Mitigating Spoken Language Barriers in AI-Assisted Programming: Evidence from a Field Experiment

Rohit Aggarwal
David Eccles School of Business, University of Utah, Salt Lake City, Utah, United States 84112
Email: rohit.aggarwal@eccles.utah.edu

Raveesh Mayya
Leonard N. Stern School of Business, New York University, New York, New York, United States 10012
Email: raveesh@stern.nyu.edu

Harpreet Singh
Naveen Jindal School of Management, University of Texas at Dallas, Richardson, Texas, United States 75080
Email: harpreet@utdallas.edu

Draft Date: April 27th, 2025
(Click Here for the Latest Version)

Programming-specific Generative Artificial Intelligence (GenAI) tools have already been shown to significantly enhance programmer productivity. However, given that most software programming syntax, documentation and discussions are in English, the language barrier may often keep non-native English speakers from effectively using GenAI, potentially exacerbating the digital divide. This issue is particularly acute for programmers whose native languages are classified as low-resource languages (LRLs)—languages with limited digital presence or scarce annotated datasets for model training. As a result, programmers with LRLs as the primary language often spend excessive time crafting instructions in English, experience heightened frustration, and ultimately abandon these valuable AI tools. To address this issue, we examine "Metadata Prompting" as an intermediate representation between natural language and programming instructions, which provides a structured framework to express program requirements, thereby minimizing English grammatical structures. By changing from a conversational language approach to a programmatic approach, which programmers understand well, Metadata Prompting minimizes cognitive load and enhances clarity, enabling non-native English speakers to convey ideas effectively. By running a field experiment in India, we quantify the effectiveness of Metadata prompting: programmers who are not proficient in English are more likely to use GenAI as well as match the success rates and efficiency level of programmers who are proficient in English. Our finding highlights that by incorporating simple changes in prompting, GenAI platforms can boost their usage while simultaneously fostering an inclusive environment. Our findings have implications for policymakers, GenAI platforms, and gig economy platforms.

Key words: Generative AI, Productivity, Large Language Models, Metadata Prompting, Field Experiment

---

[0] Author names are listed in the alphabetical order.

1. Introduction

This decade has witnessed a meteoric rise in the capabilities of Generative Artificial Intelligence (GenAI) tools, driving significant productivity gains across diverse fields. From customer support (Brynjolfsson et al. 2023), creative writing (Noy and Zhang 2023, Doshi and Hauser 2024, Chen and Chan 2024, Qiu et al. 2023) to artistic expression (Zhou and Lee 2024, Boussioux et al. 2024) and knowledge dissemination (Burtch et al. 2024, Horton 2023, Quinn and Gutt 2025), GenAI is proving its versatility, while simultaneously reshaping labor markets (Liu et al. 2023a).

Software development stands out as a prime example where GenAI is making significant inroads, enhancing productivity in tasks ranging from code generation and debugging to documentation and testing. As these AI tools evolve, researchers are developing methodological frameworks to guide their effective design and implementation (Abbasi et al. 2024), their use as autonomous agents that assist programmers (Ni et al. 2025), or their application as effective annotation assistants (Cheng et al. 2024, Pangakis et al. 2023, de Kok 2025). In fact, these tools are proving to be effective in both unguided programming settings (Yeverechyahu et al. 2024, Song et al. 2023) and guided programming settings (Eloundou et al. 2024, Peng et al. 2023, Cui et al. 2024). As GenAI models continue to evolve and better understand context, their impact on programming productivity is only expected to grow (Yeverechyahu et al. 2024).

Whereas the growth in the use of GenAI in software development is expected to continue, a critical challenge lies in the spoken-linguistic diversity of the global developer community. Especially over the last two decades, there has been a surge in programming contributions from Asian countries,[1] where English is not the primary language. Extant studies have already documented the linguistic barriers that hinder non-native English speakers from excelling in programming (Guo 2018). Along with learning the logic and syntax, the cognitive burden of understanding the documentation, code examples, or even the intuition behind naming certain commands and procedures, is high on non-native English speakers (Fedorenko et al. 2019). Given the barriers to effective

[1] The Octoverse Report: 2024 - GitHub

English usage (e.g., Noll et al. 2011), a common practice involves developers switching to native languages while ideating, discussing requirements or brainstorming solutions.

This linguistic challenge is magnified for developers whose native languages are classified as low-resource languages (LRLs). Compared to high-resource languages such as English, Mandarin, or Spanish, LRLs such as Hindi are characterized by limited digital presence (Conneau et al. 2020) and scarce annotated datasets for training NLP models (Zoph et al. 2016, Adams et al. 2017). Whereas some non-English languages like Mandarin have extensive digital content repositories (with platforms like Baidu Baike containing over 25 million articles, rivaling or exceeding English-language resources), many languages from South and Southeast Asia remain severely underrepresented digitally. Many of these LRLs also typically have high information density per word compared to Latin and Cyrillic languages, leading to inefficient tokenization when processed by LLMs (Shi et al. 2023). This "token fertility" issue—where more tokens are needed to encode equivalent information compared to English—creates an additional technological barrier for efficient use of LRLs in training a high quality multilingual LLMs (Ahuja et al. 2023) as well as the cost of inference (Ahia et al. 2023).

The optimization of GenAI programming assistants primarily around high-resource languages (e.g., Joshi et al. 2020) further exacerbates these linguistic disparities. Beyond the inherent challenges of LRLs, the technical architecture of these models creates additional barriers. The tokenization methods employed by LLMs—often optimized for English and similar languages—struggle with the higher information density found in many South and Southeast Asian languages. For programming contexts specifically, the scarcity of technical documentation and resources in LRLs further exacerbates these challenges, as LLMs struggle to map requirements specified in LRLs to accurate code implementations (Zheng et al. 2023). This problem is magnified because the authoritative resources—textbooks, programming language documentation, and open-source project discussions—predominantly exist in high-resource languages, creating a knowledge gap that LLMs cannot easily bridge when processing requirements in LRLs.

Consequently, this confluence of factors—the inherent language bias in GenAI and the existing challenges for non-native English speakers—has a potential to widen the digital divide in the programming world (Fu et al. 2020)—not through access disparities but through effective utilization barriers tied to language proficiency. The cognitive burden to translate their thoughts and requirements into English (e.g., Neeley 2013) for interaction with AI-powered development tools can introduce two significant challenges. The first one is regarding errors: even minor errors in word choice or phrasing can lead AI models astray, resulting in misinterpretations and inaccurate code generation. An inability to articulate complex technical concepts precisely may lead AI to produce irrelevant code or misunderstand the intended system architecture. The second one is regarding the reluctance of GenAI usage: non-English speaking developers may hesitate to utilize AI tools because of a lack of confidence in their ability to express requirements in English. Both of these challenges create a situation where English-proficient developers gain a significant advantage in utilizing cutting-edge AI assistants, even as the majority of emerging contributors hail from non-native English speaking countries, potentially disrupting collaborations (Neeley et al. 2009).

In this paper, we first investigate how English language proficiency among developers affects their ability to leverage GenAI for programming tasks efficiently. Despite substantial evidence documenting language barriers in global collaborations (Hinds et al. 2014) and digital tool adoption (Neeley et al. 2009), empirical research quantifying how these barriers affect GenAI usage in programming remains limited. This gap is particularly concerning given the rapid adoption of GenAI tools in software development and the increasing globalization of the developer community. This leads to our first research question: *"How does English language proficiency affect developers' ability to effectively utilize GenAI tools for software development tasks?"*

We then examine whether structured intermediate representations can mitigate these language barriers. While recent advances in GenAI research have demonstrated the effectiveness of controlled intermediate languages that serve as a bridge between human expression and formal systems (Kesseli et al. 2025, Lehmann et al. 2023), and studies have shown how structured approaches can

reduce learning barriers in AI interactions (Ma et al. 2024), their potential to bridge language-based performance gaps remains unexplored, particularly in contexts where developers' domain expertise might compensate for linguistic limitations. This leads to our second research question: *"To what extent can structured intermediate representations mitigate the impact of limited English proficiency on developers' ability to effectively utilize GenAI tools?"*

Finally, we explore developers' preferences regarding prompting techniques. Existing adoption research suggests that perceived ease of use significantly influences technology acceptance (Davis 1989), yet little is known about how language proficiency moderates these perceptions in GenAI contexts, when alternative interaction paradigms are available. This leads to our question: *"How does English language proficiency influence developers' preferences between traditional and structured intermediate prompting approaches, and what factors drive these preferences?"*

To address these questions, we examine "Metadata Prompting" as an intermediate representation between natural language and programming instructions, which provides a structured framework to express program requirements, thereby minimizing English grammatical structures, as visualized in Figure 1. This approach conceptually aligns with research on controlled intermediate languages that serve as a bridge between human expression and formal systems (Kesseli et al. 2025, Lehmann et al. 2023). We posit that this JSON-like formatting structure helps developers articulate their requirements in a format they are already proficient with, while minimizing reliance on English grammatical structures. By shifting the interaction paradigm from one specific spoken language to a programmatic format as an intermediate language, developers are allowed to express their needs in alignment with their technical expertise rather than their linguistic capabilities.

