, 2017	176	1500	217
Vesuvius Challenge - Ink Detection	Vesuvius Challenge	detection of ink on x-rays of papyri buried in the eruption of vesuvius	Mar 15, 2023	91	1000	1514
LLM - Detect AI Generated Text	The Learning Agency Lab	classification of essays into student or large language model written	Oct 31, 2023	83	110	5264
RSNA Screening Mammography Breast Cancer Detection	Radiological Society of North America	detection of breast cancers in screening mammogram images	Jan 28, 2022	91	50	2146
Panel B: Research Competitions						
Google - American Sign Language Fingerspelling Recognition	Google	detection and transcription of signed ASL characters from video	May 10, 2023	106	200	1530
BirdCLEF 2023	Cornell Lab of Ornithology	detection and classification of bird calls in audio data	Mar 7, 2023	78	50	1397
IceCube - Neutrinos in Deep Ice	IceCube Neutrino Observatory	reconstruction of the direction of neutrinos from sensor data	Jan 19, 2023	90	50	901
OpenVaccine: COVID-19 mRNA Vaccine Degradation Prediction	Stanford University	prediction of degradation points in mRNA molecules	Sep 11, 2020	25	25	1839

B.2 User Demographics

Coding User Gender I code user gender into two categories: female and male.[27] To do so, I proceed in four steps.

1. If a user reports pronouns, I use pronouns to assign gender (she=female, he=male). Only 1.11% of users report pronouns.

I next use user *display names*, i.e. the name that is shown on a user's profile, to assign gender. After cleaning names from numbers, transcribing non-roman alphabets (mostly Chinese, Japanese, Korean and Russian) as well as removing symbols and emojis, and splitting names into first and last names, I assign gender by:

2. I retrieve information on the most likely gender for each name and country from gender-api.com. I only code a name as male or female if the probability of a name being associated with one gender is higher than 85%. If no country is reported, I assign a gender only if the gender most associated with a name does not change across countries. I can assign gender for 45.73% of users in this step.

3. If a name is not present in gender-api.com and reported at least twice, my research assistant and I try to determine the name's gender by a google search.

4. For the remaining names, which are often nicknames like "computervisionjedi" or "aimuaddib"[28], my research assistant hand-checked the user's profile for additional information such as profile pictures to code user gender.

Overall, I can assign a gender for 79.51% of users in my sample.

Coding User Occupation I use information provided by Kaggle users on their profile page to code occupation. I extract occupations from two fields: Occupation and Bio. I consider both these fields for three reasons: One, occupation is not always filled, two, contextual clues are necessary to disambiguate occupations like "architect", which could both refer to the traditional use of the term – someone who designs and builds houses – as well as the role of a data science professional, and three, some users report occupation aspirations as their occupation, but details in their bio reveal that they are not currently active in that occupation. For example, some users report "machine learning engineer" as their occupation, but write in their bio that they are a college junior. Some users also do not report occupation but their position in a firm's hierarchy, e.g. CEO at startup, but provide more detail in their bios. Table A2 displays chosen occupational categories,

[27] Since users are able to report their preferred pronouns, I originally included a non-binary category. However, no user in my sample reports they/them as preferred pronouns.

[28] For data protection, these are not real Kaggle user names by Kaggle users, but composites from popular references.

as well as the size of a given category and examples for reported occupations within that category. Category “Other” includes both users who do not report occupation information or who report an occupation that is not classifiable within these categories, for example “teacher”, “farmer” or “photographer”.

Table A2: Occupation Categories and Examples

Occupation Category	Observations	%	Modal Occupation	Examples
Academic	125	0.92	Academic at -	Postdoc at University College London Professor in Urban Water Infrastructure at TU Delft
Analyst	243	1.78	Analyst at Deloitte	Analyst at OECD Business Research Analyst-I at Amazon
Artificial Intelligence Professional	160	1.17	AI Engineer	AI Engineer at IBM AI Vision Engineer at Samsung Electronics
Business Professional	202	1.48	Manager at Accenture	Manager at American Express Marketing Manager at Honda Motor Co. Project Manager at JP Morgan
Computer Science Professional	153	1.12	Research Engineer	Computer Systems Engineer at Lawrence Berkeley National Labs Engineering Manager at Adobe
Consultant	183	1.34	Consultant at EY	Analytics Consultant at Deloitte Partner at Mckinsey & Company
Data Analyst	270	1.98	Data Analyst at Home	Aircraft Data Analyst at Airbus Data and Analytics at H&M
Data Engineer	152	1.11	Data Engineer at Accenture	Dat Engineer at Paypal Inc Data Engineer at Google
Data Scientist	2233	16.36	Data Scientist at Freelance	Data Scientist at Uber Datascience at Telecom DS at H2O.Ai
Developer	100	0.73	Developer at Kaggle	Deleloper at Intel Python Developer at Infosys
Engineer	351	2.57	Engineer at JTC	Civil Engineer Engineer at BMW Group
Machine Learning Engineer	713	5.22	ML Engineer	Applied ML at Nvidia Cloud Solution Architect Data and AI at Microsoft Machine Learning Engineer at Apple
Master Student	206	1.51	Master	M.S. Canditate at ETH ZüRich MSBA at Ut Austin
Phd Student	338	2.48	Graduate Student at Carnegie Mellon University	Ph.D. Candidate in Economics at University Of Chicago CS Phd Student at UC Irvine Phd Student at Columbia University
Researcher	380	2.78	Research Scientist	Applied Scientist at Microsoft Computational Scientist at IBM Quantum Data Scientist & Researcher at Nvidia
Software Developer	696	5.10	Software Engineer	Algorithm Engineer at Alibaba SDE at Amazon Software Engineer at Bank Of America
Undergraduate Student	2369	17.36	Student	B.Sc Data Science Student Computer Science Undergraduate
Other	4772	34.97		

Notes: The table displays all occupation categories used as covariates in Equation (8). Spelling of modal occupation and examples correspond to original data.