To investigate these questions, we partner with a technology firm in India that was recruiting full-stack engineers in the last quarter of 2023. The experimental design leveraged the firm's recruitment process to create a natural testing environment with 965 candidates. After having assessed their coding and English proficiency using standardized assessments, we implemented a 2×2 crossover design (e.g., Jones and Kenward 2003) where participants completed two distinct coding tasks

Traditional Prompting

"Create a Node.js function that generates notifications based on inventory changes in a retail system. The function should analyze differences between current and previous day's inventory levels. It will receive the inventory data for both days and notification templates as parameters. Products are classified into three priority levels: high-priority (essential items), medium-priority (regular stock), and low-priority (seasonal items). Different notifications should be sent based on threshold changes: If high-priority items fall below 20% stock, send an 'urgent-restock' alert. If medium-priority items fall below 30% stock, send a 'standard-reorder' notification. When any product drops more than 40% in a single day, send a 'rapid-depletion' warning. Finally, any low-priority item below 15% stock should trigger a 'review-necessity' message to evaluate if reordering is needed."

Metadata Prompting

```
{
"task": "Create Node.js function for inventory notifications",
"reason": "monitor stock changes between days",
"inputs": ["currentStock", "previousStock", "alertType",
"itemPriority"],
"notification_rules": [
{
"itemPriority": "high-priority",
"condition": "currentStock < 20%",
"alertType": "urgent-restock"
},
{
"itemPriority": "medium-priority",
"condition": "currentStock < 30%",
"alertType": "standard-reorder"
},
{
"itemPriority": "any",
"condition": "(previousStock - currentStock) > 40%",
"alertType": "rapid-depletion"
},
{
"itemPriority": "low-priority",
"condition": "currentStock < 15%",
"alertType": "review-necessity"
}
]
}
```

Figure 1 Comparison of Traditional and Metadata Prompting

with different combinations of task type (Task A and Task B), prompting approach (Traditional Prompting and Metadata Prompting), and the task order. Performance was measured across three key dimensions: task completion time, code accuracy, and task abandonment rates. This design allowed us to isolate the effects of prompting techniques while controlling for task-specific and order effects (See Section 3.1 for details of the design).

Our findings reveal three key insights about the relationship between English proficiency and the use of GenAI tools. First, we document clear evidence of a language-based digital divide: compared to developers with high English language proficiency, those with low proficiency show significantly lower performance across all metrics—with higher task abandonment, higher time taken, and lower accuracy—when using traditional prompting approaches. Second, we find that metadata prompting substantially mitigates these disparities. When using metadata prompting, the

performance gap between developers with high and low English proficiency narrows considerably across all metrics. In particular, the effect of English proficiency on both task completion time and accuracy becomes statistically insignificant. In other words, while lower English proficiency significantly predicts lower performance under traditional prompting, this relationship effectively disappears when metadata prompting is employed. Third, we find that *all* developers, regardless of their English proficiency, express a strong preference for metadata prompting over traditional approaches, with the preference being particularly pronounced among those with lower English proficiency. These results suggest that simple structural changes to prompting techniques in a format that programmers are comfortable with can significantly enhance the accessibility and effectiveness of GenAI tools across diverse linguistic backgrounds. The results are robust to different model specifications and accuracy measures, and after controlling for learning effects and the selection bias on individuals who abandon the task.

Our paper makes several important contributions to the literature on digital inequality, global work, and technology adoption. First, while prior work has documented how language barriers affect collaboration in global teams (Hinds et al. 2014) and digital tool adoption (Neeley et al. 2009), we present the first empirical evidence demonstrating how language proficiency contributes to the digital divide in the effective use of GenAI for coding. Second, we extend research on reducing barriers in global work (Fayard and Metiu 2014) by demonstrating how structured intermediate representations—conceptually similar to controlled natural languages (Lehmann et al. 2023)—can mitigate language-based performance disparities. Relatedly, we contribute to the literature on GenAI adoption in organizations (e.g., Raisch and Krakowski 2021) by validating a novel prompting technique that enhances accessibility across diverse linguistic backgrounds.

Our findings also have important practical implications for software development firms adopting GenAI tools. The global IT outsourcing market, valued at $651.54 billion in 2025 and expected to reach $850.73 billion by 2030, with India as one of the most dominant contributors.[2] Software

[2] IT Outsourcing (ITO) Market Size, Share, Trends Analysis & Statistics (2025 - 2030) - Mordor Intelligence, Accessed 9th March 2025

development firms, particularly those with global operations with linguistically diverse teams can use metadata prompting to avoid the communication anxieties and collaboration disruptions documented in prior research (Neeley et al. 2009). By integrating metadata prompting approach into their workflows, companies can empower developers who are not proficient in English to effectively utilize GenAI tools, improving overall efficiency and team collaboration. To maximize these benefits, firms should consider actively educating their teams on structured prompting techniques and enable them to prompt effectively for AI-assisted coding.

2. Literature Review

Our paper draws on and contributes to three distinct literature streams: research on cognitive load and digital divides in technology usage, studies examining low-resource languages in NLP and AI interactions, and work related to structured approaches in human-AI interactions.

2.1. Cognitive Load and Digital Divide in Technology Usage

The relationship between language proficiency and technology adoption has been extensively studied in the technology adoption literature. When users interact with technology in a non-native language, they experience increased cognitive load (Sweller 1988)—a phenomenon well-documented in cognitive psychology within the multi-lingual setting (e.g., Linck et al. 2009, Roussel et al. 2017). This additional cognitive burden stems from users simultaneously processing both the technical requirements of the task and the linguistic elements of interaction (Robinson 2001). Studies examining cognitive load in bilingual contexts reveal that working memory capacity becomes particularly constrained when individuals must operate in their non-primary language (Lundqvist et al. 2018).

In global work contexts, non-native English speakers often face significant challenges in effectively participating in technology-mediated communication(Neeley 2013). This constraint manifests more prominently in tasks requiring complex information processing and precise communication—characteristics inherent in software development. The additional mental effort required for language processing can detract from the cognitive resources available for primary task completion (Paas et al. 2003), further affecting technology adoption patterns and potentially reinforcing existing biases in technology (Fu et al. 2020).

Beyond the general literature on digital divide, recent research has begun to document the emergence of an "AI divide" - a new form of inequality specifically related to access to and benefits from artificial intelligence technologies. McElheran et al. (2024) provide evidence of geographical and sectoral disparities in AI adoption across America, highlighting how AI technologies may exacerbate existing socioeconomic divides rather than bridge them. Whereas extensive research exists on how language proficiency affects technology adoption and usage, there is a gap in understanding how such language proficiency affects GenAI interaction. Understanding how language proficiency affects their adoption, effectiveness, and potential abandonment is an open empirical question that relates to emerging concerns about digital divides in AI adoption (Ma et al. 2024), where barriers emerge not from access limitations, but from the design of interfaces that privilege certain languages.

2.2. Low-Resource Languages in AI Interactions

The technological landscape of GenAI models presents significant challenges for users of low-resource languages (LRLs). LRLs are characterized by limited digital corpus (Conneau et al. 2020) and scarce annotated datasets for training NLP models (Zoph et al. 2016, Adams et al. 2017), disadvantaging speakers of these languages when interacting with AI systems (Shi et al. 2023). Hindi, a language spoken by over 600 million people, is a prominent example of an LRL.

The tokenization processes employed by most LLMs further exacerbate these challenges. Many LRLs, particularly those from South and Southeast Asia, have high information density per word compared to Latin and Cyrillic languages. This leads to what researchers term "token fertility" issues, where more tokens are needed to encode equivalent information compared to English (Ahuja et al. 2023, Ahia et al. 2023). While fine-tuning on domain-specific datasets has proven effective for improving LLM performance in software engineering contexts (Chronopoulou et al. 2022), this approach rarely resolves token fertility problems for LRLs. The data scarcity inherent to low-resource languages makes comprehensive fine-tuning infeasible, leaving the base model's suboptimal tokenization unchanged and perpetuating performance disparities.

For programming contexts specifically, the scarcity of technical documentation and technical terminology in LRLs creates additional barriers, resulting in LLMs struggling to map requirements specified in LRLs to accurate code implementations (Zheng et al. 2023). The lack of sufficient domain-specific training data to establish reliable correlations between LRL expressions and programming constructs creates a disconnect for developers from LRL backgrounds when attempting to leverage GenAI tools for programming tasks.

While research demonstrates clear performance disparities between HRLs and LRLs in general language tasks (Zhong et al. 2024), whether these disparities extend to specialized contexts like programming, and whether alternative interaction paradigms might mitigate them, remains unclear.

2.3. Structured Approaches in Human-AI Interaction

Research on human-AI interaction has demonstrated the effectiveness of structured approaches in reducing cognitive load and improving communication clarity Abbasi et al. (2024). Transitioning from a natural human-like communication to more structured strategy can significantly enhance information processing and reduce ambiguity, especially in scenarios needing precise communication (Karten et al. 2023, Zhang et al. 2023, Pangakis et al. 2023, Cheng et al. 2024, de Kok 2025).

Recent work on GenAI prompting techniques has revealed that different prompting strategies, especially related to structuring the prompts, can significantly impact model performance and user experience (Liu et al. 2023b, Qiu et al. 2023). For instance, Jiang et al. (2020) notes that paraphrasing the questions in a "right" way can improve the quality of responses, aligning with emerging research on how to systematically shape and guide AI behavioral patterns (Ni et al. 2025). Wei et al. (2022) demonstrated that chain-of-thought prompting, i.e., asking for a step-by-step reasoning for a response before the actual response, improves reasoning capabilities, while Kojima et al. (2022) showed that carefully structured prompts enable sophisticated reasoning even without task-specific training. Zhou et al. (2022) further illustrated that breaking down complex tasks into smaller, structured components through "least-to-most" prompting yields superior results.

A particularly promising direction involves controlled natural languages (CNLs) as intermediate representations between human expression and formal systems. Lehmann et al. (2023) demonstrated that language models can effectively function as CNL semantic parsers for knowledge graph question answering, creating a structured bridge between natural language queries and formal database operations. Similar intermediate representations using constrained solvers have proven effective in reducing ambiguity and improving the outcome from LLMs (Kesseli et al. 2025).