B.3 Example Notebook

Figure A1: Kaggle Code Notebook Example

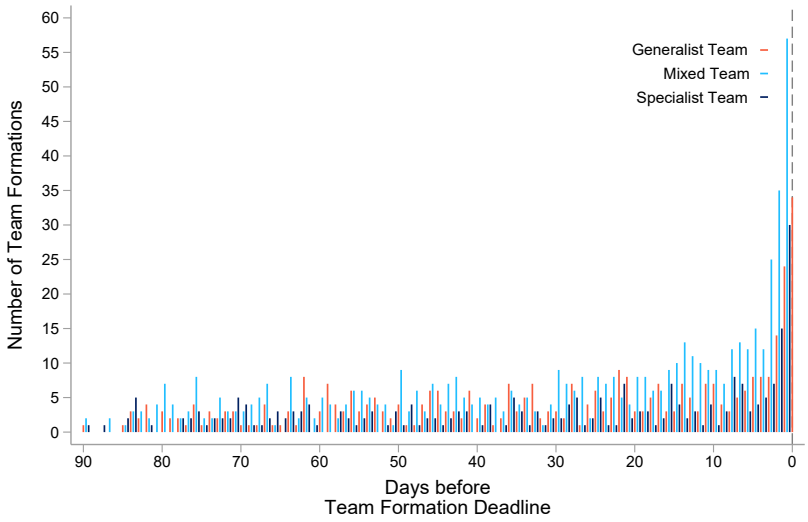


Note: Example adapted from Howard (2024).

C Additional Results for Section 4

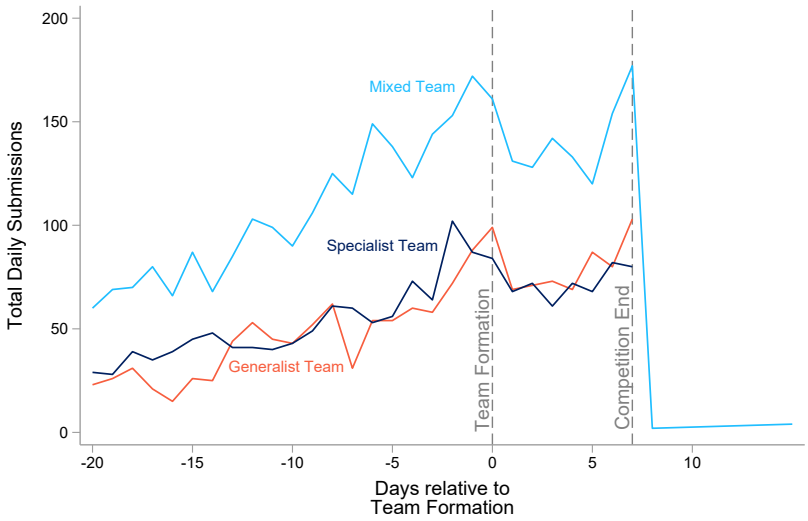
C.1 Additional Results for Section 4.4

Figure A2: Team Formations per Day



Notes: The figure plots the average number of team formations for generalist, mixed and specialist teams in the three months leading up to the team formation deadline. Team formations are zero after the deadline by definition.

Figure A3: Total Daily Submissions



Notes: The figure plots aggregate daily submission counts for generalist, mixed and specialist teams in the three weeks leading up to and the week after the team formation deadline. The dashed gray line indicates the modal competition end date at seven days after the team formation deadline.

Table A3: Event Study Coefficients and F-Tests for Figure 2

Days relative to Team Formation	before				after			
	more than 8	7-8	5-6	3-4	0-1	2-3	4-5	more than 5
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Generalist Team	1.02	3.17	-0.16	3.48	6.98**	8.52*	12.28***	13.08***
	(4.58)	(5.07)	(3.75)	(4.01)	(3.08)	(4.92)	(4.22)	(4.59)
Specialist Team	-2.61	-3.98	-4.68	-5.41	-1.09	-0.59	0.85	2.79
	(5.45)	(6.12)	(5.66)	(4.95)	(3.26)	(3.20)	(3.72)	(3.00)
Mixed Team	-1.93	0.20	0.10	2.43	4.69	5.58*	9.53***	7.36**
	(3.72)	(3.63)	(3.74)	(2.70)	(2.93)	(3.02)	(2.92)	(3.16)
p-value (Generalist Team = Mixed Team)	0.62	0.63	0.96	0.83	0.59	0.61	0.59	0.30
p-value (Generalist Team = Specialist Team)	0.61	0.37	0.50	0.16	0.07	0.12	0.04	0.06
p-value (Specialist Team = Mixed Team)	0.92	0.56	0.48	0.16	0.19	0.16	0.07	0.29
R^2	0.53							
Observations	1,285,813							

Notes: The table reports the estimates of equation (10). The dependent variable is the percentile rank of a user's submission among all submissions to a competition c. Each column presents the percentile change in submission rank percentile relative to those of solo competitors for a time period relative to the period immediately before team formation for each team group. All regressions include user $\times$ team fixed effects and competition day fixed effects. Standard errors are clustered at the team level. Significance levels: $***$ $p<0.01$, $**$ $p<0.05$, and $*$ $p<0.1$.