While structured approaches have proven effective in enhancing human interaction with GenAI (Wei et al. 2022, Kojima et al. 2022, Zhou et al. 2022), their potential to overcome spoken language barriers remains unexplored. This gap is particularly significant in software development contexts, where the core task of programming operates on a formal language distinct from spoken language.

2.4. Research Gaps and Questions

Despite the growing body of research across these domains, several critical gaps remain. First, while studies have documented how language barriers affect technology adoption and usage in global work contexts (Neeley 2013, Hinds et al. 2014), quantitative evidence on how language proficiency specifically impacts GenAI tool usage in programming contexts is lacking. Second, although research demonstrates the effectiveness of structured prompting approaches in improving model performance (Kesseli et al. 2025, Lehmann et al. 2023), whether such approaches can mitigate language barriers in technical contexts remains untested. Finally, factors influencing user preferences between traditional and structured interaction paradigms, particularly as moderated by language proficiency, have not been systematically examined.

Our investigation centers on understanding how language proficiency shapes developers' interaction with GenAI tools and whether structured prompting approaches can mitigate potential barriers. First, to understand the relationship between English proficiency and GenAI tool usage, we examine how varying levels of language proficiency affect developers' ability to effectively utilize these tools. Quantitatively establishing *whether* (and if yes, by *how much*) English proficiency affects GenAI tool usage is crucial for comprehending potential digital divides in AI-assisted software development. This leads us to our first research question: *"How does English language proficiency affect developers' ability to effectively utilize GenAI tools for software development tasks?"*

Next, we examine whether a structured approach, specifically a domain-specific intermediate representation, to prompting can enhance outcomes and potentially bridge language-related barriers. This leads us to our second research question *"To what extent can metadata prompting mitigate the impact of limited English proficiency on developers' ability to effectively utilize GenAI tools?"*

Finally, we explore whether developers' English language proficiency impacts their preference to use traditional linguistic prompting or the structured prompting strategies. Understanding these preferences is crucial because even effective solutions may see limited adoption if users find them uncomfortable or challenging to use. This leads to our question: *How does English language proficiency influence developers' preferences between traditional and metadata prompting approaches, and what factors drive these preferences?*

3. Empirical Context and Experiment

Our study is conducted in partnership with a medium-sized technology firm in India that specializes in full-stack software development solutions. The firm provides custom software development services to clients across multiple industries, including retail, real estate, hospital, education, and restaurants. Our study benefits from the firm's active recruitment of full-stack engineers, which presents an ideal opportunity to examine how developers with varying levels of English proficiency interact with AI tools in a real field setting.[3] This setting provides a natural environment for the technology partner to run randomized experiments, as candidates represent a diverse pool of developers with varying levels of English proficiency and technical expertise, typical of the global software development workforce.

The candidates in our study apply for the position of *full-stack engineer*, a role that requires proficiency in the MERN (MongoDB, Express.js, React.js, Node.js) technology stack. The role specifically demands expertise in developing and deploying scalable web applications, with responsibilities spanning both front-end user interfaces and back-end server logic. Our sample comprises 965 candidates who participated in the recruitment process. The average candidate in our sample

[3] The firm identifies proficiency in using AI-assisted development tools as an important skill for their prospective hires, given their increasing integration of these tools into their development workflow.

has 0-10 years of programming experience, with 22.5% holding master's degrees in computer science or related fields. This distribution aligns with the broader demographics of India's technology workforce, where developers often learn English as their second or third language while pursuing technical education.[4] As a part of the application process, the candidates also provided information such as their experience in using AI tools for programming, their gender, and other relevant background information. Table 1 provides the details about the background of the candidates.

3.1. Experimental Design

The experiment comprises three distinct stages designed to comprehensively assess the relationship between language proficiency and AI tool usage effectiveness. Figure 2 summarizes the flow of the experiment across three stages.

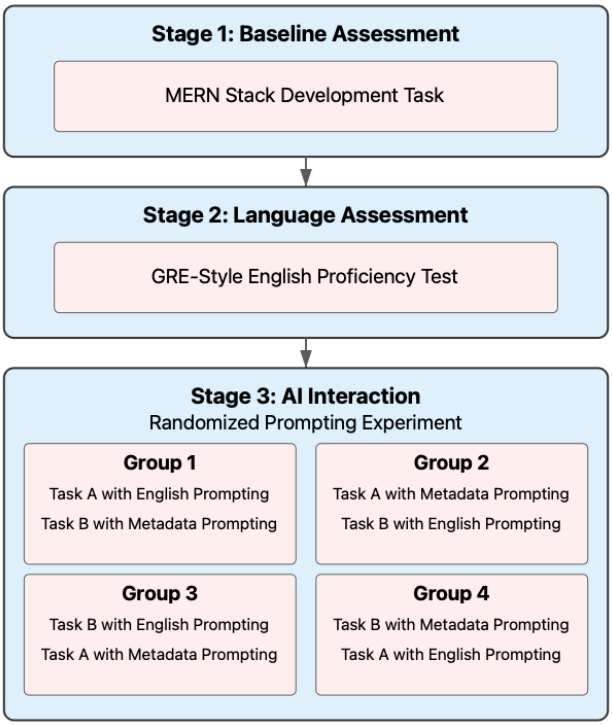


Figure 2 Experimental Design Overview

Stage 1: This stage evaluates candidates' baseline coding proficiency through a standard software development task. Candidates created a MERN (MongoDB, Express.js, React.js, Node.js)

[4] In India, who speaks in English, and where? - Mint, Accessed 22 December 2024.

Table 1 Variable Description and Summary Statistics

Table 1(a): Variable Description

Variable Name	Description
Individual-level Variables	
$EnglishProficiency_i$	GRE-based English proficiency score for candidate i, scaled from 0 to 1
$CodingAbility_i$	Baseline coding proficiency score for candidate i on a 100-point scale
$AcademicPerformance_i$	Undergraduate academic performance of candidate i on a 10-point scale
$Experience_i$	Years of programming experience for candidate i
$AIExperience_i$	Weekly AI tool usage in hours for candidate i
$Gender_i$	Binary indicator for candidate i's gender (1 = Male, 0 = Female)
$Masters_i$	Binary indicator for whether candidate i holds a master's degree
$Abandonment_i$	Binary indicator for whether candidate i abandoned Test 1
Task-level Variables	
$CompletionTime_{it}$	Time taken in minutes by candidate i to complete task t
$Accuracy_{it}$	Average percentage of unit tests passed for code generated from candidate i's prompt in task t, calculated over two runs, to enhance reliability
$MetaData_{it}$	Binary indicator for whether candidate i used metadata prompting in task t
$Test_t$	Binary indicator for test number (1 = Test 2, 0 = Test 1)
$Task_t$	Binary indicator for task type (1 = Task B, 0 = Task A)

Table 1(b): Summary Statistics

Variable	Obs	Mean	Std. Dev.	Min	Max
Individual-level Variables					
$EnglishProficiency_i$	965	0.560	0.110	0.20	0.93
$CodingAbility_i$	965	85.041	5.218	75	100
$AcademicPerformance_i$	965	7.298	0.601	6.30	9.50
$Experience_i$	965	0.762	1.669	0	10
$AIExperience_i$	965	1.180	1.011	0	4.00
$Gender_i$	965	0.634	0.482	0	1
$Masters_i$	965	0.225	0.418	0	1
$Abandonment_i$	965	0.284	0.451	0	1
Task-level Variables					
$CompletionTime_{it}$	1347	44.476	14.675	20	85
$Accuracy_{it}$	1347	73.085	13.649	30	100
$MetaData_{it}$	1347	0.509	0.500	0	1
$Test_t$	1347	0.487	0.500	0	1
$Task_t$	1347	0.501	0.500	0	1

Note: The difference in observation counts between individual-level (965) and task-level variables (1347) reflects our experimental design. Each candidate is expected to complete two tasks (one with each prompting technique). Only 691 completed Test 1, and 656 of these completed Test 2, resulting in 1347 (691+656) task-level observations.

application without AI assistance, allowing us to establish their fundamental software development skills and capabilities. This is a 90 minutes long assessment test that is taken by every candidate.

Stage 2: This stage focuses on assessing candidates' English language proficiency using a standardized GRE-style assessment. The 60-minute test evaluates three critical dimensions of language ability: Reading Comprehension, Text Completion, and Sentence Equivalence. This comprehensive evaluation helps quantify the candidates' ability to understand and articulate complex technical requirements in English, a crucial skill for effective AI tool interaction.

Stage 3: In this stage, we examine how candidates interact with AI tools under different prompting conditions. Each participant completes two coding tasks, each of 90 minutes long—Task A and Task B—using different prompting techniques. Figure 3 visualizes the test interface for this stage. The task requirements are communicated through videos in *Hindi*, candidates' native language.[5] This design choice ensures that understanding the requirements is not a confounding factor in our assessment. For each task, candidates were asked to translate their understanding of the requirements into English prompts for the AI tool, either using traditional natural language prompting or "metadata" prompting approach. We implement a 2×2 crossover design (e.g., Jones and Kenward 2003) where candidates who used metadata prompting for Task A employed traditional prompting for Task B and vice versa, as well as switch the sequence.[6]

3.2. Variables

Table 1 provides the details about the key dependent variables, which measure different aspects of task performance. $CompletionTime_{it}$ represents the time in minutes taken by participant i to complete the task t. $Accuracy_{it}$ captures the average percentage of unit tests passed by the code generated from participant i's prompts in task t, based on two separate runs.[7] $Abandonment_i$ is a binary indicator that takes the value of 1 if participant i abandons the task, and 0 otherwise.

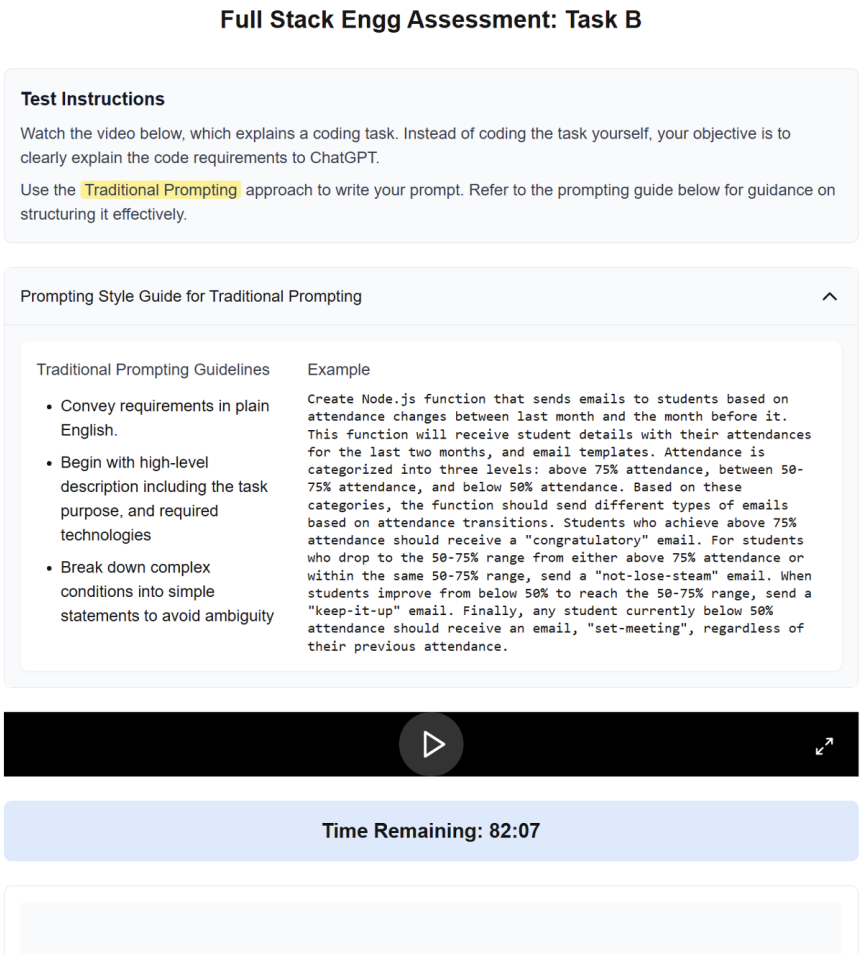
For independent variables, we focus on English proficiency and prompting technique. $EnglishProficiency_i$ represents participant i's score on the GRE-style English assessment, scaled from 0 to 1 to enable direct coefficient comparison. $MetaData_{it}$ is a binary variable indicating whether participant i used metadata prompting (1) or traditional prompting (0) for task t.

We also include several control variables to account for individual characteristics and prior experience. $CodingAbility_i$ captures the baseline programming proficiency of participant i on a 100-point scale, measured through the initial MERN stack development task. $AcademicPerformance_i$

[5] All candidates who applied to this position and are a part of this experiment are Hindi natives.

[6] The 4 combinations are: (1) traditional prompting for Task A + metadata prompting for Task B; (2) traditional prompting for Task B + metadata prompting for Task A; (3) metadata prompting for Task A + traditional prompting for Task B; (4) metadata prompting for Task B + traditional prompting for Task A.

[7] Each prompt by the candidate was used twice to generate code, and the final accuracy score is the average of the two percentages of unit tests passed, enhancing the measure's reliability by accounting for variability in the AI model's output. The results remain qualitatively similar if we use either of the individual scores instead of their average, as detailed in the robustness check section.



Note: The requirements are communicated through a video in Hindi, the native language of candidates.

Figure 3 Test Interface Screenshot

represents undergraduate academic performance on a 10-point scale. $Experience_i$ indicates years of programming experience, while $AIExperience_i$ measures weekly AI tool usage in hours. $Gender_i$ and $Masters_i$ are binary indicators for participant gender (1 = Male) and educational qualification (1 = Master's degree) respectively. Our experimental design also includes task-specific controls. $Test_t$ is a binary variable taking the value 0 for Test 1 and 1 for Test 2. $Task_t$ similarly indicates whether the participant is working on Task A (0) or Task B (1). These variables help control for potential order effects and task-specific variations in our analysis.

3.3. Empirical Framework

Our analysis employs three distinct empirical models to examine how *metadata* prompting moderates the impact of language deficiency on developers' interaction with AI tools.

Table 2 Randomization Check

	Group 1 (C1)		Group 2(T1)		Group 3(C2)		Group 4(T2)		p-value
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	$(C1 = T1 = C2 = T2)$
Observations	241		242		241		241		
English Proficiency	0.56	0.11	0.56	0.11	0.55	0.10	0.56	0.12	0.99
Experience	0.73	1.59	0.70	1.66	0.77	1.55	0.85	1.86	0.75
CodingAbility	85.15	5.23	84.94	5.24	85.04	5.26	85.04	5.16	0.98
Undergrad Grades	7.30	0.60	7.29	0.60	7.30	0.60	7.30	0.60	1.00
AI Experience	1.13	1.01	1.22	1.01	1.21	1.05	1.15	0.97	0.68
Gender	0.61	0.49	0.62	0.49	0.65	0.48	0.65	0.48	0.74
Masters	0.20	0.40	0.26	0.44	0.24	0.43	0.20	0.40	0.24

Note: Group 1 (C1) → English Prompting for Task A, Metadata Prompting for Task B in that order
Group 2(T1) → Metadata Prompting for Task A, English Prompting for Task B in that order
Group 3(C2) → English Prompting for Task B, Metadata Prompting for Task A in that order
Group 4(T2) → Metadata Prompting for Task B, English Prompting for Task A in that order

3.3.1. Identification Strategy: Our identification strategy relies on the random assignment of participants across treatment conditions in a 2×2 experimental design, defined by the combination of two tasks (Task A and Task B) and two prompting techniques (regular and metadata) in a sequential order. Randomization ensures that, on average, both observable and unobservable characteristics are balanced across the four resulting groups.

To validate this assumption, we conduct balance checks on key participant characteristics using analysis of variance (ANOVA). Table 2 reports the means, standard deviations, and p-values testing equality across groups. The results—English Proficiency ($p = 0.99$), Experience ($p = 0.75$), Coding Ability ($p = 0.98$), Undergraduate Grades ($p = 1.00$), AI Experience ($p = 0.68$), Gender ($p = 0.74$), and Masters ($p = 0.24$)—indicate no statistically significant differences, supporting the validity of our randomization. This strengthens the credibility of our causal claims in the subsequent analyses.

3.3.2. Task Abandonment: To analyze how English proficiency and metadata prompting influence task abandonment probability, we specify the following logistic regression model:

$$\ln\left(\frac{P(Abandonment_i = 1)}{1 - P(Abandonment_i = 1)}\right) = \beta_0 + \beta_1 EnglishProficiency_i + \beta_2 MetaData_i + \beta_3 EnglishProficiency_i \times MetaData_i + \gamma X_i + \epsilon_i \tag{1}$$

where $Abandonment_i$ is a binary indicator for whether participant i abandoned the first task. The left-hand side represents the log-odds of task abandonment, while the coefficients capture how English proficiency and metadata prompting affect these odds. Of particular interest is β_3,

which measures how metadata prompting moderates the relationship between English proficiency and task abandonment probability. The vector X_i includes control variables such as coding ability, undergraduate grades, AI experience, gender, education, and prior experience.

3.3.3. Completion Time: The second analysis investigates whether metadata prompting moderates the impact of language deficiency on task completion time. We specify a linear model:

$$\begin{aligned} CompletionTime_{it} = \alpha_0 + \alpha_1 EnglishProficiency_i + \alpha_2 MetaData_{it} + \\ \alpha_3 EnglishProficiency_i \times MetaData_{it} + \gamma X_i + \epsilon_{it} \end{aligned} \tag{2}$$

where $CompletionTime_{it}$ represents the time (in minutes) taken by participant i to complete the task t. The interaction term allows us to assess whether metadata prompting differentially affects completion time based on English proficiency levels.

3.3.4. Task Accuracy: Finally, to analyze the factors affecting task accuracy, we specify:

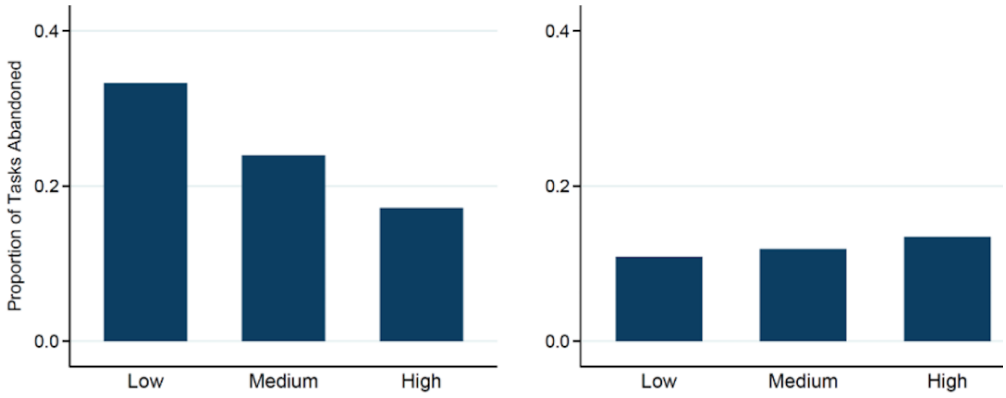
$$\begin{aligned} Accuracy_{it} = \delta_0 + \delta_1 EnglishProficiency_i + \delta_2 MetaData_{it} + \\ \delta_3 EnglishProficiency_i \times MetaData_{it} + \gamma X_i + \epsilon_{it} \end{aligned} \tag{3}$$

where $Accuracy_{it}$ represents the percentage of unit tests passed for participant i's generated code for task t. This specification enables us to examine how metadata prompting influences the relationship between English proficiency and code accuracy.