C.2 Additional Results for Section 4.5

Table A4: Robustness to Alternative Ability Proxies: Performance Lags

Ability Control:	Lag-1 Solo Performance			Lag-1&2 Solo Performance			Lag-1, 2& 3 Solo Performance		
Dependent Variable:	*Rank*	*Medal Win*	*Top 3*	*Rank*	*Medal Win*	*Top 3*	*Rank*	*Medal Win*	*Top 3*
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Generalist Team	20.77***	0.32***	0.06***	21.22***	0.32***	0.06***	20.32***	0.32***	0.06***
	(1.15)	(0.03)	(0.01)	(1.13)	(0.03)	(0.01)	(1.15)	(0.03)	(0.01)
Mixed Team	17.11***	0.26***	0.02**	16.91***	0.26***	0.02***	17.03***	0.27***	0.02***
	(0.93)	(0.02)	(0.01)	(0.93)	(0.02)	(0.01)	(0.94)	(0.02)	(0.01)
Specialist Team	15.63***	0.23***	0.01	15.65***	0.23***	0.01	16.39***	0.26***	0.01
	(1.79)	(0.03)	(0.01)	(1.99)	(0.04)	(0.01)	(2.18)	(0.04)	(0.01)
3-Person Team	6.73***	0.18***	0.04**	6.69***	0.19***	0.04**	6.55***	0.19***	0.04**
	(1.60)	(0.04)	(0.02)	(1.43)	(0.04)	(0.02)	(1.45)	(0.04)	(0.02)
4-Person Team	12.87***	0.30***	0.05	11.14***	0.28***	0.05	10.27***	0.26***	0.05
	(2.13)	(0.06)	(0.03)	(2.17)	(0.06)	(0.03)	(2.19)	(0.06)	(0.03)
5-Person Team	16.82***	0.30***	-0.00	14.88***	0.28***	-0.01	13.28***	0.28***	-0.01
	(1.80)	(0.06)	(0.02)	(1.80)	(0.06)	(0.02)	(1.85)	(0.06)	(0.02)
R^2	0.17	0.12	0.05	0.19	0.14	0.05	0.21	0.16	0.06
Observations	60090	60090	60090	51732	51732	51732	45557	45557	45557
Demographic Controls	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Code Quality Controls	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Competition FEs	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
p-value (Generalist Team = Mixed Team)	0.01	0.09	0.00	0.00	0.10	0.01	0.02	0.18	0.01
p-value (Generalist Team = Specialist Team)	0.01	0.04	0.00	0.01	0.06	0.00	0.10	0.23	0.01
p-value (Specialist Team = Mixed Team)	0.44	0.39	0.51	0.55	0.45	0.57	0.78	0.78	0.72

Notes: The table reports the estimates of equation (8) with alternative ability proxies. The dependent variables are the percentile ranks of a team's solution, an indicator whether a team's solution has won a medal and an indicator whether a team's solution was among the top three solutions in competition c. The main explanatory variables are indicators for the team's type: Generalist Team (a team of only generalists), Mixed Team (a team of both generalists and specialists) and Specialist Team (a team of only specialists). Solo competitors are the omitted category. All regressions include competition experience, measuring the average of the number of competitions each team member participated in before competition c. Ability controls are the average percentile rank achieved by each team member in their last, second to last, and third to last solo competition (lag 1, 2 and 3 solo rank) for dependent variables *Rank* and *Top 3*, or the share of team members who won a medal in their last, second to last, and third to last solo competition (lag 1, 2 and 3 solo rank). Demographic controls include share female, average user platform age, country shares for all countries with at least 100 users, and occupation shares for all occupations reported by at least 100 users (c.f. Table A2). Code quality controls include the average of each team member's lines of code, lines of comment, number of code files, and votes received by code files. Standard errors are clustered at the stable team level. Significance levels: $***$ p<0.01, $**$ p<0.05, and $*$ p<0.1.

Table A5: Robustness to Alternative Ability Proxies: Performance Change

Ability Control:	L1-L2 Solo Rank Change		L2-L3 Solo Rank Change		Average Solo Rank Change	
Dependent Variable:	*Rank*	*Top 3*	*Rank*	*Top 3*	*Rank*	*Top 3*
	(1)	(2)	(3)	(4)	(5)	(6)
Generalist Team	22.78***	0.06***	22.89***	0.06***	22.88***	0.06***
	(1.21)	(0.01)	(1.25)	(0.01)	(1.25)	(0.01)
Mixed Team	18.52***	0.02***	18.73***	0.02***	18.72***	0.02***
	(0.99)	(0.01)	(1.02)	(0.01)	(1.02)	(0.01)
Specialist Team	16.09***	0.01	16.84***	0.01	16.82***	0.01
	(1.95)	(0.01)	(2.03)	(0.01)	(2.03)	(0.01)
3-Person Team	6.97***	0.04**	6.75***	0.04**	6.77***	0.04**
	(1.49)	(0.02)	(1.53)	(0.02)	(1.53)	(0.02)
4-Person Team	12.30***	0.05	11.36***	0.05	11.37***	0.05
	(2.08)	(0.03)	(2.12)	(0.03)	(2.12)	(0.03)
5-Person Team	15.69***	-0.01	14.80***	-0.01	14.83***	-0.01
	(1.79)	(0.02)	(1.86)	(0.02)	(1.86)	(0.02)
R^2	0.11	0.05	0.11	0.06	0.11	0.06
Observations	51732	51732	45557	45557	45557	45557
Demographic Controls	Yes	Yes	Yes	Yes	Yes	Yes
Code Quality Controls	Yes	Yes	Yes	Yes	Yes	Yes
Competition FEs	Yes	Yes	Yes	Yes	Yes	Yes
p-value (Generalist Team = Mixed Team)	0.00	0.01	0.00	0.01	0.00	0.01
p-value (Generalist Team = Specialist Team)	0.00	0.00	0.01	0.01	0.01	0.01
p-value (Specialist Team = Mixed Team)	0.24	0.56	0.38	0.70	0.38	0.70

Notes: The table reports the estimates of equation (8) with alternative ability proxies. The dependent variables are the percentile rank of a team's solution and an indicator whether a team's solution was among the top three solutions in competition c. The main explanatory variables are indicators for the team's type: Generalist Team (a team of only generalists), Mixed Team (a team of both generalists and specialists) and Specialist Team (a team of only specialists). Solo competitors are the omitted category. All regressions include competition experience, measuring the average of the number of competitions each team member participated in before competition c. Ability controls is the team-level average of the change in solo ranks between the last and second to last, between second to last and third to last, or their average. Demographic controls include share female, average user platform age, country shares for all countries with at least 100 users, and occupation shares for all occupations reported by at least 100 users (c.f. Table A2). Code quality controls include the average of each team member's lines of code, lines of comment, number of code files, and votes received by code files. Standard errors are clustered at the stable team level. Significance levels: $***$ p<0.01, $**$ p<0.05, and $*$ p<0.1.

Table A6: Robustness to Different Specialization Indices

Specialization Index:	Excluding Post-ChatGPT Code			Python Package Diversity		
Dependent Variable:	*Rank*	*Medal Win*	*Top 3*	*Rank*	*Medal Win*	*Top 3*
	(1)	(2)	(3)	(4)	(5)	(6)
Generalist Team	19.50***	0.32***	0.05***	18.12***	0.27***	0.03***
	(1.25)	(0.03)	(0.01)	(0.82)	(0.02)	(0.01)
Mixed Team	16.72***	0.26***	0.02***	17.03***	0.27***	0.03***
	(0.91)	(0.02)	(0.01)	(1.18)	(0.02)	(0.01)
Specialist Team	18.95***	0.24***	0.01	9.47**	0.15**	0.02
	(1.72)	(0.03)	(0.01)	(4.07)	(0.07)	(0.02)
Competition Experience	0.08***	-0.00***	-0.00***	0.08***	-0.00***	-0.00***
	(0.01)	(0.00)	(0.00)	(0.01)	(0.00)	(0.00)
Previous Rank	0.50***		0.00***	0.49***		0.00***
	(0.01)		(0.00)	(0.01)		(0.00)
Previous Medal Wins		0.39***			0.39***	
		(0.01)			(0.01)	
3-Person Team	6.07***	0.18***	0.04**	5.91***	0.17***	0.04**
	(1.64)	(0.04)	(0.02)	(1.62)	(0.04)	(0.02)
4-Person Team	12.36***	0.29***	0.05	11.51***	0.28***	0.04
	(2.21)	(0.06)	(0.03)	(2.17)	(0.06)	(0.03)
5-Person Team	14.55***	0.29***	-0.01	13.64***	0.27***	-0.02
	(1.80)	(0.06)	(0.02)	(1.75)	(0.06)	(0.02)
R^2	0.21	0.16	0.05	0.20	0.15	0.05
Observations	56183	56183	56183	59047	59047	59047
Demographic Controls	Yes	Yes	Yes	Yes	Yes	Yes
Code Quality Controls	Yes	Yes	Yes	Yes	Yes	Yes
Competition FEs	Yes	Yes	Yes	Yes	Yes	Yes
p-value (Generalist Team = Mixed Team)	0.05	0.06	0.07	0.39	1.00	0.94
p-value (Generalist Team = Specialist Team)	0.79	0.06	0.03	0.04	0.09	0.57
p-value (Specialist Team = Mixed Team)	0.23	0.63	0.44	0.07	0.10	0.56

Notes: The table reports the estimates of equation (8) with alternative ways of classifying specialists and generalists. Columns 1-3 use the specialization index as described in Section 3.3, excluding all code files published after the introduction of ChatGPT in November 2022 from the code base. Columns 4-6 use a measure of code diversity based on used python packages (c.f. section 4.5). The dependent variable is the percentile rank of a team's solution quality, an indicator whether a team's solution has won a medal or an indicator whether a team's solution was among the top three solutions in competition c. Standard errors are clustered at the stable team level. Significance levels: $***$ p<0.01, $**$ p<0.05, and $*$ p<0.1.

D Additional Results for Section 5

D.1 Additional Results for Section 5.1

Table A7: Team Type, Complexity and Solution Quality

Dependent Variable:	*Rank*		*Medal Win*		*Top 3*	
Complexity:	Low	High	Low	High	Low	High
	(1)	(2)	(3)	(4)	(5)	(6)
Generalist Team	20.49***	18.34***	0.33***	0.25***	0.06***	0.05**
	(1.20)	(2.16)	(0.03)	(0.05)	(0.02)	(0.02)
Mixed Team	17.19***	14.52***	0.27***	0.20***	0.01*	0.02*
	(1.03)	(1.96)	(0.02)	(0.03)	(0.01)	(0.01)
Specialist Team	18.59***	8.52**	0.27***	0.09*	0.02	-0.00
	(2.13)	(3.76)	(0.04)	(0.05)	(0.01)	(0.00)
Competition Experience	0.10***	0.03**	-0.00***	-0.00***	-0.00***	-0.00***
	(0.01)	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)
Previous Rank	0.55***	0.34***			0.00***	0.00***
	(0.01)	(0.01)			(0.00)	(0.00)
Previous Medal Wins			0.42***	0.29***		
			(0.01)	(0.02)		
3-Person Team	5.81***	6.99**	0.20***	0.11*	0.06**	0.01
	(1.67)	(3.08)	(0.04)	(0.06)	(0.02)	(0.03)
4-Person Team	13.04***	7.82	0.32***	0.13	0.06*	-0.02**
	(1.62)	(9.75)	(0.06)	(0.15)	(0.04)	(0.01)
5-Person Team	14.30***	16.71***	0.34***	0.23**	-0.02***	0.02
	(2.04)	(3.39)	(0.06)	(0.11)	(0.01)	(0.04)
R^2	0.25	0.11	0.16	0.13	0.06	0.04
Observations	43684	16406	43684	16406	43684	16406
Demographic Controls	Yes	Yes	Yes	Yes	Yes	Yes
Code Quality Controls	Yes	Yes	Yes	Yes	Yes	Yes
Competition FEs	Yes	Yes	Yes	Yes	Yes	Yes
p-value (Generalist Team = Mixed Team)	0.02	0.16	0.08	0.38	0.01	0.20
p-value (Generalist Team = Specialist Team)	0.42	0.02	0.18	0.02	0.02	0.01
p-value (Specialist Team = Mixed Team)	0.54	0.15	0.97	0.06	0.86	0.02

Notes: The table reports the estimates of equation (8), separately for the sample of low and high complexity competitions. The dependent variable is the percentile rank of a team's solution quality, an indicator whether a team's solution has won a medal or an indicator whether a team's solution was among the top three solutions in competition c. The main explanatory variables are indicators for the team's type: Generalist Team (a team of only generalists), Mixed Team (a team of both generalists and specialists) and Specialist Team (a team of only specialists). Solo competitors are the omitted category. Significance levels: $***$ p<0.01, $**$ p<0.05, and $*$ p<0.1.