4. Results

4.1. Task Abandonment

We begin by examining how English proficiency and metadata prompting affect participants' likelihood of abandoning coding tasks. Figure 4 provides a model-free evidence graphically by plotting the probabilities of task abandonment across English proficiency levels with and without metadata prompting. Sub-figure (a) shows that the abandonment rate is higher for participants in the lowest tercile of language proficiency under standard prompting technique. Sub-figure (b) shows that the differences vanish under metadata prompting technique.



(a) English Prompting (b) Metadata Prompting

Note: The graph categorizes language proficiency into three terciles: Low, Medium, and High.

Figure 4 Model Free Evidence - Prompting Strategies, Language Barriers, and Abandonment

Table 3 presents estimates from the logit model specified in equation 1. Column (1) introduces English proficiency as the key explanatory variable and shows that English proficiency has a significant negative relationship with task abandonment ($\beta_1 = -1.760$, $p < 0.01$). Column (2) incorporates metadata prompting, reveals that metadata prompting significantly reduces task abandonment ($\beta_2 = -1.032$, $p < 0.01$). The persistent significance of English proficiency ($\beta_1 = -1.859$, $p < 0.01$) in this specification indicates that language barriers remain influential even after accounting for prompting technique. The interaction specification in Column (3) demonstrates how metadata prompting moderates the relationship between English proficiency and task abandonment. Under regular prompting, English proficiency significantly reduces task abandonment ($\beta_1 = -4.241$, $p < 0.01$). However, with metadata prompting, this negative effect diminishes and becomes statistically insignificant ($\beta_1 + \beta_3 = 1.004$, $p = 0.31$), indicating that metadata prompting effectively mitigates the influence of language barriers on task completion.

4.2. Task Completion Time

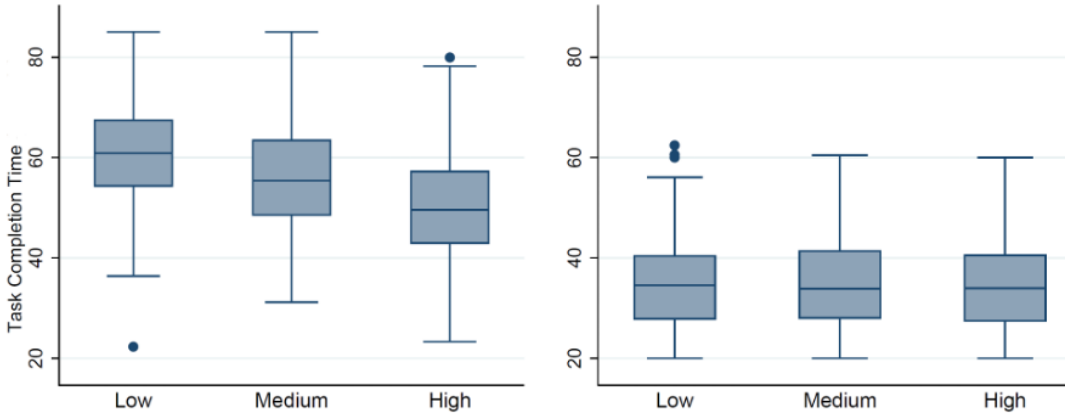
We next examine how spoken language proficiency influence the time taken to complete coding tasks and whether metadata prompting moderates this relationship. Figure 5 provides a model-free illustration of how English proficiency levels affect the task completion time with standard prompting and metadata prompting techniques. Under the standard prompting technique, participants with low language proficiency take approximately 10 additional minutes to complete the task

Table 3 Logistic Regression Results - Task Abandonment

	Base	Main Effects	Interaction
English Proficiency	-1.760***	-1.859***	-4.241***
	(0.644)	(0.667)	(0.960)
Metadata Prompting		-1.032***	-3.945***
		(0.153)	(0.791)
English Proficiency × Metadata Prompting			5.245***
			(1.386)
Observations	965	965	965
χ^2	16.53	55.37	74.35

Notes: Huber-White heteroskedasticity-consistent standard errors in parentheses
* p < 0.10, ** p < 0.05, *** p < 0.01
A baseline model with only control variables yields qualitatively similar results
Control variables such as coding ability, undergraduate grades, AI experience, gender, education and prior experience used but not displayed

compared to participants with high language proficiency. Under metadata prompting technique, this difference disappears, with everyone taking approximately 35 minutes to complete the task.



(a) English Prompting (b) Metadata Prompting

Note: The graph categorizes language proficiency into three terciles: Low, Medium, and High.

Figure 5 Model Free Evidence - Prompting Strategies, Language Barriers, and Task Completion Time

Table 4 presents estimates from the linear regression specified in equation 2. Column (1) shows that English proficiency significantly affects completion time ($\alpha_1 = -23.671$, $p < 0.01$). Column (2) demonstrates that metadata prompting significantly reduces task completion time ($\alpha_2 = -20.480$, $p < 0.01$). The interaction specification in Column (3) reveals how metadata prompting moderates the relationship between English proficiency and completion time. Under regular prompting, English proficiency significantly decreases the completion time ($\alpha_1 = -49.223$, $p < 0.01$). However,

with metadata prompting, this negative effect becomes statistically insignificant ($\alpha_1 + \alpha_3 = -2.064$, $p = 0.49$), indicating that metadata prompting effectively neutralizes the impact of English proficiency on completion time.

Table 4 Linear Regression - Task Completion Time

	Base	Main Effects	Interaction
English Proficiency	-23.671***	-24.940***	-49.223***
	(3.480)	(2.527)	(3.413)
Metadata Prompting		-20.480***	-47.285***
		(0.546)	(2.643)
English Proficiency × Metadata Prompting			47.159***
			(4.544)
All Controls	Yes	Yes	Yes
Observations	1,347	1,347	1,347
Adjusted R-squared	0.058	0.540	0.573

Notes: Huber-White heteroskedasticity-consistent standard errors in parentheses
* p < 0.10, ** p < 0.05, *** p < 0.01
A baseline model with only control variables yields qualitatively similar results
Control variables such as coding ability, undergraduate grades, AI experience, gender, education and prior experience used but not displayed

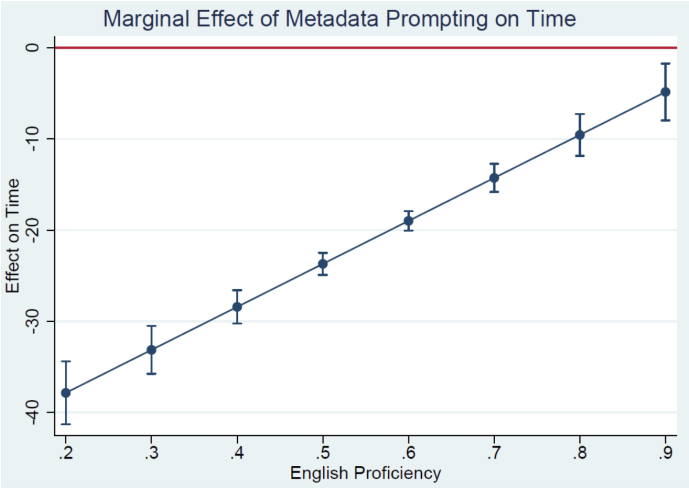
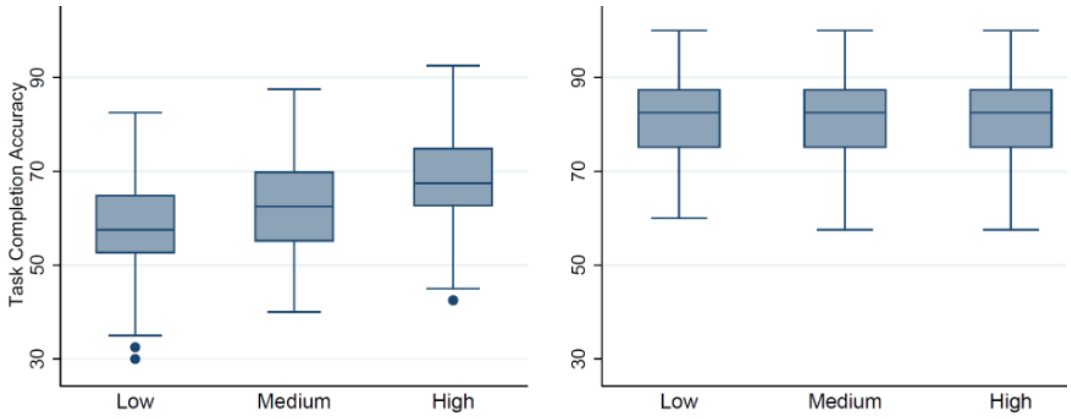


Figure 6 Marginal Effect: Impact of Metadata Prompting on Completion Time with varying English Proficiency

To further examine how the effect of metadata prompting varies across the English proficiency spectrum, Figure 6 plots the marginal effect of metadata prompting. The consistently negative marginal effects, with confidence intervals not crossing zero, indicate that metadata prompting significantly reduces completion time across all proficiency levels. The upward slope reveals that the completion time reduction is more pronounced for developers with lower English proficiency, providing evidence for metadata prompting's role in mitigating the language barrier.