Table A8: Robustness to Complexity Measure: Cutoff at 50th percentile

Dependent Variable:	*Rank*		*Medal Win*		*Top 3*	
Complexity (Cutoff at 50th percentile):	Low	High	Low	High	Low	High
	(1)	(2)	(3)	(4)	(5)	(6)
Generalist Team	21.24***	19.14***	0.36***	0.27***	0.08***	0.04***
	(1.50)	(1.42)	(0.04)	(0.03)	(0.02)	(0.01)
Mixed Team	17.80***	15.75***	0.29***	0.23***	0.01	0.02***
	(1.34)	(1.22)	(0.03)	(0.02)	(0.01)	(0.01)
Specialist Team	20.17***	13.58***	0.29***	0.18***	0.00	0.01
	(2.59)	(2.44)	(0.05)	(0.04)	(0.01)	(0.01)
Competition Experience	0.08***	0.07***	-0.00***	-0.00**	-0.00***	-0.00***
	(0.01)	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)
Previous Rank	0.56***	0.45***			0.00***	0.00***
	(0.01)	(0.01)			(0.00)	(0.00)
Previous Medal Wins			0.45***	0.33***		
			(0.02)	(0.01)		
3-Person Team	5.84***	6.04***	0.19***	0.16***	0.07**	0.02
	(1.83)	(2.32)	(0.05)	(0.05)	(0.03)	(0.02)
4-Person Team	11.52***	13.45***	0.28***	0.31***	0.02	0.07
	(2.31)	(3.32)	(0.08)	(0.08)	(0.04)	(0.05)
5-Person Team	16.37***	13.75***	0.45***	0.19**	-0.02*	0.00
	(1.95)	(2.53)	(0.06)	(0.08)	(0.01)	(0.03)
R^2	0.25	0.17	0.19	0.13	0.06	0.04
Observations	25942	34148	25942	34148	25942	34148
Demographic Controls	Yes	Yes	Yes	Yes	Yes	Yes
Code Quality Controls	Yes	Yes	Yes	Yes	Yes	Yes
Competition FEs	Yes	Yes	Yes	Yes	Yes	Yes
p-value (Generalist Team = Mixed Team)	0.06	0.05	0.09	0.28	0.01	0.13
p-value (Generalist Team = Specialist Team)	0.72	0.04	0.27	0.07	0.00	0.10
p-value (Specialist Team = Mixed Team)	0.40	0.41	0.97	0.30	0.62	0.65

Notes: The table reports the estimates of equation (8), separately for the sample of low and high complexity competitions. The dependent variable is the percentile rank of a team's solution quality, an indicator whether a team's solution has won a medal or an indicator whether a team's solution was among the top three solutions in competition c. The main explanatory variables are indicators for the team's type: Generalist Team (a team of only generalists), Mixed Team (a team of both generalists and specialists) and Specialist Team (a team of only specialists). Solo competitors are the omitted category. Significance levels: $***$ p<0.01, $**$ p<0.05, and $*$ p<0.1.

Table A9: Robustness to Complexity Measure: Cutoff at 90th percentile

Dependent Variable:	*Rank*		*Medal Win*		*Top 3*	
Complexity (Cutoff at 90th percentile):	Low	High	Low	High	Low	High
	(1)	(2)	(3)	(4)	(5)	(6)
Generalist Team	19.74***	21.01***	0.30***	0.30***	0.06***	0.04
	(1.12)	(3.42)	(0.03)	(0.07)	(0.01)	(0.03)
Mixed Team	17.27***	10.73***	0.27***	0.11**	0.01**	0.03
	(0.93)	(3.52)	(0.02)	(0.05)	(0.01)	(0.02)
Specialist Team	17.31***	5.21	0.25***	-0.05	0.01	0.00
	(2.11)	(5.47)	(0.03)	(0.05)	(0.01)	(0.00)
Competition Experience	0.09***	-0.02	-0.00***	-0.00**	-0.00***	-0.00***
	(0.01)	(0.02)	(0.00)	(0.00)	(0.00)	(0.00)
Previous Rank	0.53***	0.25***			0.00***	0.00***
	(0.01)	(0.02)			(0.00)	(0.00)
Previous Medal Wins			0.40***	0.30***		
			(0.01)	(0.03)		
3-Person Team	6.69***	1.88	0.19***	0.05	0.05**	-0.04**
	(1.50)	(5.55)	(0.04)	(0.09)	(0.02)	(0.02)
4-Person Team	12.47***	9.36	0.30***	0.22	0.05	-0.03
	(1.90)	(15.12)	(0.06)	(0.20)	(0.03)	(0.02)
5-Person Team	13.86***	20.54**	0.28***	0.32	-0.00	-0.04*
	(1.74)	(8.90)	(0.06)	(0.21)	(0.02)	(0.02)
R^2	0.23	0.08	0.16	0.13	0.05	0.03
Observations	52540	7550	52540	7550	52540	7550
Demographic Controls	Yes	Yes	Yes	Yes	Yes	Yes
Code Quality Controls	Yes	Yes	Yes	Yes	Yes	Yes
Competition FEs	Yes	Yes	Yes	Yes	Yes	Yes
p-value (Generalist Team = Mixed Team)	0.06	0.03	0.27	0.03	0.00	0.71
p-value (Generalist Team = Specialist Team)	0.30	0.01	0.23	0.00	0.00	0.16
p-value (Specialist Team = Mixed Team)	0.99	0.38	0.68	0.02	0.76	0.13

Notes: The table reports the estimates of equation (8), separately for the sample of low and high complexity competitions. The dependent variable is the percentile rank of a team's solution quality, an indicator whether a team's solution has won a medal or an indicator whether a team's solution was among the top three solutions in competition c. The main explanatory variables are indicators for the team's type: Generalist Team (a team of only generalists), Mixed Team (a team of both generalists and specialists) and Specialist Team (a team of only specialists). Solo competitors are the omitted category. Significance levels: $***$ p<0.01, $**$ p<0.05, and $*$ p<0.1.

Table A10: Robustness to Alternative Complexity Measure: Competition Difficulty

Dependent Variable:	*Rank*		*Medal Win*		*Top 3*	
Difficulty:	Low	High	Low	High	Low	High
	(1)	(2)	(3)	(4)	(5)	(6)
Generalist Team	19.51***	21.24***	0.32***	0.27***	0.07***	0.04*
	(1.26)	(2.09)	(0.03)	(0.05)	(0.02)	(0.02)
Mixed Team	17.62***	13.49***	0.27***	0.19***	0.01*	0.03**
	(1.03)	(2.00)	(0.02)	(0.04)	(0.01)	(0.01)
Specialist Team	17.67***	10.41***	0.25***	0.12**	0.01	-0.01
	(2.19)	(3.59)	(0.04)	(0.05)	(0.01)	(0.00)
Competition Experience	0.09***	0.04***	-0.00***	-0.00***	-0.00***	-0.00***
	(0.01)	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)
Previous Rank	0.53***	0.39***			0.00***	0.00***
	(0.01)	(0.01)			(0.00)	(0.00)
Previous Medal Wins			0.42***	0.31***		
			(0.01)	(0.02)		
3-Person Team	5.68***	6.58**	0.18***	0.15**	0.05**	0.02
	(1.62)	(3.30)	(0.04)	(0.07)	(0.02)	(0.03)
4-Person Team	11.58***	14.29**	0.28***	0.34***	0.05	0.06
	(2.24)	(6.38)	(0.06)	(0.12)	(0.03)	(0.08)
5-Person Team	16.22***	12.57***	0.35***	0.19	0.00	-0.03**
	(1.49)	(4.18)	(0.06)	(0.12)	(0.02)	(0.01)
R^2	0.23	0.14	0.16	0.12	0.05	0.04
Observations	44944	15146	44944	15146	44944	15146
Demographic Controls	Yes	Yes	Yes	Yes	Yes	Yes
Code Quality Controls	Yes	Yes	Yes	Yes	Yes	Yes
Competition FEs	Yes	Yes	Yes	Yes	Yes	Yes
p-value (Generalist Team = Mixed Team)	0.21	0.00	0.15	0.18	0.00	0.60
p-value (Generalist Team = Specialist Team)	0.46	0.01	0.14	0.04	0.01	0.03
p-value (Specialist Team = Mixed Team)	0.98	0.45	0.63	0.27	0.87	0.01

Notes: The table reports the estimates of equation (8), separately for the sample of low and high complexity competitions. The dependent variable is the percentile rank of a team's solution quality, an indicator whether a team's solution has won a medal or an indicator whether a team's solution was among the top three solutions in competition c. The main explanatory variables are indicators for the team's type: Generalist Team (a team of only generalists), Mixed Team (a team of both generalists and specialists) and Specialist Team (a team of only specialists). Solo competitors are the omitted category. Significance levels: $***$ p<0.01, $**$ p<0.05, and $*$ p<0.1.

Table A11: Robustness to Alternative Complexity Measure: Instruction Length

Dependent Variable:	*Rank*		*Medal Win*		*Top 3*	
Instruction Length:	Short	Long	Short	Long	Short	Long
	(1)	(2)	(3)	(4)	(5)	(6)
Generalist Team	20.92***	19.53***	0.33***	0.29***	0.06***	0.06***
	(1.78)	(1.37)	(0.04)	(0.03)	(0.02)	(0.02)
Mixed Team	17.84***	16.17***	0.29***	0.23***	0.01	0.02**
	(1.49)	(1.13)	(0.03)	(0.02)	(0.01)	(0.01)
Specialist Team	21.90***	13.41***	0.33***	0.17***	-0.00	0.01
	(2.61)	(2.27)	(0.06)	(0.04)	(0.00)	(0.01)
Competition Experience	0.10***	0.07***	-0.00	-0.00***	-0.00***	-0.00***
	(0.01)	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)
Previous Rank	0.50***	0.49***			0.00***	0.00***
	(0.01)	(0.01)			(0.00)	(0.00)
Previous Medal Wins			0.39***	0.38***		
			(0.02)	(0.01)		
3-Person Team	2.54	7.54***	0.14**	0.20***	0.04	0.04*
	(3.00)	(1.70)	(0.06)	(0.04)	(0.03)	(0.02)
4-Person Team	9.48***	13.52***	0.24**	0.32***	0.06	0.05
	(2.58)	(2.61)	(0.12)	(0.06)	(0.07)	(0.03)
5-Person Team	13.85***	15.53***	0.41***	0.29***	-0.02	-0.00
	(3.49)	(1.97)	(0.14)	(0.06)	(0.01)	(0.02)
R^2	0.23	0.18	0.16	0.14	0.06	0.04
Observations	24558	35532	24558	35532	24558	35532
Demographic Controls	Yes	Yes	Yes	Yes	Yes	Yes
Code Quality Controls	Yes	Yes	Yes	Yes	Yes	Yes
Competition FEs	Yes	Yes	Yes	Yes	Yes	Yes
p-value (Generalist Team = Mixed Team)	0.16	0.04	0.47	0.11	0.05	0.03
p-value (Generalist Team = Specialist Team)	0.75	0.02	0.94	0.01	0.00	0.03
p-value (Specialist Team = Mixed Team)	0.17	0.25	0.54	0.14	0.16	0.83

Notes: The table reports the estimates of equation (8), separately for the sample of competitions with short (below median) and long instructions (above median instruction length). Competition instructions are extracted from the overview page of each competition on Kaggle (competition description). The dependent variable is the percentile rank of a team's solution quality, an indicator whether a team's solution has won a medal or an indicator whether a team's solution was among the top three solutions in competition c. The main explanatory variables are indicators for the team's type: Generalist Team (a team of only generalists), Mixed Team (a team of both generalists and specialists) and Specialist Team (a team of only specialists). Solo competitors are the omitted category. Significance levels: $***$ p<0.01, $**$ p<0.05, and $*$ p<0.1.