4.3. Task Accuracy

Finally, we examine how English proficiency and metadata prompting influence the accuracy of the code generated through the prompts created by the participants. Figure 7 provides a visual representation of the model-free relationship between language proficiency and code accuracy under the two prompting techniques. Under the standard prompting technique, participants in the lowest tercile of language proficiency achieved under 60% accuracy whereas those in the top tercile achieved accuracy close to 70%. Under metadata prompting, the accuracy level of all terciles increases to more than 80% and the gap between the lowest and highest terciles disappears.



(a) English Prompting (b) Metadata Prompting

Note: The graph categorizes language proficiency into three terciles: Low, Medium, and High.

Figure 7 Model Free Evidence - Prompting Strategies, Language Barriers, and Task Completion Accuracy

Table 5 presents estimates from the linear regression specified in equation 3. Column (1) shows that English proficiency significantly affects code accuracy ($\delta_1 = 21.190,\ p < 0.01$). Column (2) demonstrates that metadata prompting substantially improves code accuracy ($\delta_2 = 18.633,\ p < 0.01$). The interaction specification in Column (3) reveals how metadata prompting moderates the relationship between English proficiency and code accuracy. Under regular prompting, English proficiency significantly increases accuracy ($\delta_1 = 41.224, p < 0.01$). However, with metadata prompting, this positive effect becomes statistically insignificant ($\delta_1 + \delta_3 = 4.56,\ p = 0.14$), indicating that metadata prompting effectively neutralizes the impact of English proficiency on code accuracy.

Table 5 Linear Regression - Task Accuracy

	Base	Main Effects	Interaction
English Proficiency	21.190***	22.345***	41.224***
	(3.264)	(2.421)	(3.319)
Metadata Prompting		18.633***	39.473***
		(0.522)	(2.621)
English Proficiency × Metadata Prompting			-36.664***
			(4.512)
All Controls	Yes	Yes	Yes
Observations	1,347	1,347	1,347
Adjusted R-squared	0.055	0.517	0.539

Notes: Huber-White heteroskedasticity-consistent standard errors in parentheses
* $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$
A baseline model with only control variables yields qualitatively similar results
Control variables such as coding ability, undergraduate grades, AI experience, gender, education and prior experience used but not displayed.

To further investigate the relationship between metadata prompting and English proficiency, Figure 8 depicts the marginal effect of metadata prompting on the full spectrum of English proficiency scores. The plot reveals that metadata prompting leads to statistically significant improvements in accuracy at all proficiency levels, as evidenced by the consistently positive marginal effects and confidence intervals that do not intersect with zero. However, the downward trend of the marginal effects underscores that the magnitude of improvement is inversely related to English proficiency. In other words, the benefits of metadata prompting in terms of enhanced accuracy are most substantial for developers with lower English proficiency, further reinforcing the notion that this approach is particularly effective in mitigating the challenges posed by language barriers in AI-assisted development contexts.

4.4. Robustness Checks

4.4.1. Learning Effects: Although our 2×2 crossover experimental design helps address potential learning effects by randomizing the order of prompting techniques, we conduct additional analysis to explicitly test whether our results might be driven by participants becoming more familiar with the tasks over time. We augment our main specifications with a three-way interaction between English proficiency, metadata prompting, and a Test 2 indicator, as follows:

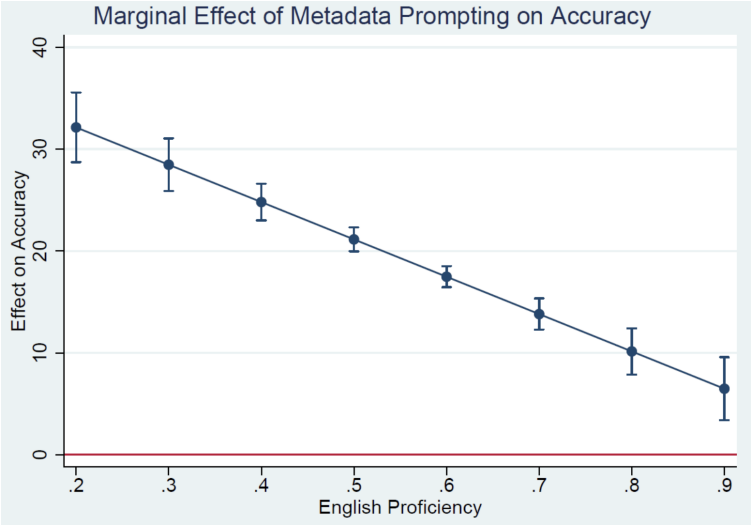


Figure 8 Marginal Effect: Impact of Metadata Prompting on Task Accuracy with varying English Proficiency

$$Y_{it} = \gamma_0 + \gamma_1 EnglishProficiency_i + \gamma_2 MetaData_{it} + \gamma_3 Test2_{it} + \gamma_4 (EnglishProficiency_i \times Test2_{it}) + \gamma_5 (EnglishProficiency_i \times MetaData_{it}) + \gamma_6 (MetaData_{it} \times Test2_{it}) + \gamma_7 (EnglishProficiency_i \times MetaData_{it} \times Test2_{it}) + \gamma X_i + \epsilon_{it} \quad (4)$$

Here, Y_{it} are the two dependent variables, task completion time and task completion accuracy. Table A1 presents these results. We see that the Test 2 indicator and its interactions with English proficiency and metadata prompting (i.e., γ_3, γ_6 and γ_7) are statistically insignificant across both outcome measures: completion time and accuracy. Meanwhile, our main effects remain qualitatively similar to our baseline specifications - the coefficient for English Proficiency, Metadata Prompting and their interaction continues to suggest that metadata prompting effectively mitigates language barriers. This robustness check effectively addresses the concerns that participants simply becoming more familiar with the tasks over time, may be driving the results.

4.4.2. Alternative Accuracy Measures: Our main analysis uses a composite accuracy measure that averages the percentage of unit tests passed across two independent code generations from each prompt. This approach helps account for potential variability in the AI model's output. Here, we examine whether our results are sensitive to this measurement choice by considering alternative specifications of accuracy.

First, we replicate our analysis using only the first code generation attempt for each prompt. The results, presented in Column (2) of Table A2, remain consistent with our main findings from Table 5, which we present as Column (1) here. The interaction between English proficiency and metadata prompting maintains its significance and direction ($\delta_3 = -36.553$, $p < 0.01$), with similar standard errors to our main specification (4.341 vs. 4.512). Next, we examine performance using only the second attempt (Column 3), which yields nearly identical results ($\delta_3 = -36.775$, $p < 0.01$).

Finally, we consider using the maximum accuracy score between the two code generations, reflecting the best performance achieved from each prompt. Column (3) shows that this specification also yields qualitatively similar results. The moderating effect of metadata prompting remains strong ($\delta_3 = -36.170$, $p < 0.01$), and the pattern of coefficients closely mirrors our main findings.

These consistent results across different accuracy specifications suggest that the effectiveness of metadata prompting in mitigating language barriers are not artifacts of how we measure task accuracy.

4.4.3. Addressing Sample Selection: We examine whether our results are robust to excluding Test 1 observations from participants who unexplainably abandoned Test 2. This approach removes any potential bias from including initial performance data from participants who may have had a systematically different propensity to complete the full experiment. After excluding these 35 observations, reducing our sample from 1,347 to 1,312 observations, we find that our main results remain qualitatively unchanged. As shown in Table A3, the interaction between English proficiency and metadata prompting maintains its significance and magnitude for both accuracy ($\delta_3 = -36.216$, $p < 0.01$) and completion time ($\alpha_3 = 48.023$, $p < 0.01$). The stability of these coefficients, compared to our main specification ($\alpha_3 = 47.159$ and $\delta_3 = -36.664$ respectively), suggests that selective attrition between tests does not substantially influence our findings about the effectiveness of metadata prompting in mitigating language barriers.

4.4.4. Alternative Model Specifications: For task abandonment, whereas our main analysis employs a logistic regression, we consider two alternatives. First, we estimate a linear model

that offers directly interpretable coefficients. As shown in Column (1) of Table A4, the interaction between English proficiency and metadata prompting remains significant ($\beta_3 = 1.085$, $p < 0.01$), indicating that metadata prompting reduces the abandonment gap associated with lower English proficiency. A probit specification, presented in Column (2) of Table A4, yields qualitatively similar results ($\beta_3 = 3.190$, $p < 0.01$), suggesting our findings are not sensitive to assumptions about the models or the error distribution.

For completion time, we consider its potential interpretation as a count variable (measured in minutes) and estimate both Poisson and negative binomial regression models. These specifications account for the non-negative, discrete nature of time measurements and allow for potential non-linear effects. Under both specifications, our main conclusions about the effectiveness of metadata prompting in reducing language-based performance gaps remain unchanged. The interaction term remains significant in both the Poisson ($\alpha_3 = 0.835$, $p < 0.01$) and negative binomial models ($\alpha_3 = 0.848$, $p < 0.01$), as seen in Columns (3) and (4) of Table A4, respectively. All results confirm that metadata prompting significantly reduces completion time gaps between participants with different levels of English proficiency and is not driven by our model specification.

5. Analysis of User Preferences for Prompting Techniques

Following the completion of Stage 3, we conducted a survey to understand participants' preferences between regular and metadata prompting techniques. The goal is to understand how English proficiency relates to participants' preferences between prompting techniques. The survey evaluated three distinct dimensions of user experience, each measured on a 5-point scale (1 = strongly prefer regular prompting, 3 = neutral, 5 = strongly prefer metadata prompting). The first dimension, *preferred future use* captured their preference for a prompting strategy. The second dimension measured participants' self-reported *confidence in prompting* using each technique. For the third dimension, participants evaluated the *expected effectiveness* in communicating requirements.

We begin our analysis by dividing participants into three groups based on their English proficiency scores: low (bottom tercile, $n = 255$), medium (middle tercile, $n = 227$), and high (top

tercile, $n = 174$). Results revealed a consistent pattern: participants in the low proficiency group demonstrated the strongest preference for metadata prompting across all dimensions (*preferred future use*: $M = 4.28$, $SD = 1.03$; *confidence in prompting*: $M = 4.32$, $SD = 0.92$; *expected effectiveness*: $M = 4.26$, $SD = 1.04$), followed by the medium proficiency group (*preferred future use*: $M = 3.92$, $SD = 1.08$; *confidence in prompting*: $M = 4.00$, $SD = 1.03$; *expected effectiveness*: $M = 3.97$, $SD = 1.11$;), while the high proficiency group exhibited the weakest preference for metadata prompting (*preferred future use*: $M = 3.62$, $SD = 1.21$; *confidence in prompting*: $M = 3.58$, $SD = 1.23$; *expected effectiveness*: $M = 3.67$, $SD = 1.30$). Notably, even participants with high English proficiency demonstrated a preference for metadata prompting across all dimensions.

Next, we perform the one-way ANOVA to understand the differences. The results are presented in Table 6. The results revealed significant differences across proficiency groups for all dimensions (*preferred future use*: $F(2, 653) = 19.45$, $p < .001$; *confidence in prompting*: $F(2, 653) = 26.35$, $p < .001$; *expected effectiveness*: $F(2, 653) = 14.31$, $p < .001$; Post-hoc Tukey tests, presented in Table A5 confirmed systematic differences between all proficiency groups. Low-proficiency participants consistently showed stronger preferences for metadata prompting compared to both medium and high-proficiency participants across all dimensions (all $p < .01$).

Table 6 Analysis of Variance Results for Preference Measures

Source	SS	df	MS	F	p-value
Panel A: Preferred Future Use					
Between groups	46.697	2	23.348	19.45	0.000
Within groups	784.046	653	1.201		
Total	830.742	655	1.268		
Bartlett's test for equal variances:	$\chi^2(2) = 6.108$, p = 0.047				
Panel B: Confidence in Prompting					
Between groups	57.721	2	28.860	26.35	0.000
Within groups	715.156	653	1.095		
Total	772.877	655	1.180		
Bartlett's test for equal variances:	$\chi^2(2) = 17.949$, p = 0.000				
Panel C: Expected Effectiveness					
Between groups	37.082	2	18.541	14.31	0.000
Within groups	845.904	653	1.295		
Total	882.986	655	1.348		
Bartlett's test for equal variances:	$\chi^2(2) = 10.824$, p = 0.004				

Notes: SS = Sum of Squares; df = degrees of freedom; MS = Mean Square
All measures use 5-point scale (1 = prefer regular prompting, 5 = prefer metadata prompting)

To more rigorously examine the relationship between English proficiency and prompting preferences, we estimate a series of OLS regressions that control for participant characteristics. Specifically, for each preference dimension d, we estimate:

$$Preference_i^d = \theta_0 + \theta_1 EnglishProficiency_i + \gamma X_i + \epsilon_i \tag{5}$$

where $Preference_i^d$ represents the participant's average preference score across the survey dimensions and $EnglishProficiency_i$ carries a value of 1 for participants with above median language proficiency. The results, presented in Table 7, show a strong relationship between English deficiency and preference for metadata prompting across all dimensions. These effects persist after controlling for participants' coding ability, academic performance, AI experience, etc.

Table 7 Future Use Preferences Analysis

	Overall Preference	Preferred Future Use	Confidence in Prompting	Expected Effectiveness
English Proficiency	-2.691***	-2.782***	-2.887***	-2.404***
	(0.340)	(0.356)	(0.343)	(0.376)
Observations	656	656	656	656
Adjusted R-squared	0.083	0.082	0.092	0.057

Notes: Huber-White heteroskedasticity-consistent errors in parentheses
* p < 0.10, ** p < 0.05, *** p < 0.01
All preference measures use 5-point scale (1 = prefer regular prompting, 5 = prefer metadata prompting)

6. Discussion and Conclusion

The rise of programming specific Generative AI (GenAI) tools represents a "Solow moment" for artificial intelligence tools, with studies documenting significant productivity gains through AI-assisted programming (Yeverechyahu et al. 2024, Peng et al. 2023) and demonstrating alignment with human preferences in various contexts (Qiu et al. 2023). This technological advancement is also reshaping programming labor markets globally (Liu et al. 2023a) and is signaling a broader transformation in knowledge work, where researchers are developing rigorous frameworks to systematically leverage GenAI across various analytical tasks (Cheng et al. 2024, de Kok 2025, Pangakis et al. 2023). Just as the emergence of user-friendly operating systems transformed computers into general-purpose productivity devices and eventually addressed the Solow's productivity paradox,

GenAI appears to be fulfilling the long-anticipated potential of AI to revolutionize knowledge work, particularly in domains like software development. However, this technological advancement emerges against a backdrop of increasing workforce globalization, where developers from low-resource language backgrounds make substantial contributions to the global software ecosystem. The growing linguistic diversity in software development presents unique challenges, as most programming resources, documentation, and now AI interactions are predominantly in English (Guo 2018). This language barrier has historically created disparities in technology adoption and usage (Neeley 2013), potentially limiting the benefits of GenAI for speakers of low-resource languages.

Our research sits at the intersection of these three critical trends: the productivity promise enabled by GenAI, the persistent challenge of language barriers in the adoption and effective use of technologies, and the increasingly global nature of software development. By examining how English proficiency affects developers' ability to leverage GenAI tools effectively, we provide novel insights into how language barriers might exacerbate digital divides in an AI-powered development landscape, complementing research on how AI reshapes labor structures across different skill levels (Xue et al. 2022). It's worth noting that while some non-English languages like Mandarin have achieved high-resource status with extensive digital repositories, specialized NLP research, and dedicated AI development, many other languages remain severely underrepresented. Rather than advocating for language-specific models for each underrepresented language—an approach that would be resource-intensive and potentially impractical for true LRLs—our research demonstrates the effectiveness of bridging the gap through structured intermediate representations that leverage developers' existing domain knowledge.

Our findings demonstrate that lower English proficiency significantly impairs developers' ability to effectively utilize GenAI tools. When using traditional prompting approaches, developers with lower English proficiency (defined as one standard deviation below the mean) experience substantial disadvantages: they require 22.43% more time to complete tasks, achieve accuracy scores that are 13.61% lower, and are 79.95% more likely to abandon tasks compared to their high-proficiency

counterparts (defined as one standard deviation above the mean). These performance gaps align with prior research documenting how language barriers create cognitive burdens in global software development (Guo 2018) and technology-mediated communication (Neeley 2013). However, we find that metadata prompting substantially mitigates these disparities, a novel insight since the existing literature has primarily focused on documenting language barriers rather than developing solutions for them. When using metadata prompting, the performance differences between developers with varying English proficiency levels become statistically insignificant across all metrics.

This finding extends research on reducing barriers in global work (Fayard and Metiu 2014) by demonstrating that structured interaction patterns aligned with developers' technical expertise can effectively overcome language-based performance gaps, particularly important for those working in low-resource language contexts. The effectiveness of this approach suggests that rather than trying to improve English language skills, develop better translation tools, or create specialized LLMs for each low-resource language, organizations might achieve better results by redesigning AI interfaces to leverage developers' existing domain knowledge. Our exploration of user preferences further reveals that developers strongly favor metadata prompting over traditional approaches, with this preference intensifying at lower English proficiency levels. This finding aligns with research that emphasizes perceived ease of use as a crucial driver of adoption (Davis 1989) and recent findings about how structured prompting has significant gains over traditional ones (Kumar et al. 2023).

Our findings contribute to the emerging literature on the "AI divide" (McElheran et al. 2024), which documents how AI technologies may create new forms of inequality. While much of this literature focuses on organizational and geographical disparities in AI adoption, our research highlights how interface design choices can either exacerbate or mitigate these divides at the individual level.

The idea of metadata prompting builds on the idea of structuring AI interactions around users' domain expertise, rather than natural language capabilities (e.g., Lehmann et al. 2023). Naturally, we believe that these findings extend beyond the immediate context of AI-assisted programming. As organizations increasingly adopt GenAI tools across various domains, our results highlight the importance of considering language barriers in the design and implementation of AI interfaces.