Table A12: Robustness to Alternative Complexity Measure: Compute Constraints

Dependent Variable:	*Rank*		*Medal Win*		*Top 3*	
Compute Constraints:	No	Yes	No	Yes	No	Yes
	(1)	(2)	(3)	(4)	(5)	(6)
Generalist Team	20.15***	19.93***	0.32***	0.29***	0.05***	0.07***
	(1.39)	(1.58)	(0.03)	(0.03)	(0.02)	(0.02)
Mixed Team	14.27***	18.41***	0.24***	0.26***	0.01	0.02***
	(1.31)	(1.18)	(0.03)	(0.03)	(0.01)	(0.01)
Specialist Team	19.84***	12.68***	0.29***	0.16***	0.02	0.00
	(2.10)	(2.80)	(0.05)	(0.04)	(0.02)	(0.01)
Competition Experience	0.07***	0.09***	-0.00**	-0.00***	-0.00***	-0.00***
	(0.01)	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)
Previous Rank	0.51***	0.47***			0.00***	0.00***
	(0.01)	(0.01)			(0.00)	(0.00)
Previous Medal Wins			0.40***	0.37***		
			(0.01)	(0.02)		
3-Person Team	3.68*	7.56***	0.15***	0.20***	0.07**	0.03
	(2.19)	(1.94)	(0.05)	(0.04)	(0.03)	(0.02)
4-Person Team	9.06**	13.73***	0.26***	0.31***	0.03	0.06
	(4.16)	(2.35)	(0.10)	(0.07)	(0.04)	(0.04)
5-Person Team	12.07***	15.21***	0.27**	0.30***	-0.02**	-0.00
	(3.82)	(1.85)	(0.10)	(0.07)	(0.01)	(0.02)
R^2	0.23	0.17	0.17	0.13	0.06	0.04
Observations	31588	28502	31588	28502	31588	28502
Demographic Controls	Yes	Yes	Yes	Yes	Yes	Yes
Code Quality Controls	Yes	Yes	Yes	Yes	Yes	Yes
Competition FEs	Yes	Yes	Yes	Yes	Yes	Yes
p-value (Generalist Team = Mixed Team)	0.00	0.40	0.07	0.39	0.09	0.01
p-value (Generalist Team = Specialist Team)	0.90	0.02	0.68	0.01	0.29	0.00
p-value (Specialist Team = Mixed Team)	0.02	0.05	0.35	0.03	0.57	0.06

Notes: The table reports the estimates of equation (8), separately for the sample of low and high complexity competitions, where high complexity competitions are those with compute constraints. The dependent variable is the percentile rank of a team's solution quality, an indicator whether a team's solution has won a medal or an indicator whether a team's solution was among the top three solutions in competition c. The main explanatory variables are indicators for the team's type: Generalist Team (a team of only generalists), Mixed Team (a team of both generalists and specialists) and Specialist Team (a team of only specialists). Solo competitors are the omitted category. Significance levels: *** p<0.01, ** p<0.05, and * p<0.1.

D.2 Additional Results for Section 5.2

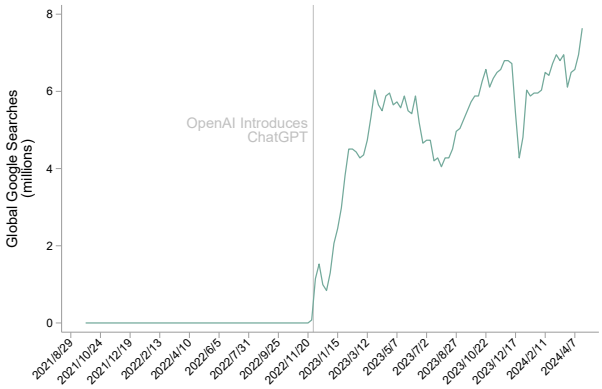
Table A13: Team Type, ChatGPT and Solution Quality

Dependent Variable:	*Rank*		*Medal Win*		*Top 3*	
ChatGPT:	Pre	Post	Pre	Post	Pre	Post
	(1)	(2)	(3)	(4)	(5)	(6)
Generalist Team	21.60***	15.26***	0.34***	0.34***	0.04*	0.05
	(2.45)	(3.13)	(0.06)	(0.06)	(0.02)	(0.04)
Mixed Team	18.11***	14.72***	0.28***	0.22***	0.03*	0.01
	(2.50)	(2.38)	(0.05)	(0.05)	(0.02)	(0.02)
Specialist Team	10.09***	17.26***	0.05	0.24***	0.02	-0.01***
	(3.91)	(3.99)	(0.07)	(0.06)	(0.03)	(0.00)
Competition Experience	0.09***	0.05***	-0.00	-0.00***	-0.00***	-0.00***
	(0.01)	(0.02)	(0.00)	(0.00)	(0.00)	(0.00)
Previous Rank	0.51***	0.48***			0.00*	0.00
	(0.02)	(0.02)			(0.00)	(0.00)
Previous Medal Wins			0.38***	0.37***		
			(0.03)	(0.03)		
3-Person Team	2.50	12.86***	0.10	0.29***	0.06	0.04
	(3.35)	(3.65)	(0.07)	(0.08)	(0.04)	(0.04)
4-Person Team	17.23***	15.78***	0.40***	0.25*	0.07	0.08
	(2.98)	(4.95)	(0.10)	(0.14)	(0.11)	(0.11)
5-Person Team	17.25***	15.56***	0.31***	0.42***	-0.03**	-0.02*
	(3.54)	(5.98)	(0.10)	(0.15)	(0.02)	(0.01)
R^2	0.20	0.16	0.15	0.11	0.05	0.04
Observations	8093	9946	8093	9946	8093	9946
Demographic Controls	Yes	Yes	Yes	Yes	Yes	Yes
Code Quality Controls	Yes	Yes	Yes	Yes	Yes	Yes
Competition FEs	Yes	Yes	Yes	Yes	Yes	Yes
p-value (Generalist Team = Mixed Team)	0.29	0.88	0.42	0.11	0.76	0.35
p-value (Generalist Team = Specialist Team)	0.01	0.69	0.00	0.29	0.51	0.10
p-value (Specialist Team = Mixed Team)	0.06	0.57	0.01	0.76	0.64	0.14

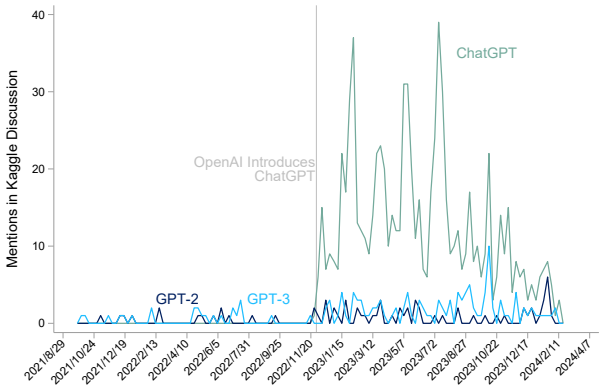
Notes: The table reports the estimates of equation (8), separately for the sample of competitions before and after the introduction of ChatGPT. The dependent variable is the percentile rank of a team's solution quality, an indicator whether a team's solution has won a medal or an indicator whether a team's solution was among the top three solutions in competition c. The main explanatory variables are indicators for the team's type: Generalist Team (a team of only generalists), Mixed Team (a team of both generalists and specialists) and Specialist Team (a team of only specialists). Solo competitors are the omitted category. Standard errors are clustered at the level of the stable team. Significance levels: $***$ p<0.01, $**$ p<0.05, and $*$ p<0.1.

Figure A4: Google Searches and Kaggle Forum Mentions of ChatGPT

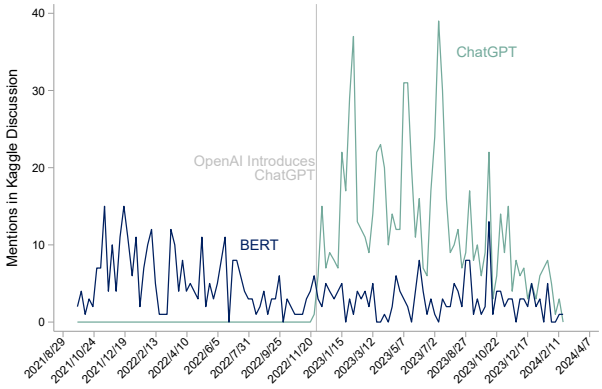
(a) Google Searches of ChatGPT



(b) Forum Mentions of ChatGPT, GPT-2 and GPT-3



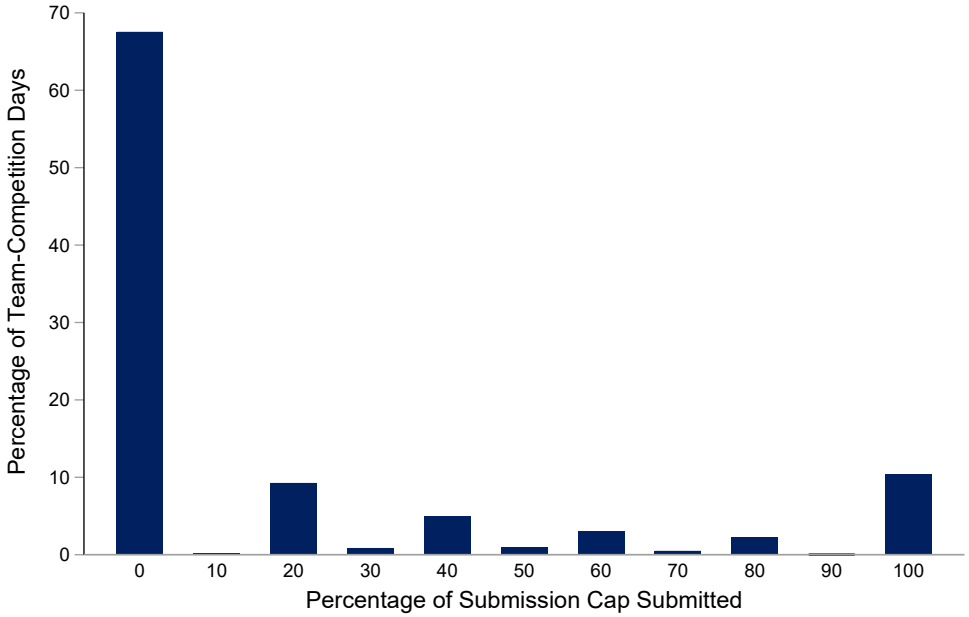
(c) Forum Mentions of ChatGPT and BERT



Notes: Panel (a) plots google searches of the term "ChatGPT" for the time period between September 2021 and July 2024. Data come from Google Trends, converted to search counts using Glimpse-Google Trends Supercharged Plugin. Panel (b) plots mentions of the term "ChatGPT" compared to "GPT-2" and "GPT-3" and Panel (c) compared to "BERT" in public Kaggle forums for the time period between September 2021 and July 2024.

D.3 Additional Results for Section 5.3

Figure A5: Percentage of Daily Submission Caps



Notes: The figure plots how frequently submission caps are exhausted to which extent. Each category on the x-axis corresponds to a share of allowed submissions, i.e. 0 indicates that no submissions were made on a given day, and 100 that all possible submissions were made on a given day. The y-axis shows the share of team-competition days a submission share was reached. Only on circa 10% of days, teams reach the submission cap. Note that this includes competitions in which the daily submission cap is one submission.