6.1. Contributions

Our study offers several key contributions at the intersection of GenAI adoption, digital inequality, and human-centered interface design:

First, we highlight a critical yet underexplored dimension of the digital divide: the impact of English language proficiency on developers' ability to leverage AI tools. This issue is particularly consequential for low-resource language regions—many of which are economically disadvantaged or part of the Global South—where developers are active contributors to the global software ecosystem. Without intervention, these language-based disparities risk excluding large segments of the global developer population from the productivity gains promised by GenAI. **Second**, we empirically validate the effects of this divide through a large-scale field experiment embedded in a real-world hiring process. We show that developers with lower English proficiency are significantly more likely to abandon tasks, take longer to complete them, and produce lower-quality code when using traditional natural language prompts—clear evidence that language barriers have meaningful consequences for AI-supported work. **Third**, we demonstrate that a JSON-based metadata prompting approach, designed around developers' technical fluency, substantially mitigates these disparities. This structured format reduces cognitive load, lowers the inhibition caused by language barriers, and enables more effective interaction with GenAI tools. **Fourth**, our findings are grounded in a field setting, not a lab experiment. Because the tasks were part of an actual recruitment process, participants were highly motivated to perform at their best. This enhances the ecological validity of our results and addresses common limitations around generalizability in experimental research involving student samples. **Fifth**, we contribute to the prompting literature by showing that the benefits of structured prompting extend beyond low-proficiency users. While the performance gap closes most dramatically for developers with limited English proficiency, we find that even high-proficiency users perform better with metadata prompts—suggesting that the design principle behind structured interaction can enhance AI usability for all users, not just those facing linguistic challenges. **Finally**, we capture developers' subjective preferences after they used

both prompting formats. Across all levels of English proficiency, participants expressed a strong preference for the metadata approach, with the preference intensifying among lower-proficiency users. This underscores the perceived usability benefits of structured prompting and offers actionable guidance for AI platform designers seeking to drive adoption in multilingual, global settings.

6.2. Practical Implications

Our study makes several important practical implications for organizations seeking to maximize the benefits of GenAI tools across linguistically diverse teams.

For organizations adopting GenAI tools, our findings emphasize that thoughtful interface design can yield significant improvements without requiring extensive model fine-tuning or language-specific AI development. Rather than investing in specialized LLMs for low-resource languages—an approach that would be both resource-intensive and technically challenging—organizations can achieve more immediate and cost-effective results by implementing structured interaction frameworks that leverage developers' existing domain expertise. The precision, efficiency, and controlled nature of metadata prompting appears to align with developers' technical mindsets, making it an attractive option for *all* users regardless of linguistic background. This insight aligns with recent frameworks for design-oriented AI research (Abbasi et al. 2024) and extends proven approaches (Lehmann et al. 2023) to professional development contexts.

For AI platform developers, our results concrete design modifications to increase global accessibility. Platforms should prioritize building native support for structured prompting patterns alongside traditional natural language interfaces. This could include template systems for common tasks, schema validators, constrained solvers, and interactive tools that help users construct effective structured prompts without requiring natural language fluency.

Finally, the implications extend beyond software firms. Online freelancer marketplace platforms also stand to benefit significantly from implementing metadata prompting tools. Online marketplaces, which heavily rely on international freelancers,[8] can enhance their operations by leveraging

[8] Upwork Revenue and Client Stats - Back Linko

structured requirements specification tools, particularly considering the differential impact of AI adoption on labor demand across education and sectors (Xue et al. 2022). First, they could implement structured requirements specification interfaces that can guide clients to articulate their project needs in a way that reduces ambiguity and potential language-related misunderstandings. They could also develop AI tools to help freelancers interpret these structured requirements to provide an appropriate solution. This may not only overcome language-related hesitations in bidding for projects but also may lead to efficient project completion and higher client satisfaction.

6.3. Avenues for Future Research

While our study demonstrates the effectiveness of metadata prompting in the context of software development, several limitations suggest avenues for future research. First, our experiment focused specifically on MERN stack development tasks, where developers are well-versed with JSON-style formatting. However, such a formatting may not be intuitive to programmers that are well-versed in statistical programs such as Stata or R. Future work should examine whether similar benefits extend to other programming domains and task types.

Second, while we measured English proficiency through standardized tests, future studies might explore how different aspects of language proficiency (reading, writing, speaking) differently affect AI tool usage, especially for developers from low-resource language backgrounds where these skills may develop unevenly due to limited exposure opportunities.

Additionally, future research could explore how metadata prompting might be enhanced through automated tools that help developers structure their prompts, or through hybrid approaches that combine structured and natural language elements. This work could build on emerging frameworks to guide AI agent behavior (Ni et al. 2025) and prior research showing how conceptual modeling can improve machine learning performance (Castellanos et al. 2021), exploring how prompting techniques could be adapted for non-programming contexts where language barriers affect technology adoption. For instance, studies could explore how metadata prompting could help address other aspects of algorithmic bias (Fu et al. 2020) by reducing systemic advantages for certain linguistic groups in AI tool adoption.

References

Abbasi A, Parsons J, Pant G, Sheng ORL, Sarker S (2024) Pathways for design research on artificial intelligence. *Information Systems Research* 35(2):441–459.

Adams O, Makarucha A, Neubig G, Bird S, Cohn T (2017) Cross-lingual word embeddings for low-resource language modeling. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 937–947.

Ahia O, Kumar S, Gonen H, Kasai J, Mortensen DR, Smith NA, Tsvetkov Y (2023) Do all languages cost the same? tokenization in the era of commercial language models. *EMNLP*, 9904–9923, URL `https://doi.org/10.18653/v1/2023.emnlp-main.614`.

Ahuja K, Diddee H, Hada R, Ochieng M, Ramesh K, Jain P, Nambi A, Ganu T, Segal S, Ahmed M, et al. (2023) Mega: Multilingual evaluation of generative ai. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 4232–4267.

Boussioux L, Lane JN, Zhang M, Jacimovic V, Lakhani KR (2024) The crowdless future? generative ai and creative problem-solving. *Organization Science* 35(5):1589–1607.

Brynjolfsson E, Li D, Raymond LR (2023) Generative ai at work. Technical report, National Bureau of Economic Research.

Burtch G, Lee D, Chen Z (2024) The consequences of generative ai for online knowledge communities. *Scientific Reports* 14(1):10413.

Castellanos A, Castillo A, Tremblay MC, Lukyanenko R, Parsons J, Storey VC (2021) Improving machine learning performance using conceptual modeling. *AAAI spring symposium: Combining machine learning with knowledge engineering*.

Chen Z, Chan J (2024) Large language model in creative work: The role of collaboration modality and user expertise. *Management Science* .

Cheng X, Mayya R, Sedoc J (2024) From human annotation to llms: Silicon annotation workflow for management research. *arXiv preprint arXiv:2412.14461* .

Chronopoulou A, Peters ME, Dodge J (2022) Efficient hierarchical domain adaptation for pretrained language models. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1336–1351.

Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, Grave É, Ott M, Zettlemoyer L, Stoyanov V (2020) Unsupervised cross-lingual representation learning at scale. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8440–8451.

Cui ZK, Demirer M, Jaffe S, Musolff L, Peng S, Salz T (2024) The effects of generative ai on high skilled work: Evidence from three field experiments with software developers. *Available at SSRN 4945566* .

Davis FD (1989) Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly* 319–340.

de Kok T (2025) Chatgpt for textual analysis? how to use generative llms in accounting research. *Management Science* .

Doshi AR, Hauser OP (2024) Generative ai enhances individual creativity but reduces the collective diversity of novel content. *Science Advances* 10(28):eadn5290.

Eloundou T, Manning S, Mishkin P, Rock D (2024) Gpts are gpts: Labor market impact potential of llms. *Science* 384(6702):1306–1308.

Fayard AL, Metiu A (2014) The role of writing in distributed collaboration. *Organization Science* 25(5):1391–1413.

Fedorenko E, Ivanova A, Dhamala R, Bers MU (2019) The language of programming: a cognitive perspective. *Trends in cognitive sciences* 23(7):525–528.

Fu R, Huang Y, Singh PV (2020) Artificial intelligence and algorithmic bias: Source, detection, mitigation, and implications. *Pushing the Boundaries: Frontiers in Impactful OR/OM Research*, 39–63 (INFORMS).

Guo PJ (2018) Non-native english speakers learning computer programming: Barriers, desires, and design opportunities. *Proceedings of the 2018 CHI conference on human factors in computing systems*, 1–14.

Hinds PJ, Neeley TB, Cramton CD (2014) Language as a lightning rod: Power contests, emotion regulation, and subgroup dynamics in global teams. *Journal of International Business Studies* 45:536–561.

Horton JJ (2023) Large language models as simulated economic agents: What can we learn from homo silicus? Technical report, National Bureau of Economic Research.

Jiang Z, Xu FF, Araki J, Neubig G (2020) How can we know what language models know? *Transactions of the Association for Computational Linguistics* 8:423–438.

Jones B, Kenward MG (2003) *Design and analysis of cross-over trials* (Chapman and Hall/CRC).

Joshi P, Santy S, Budhiraja A, Bali K, Choudhury M (2020) The state and fate of linguistic diversity and inclusion in the nlp world. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6282–6293.

Karten S, Tucker M, Li H, Kailas S, Lewis M, Sycara K (2023) Interpretable learned emergent communication for human–agent teams. *IEEE Transactions on Cognitive and Developmental Systems* 15(4):1801–1811.

Kesseli P, O'Hearn P, Cabral RS (2025) Logic. py: Bridging the gap between llms and constraint solvers. *arXiv preprint arXiv:2502.15776* .

Kojima T, Gu SS, Reid M, Matsuo Y, Iwasawa Y (2022) Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35:22199–22213.

Kumar H, Rothschild DM, Goldstein DG, Hofman JM (2023) Math education with large language models: Peril or promise? *Available at SSRN 4641653* .

Lehmann J, Gattogi P, Bhandiwad D, Ferré S, Vahdati S (2023) Language models as controlled natural language semantic parsers for knowledge graph question answering. *ECAI 2023*, 1348–1356 (IOS Press).

Linck JA, Kroll JF, Sunderman G (2009) Losing access to the native language while immersed in a second language: Evidence for the role of inhibition in second-language learning. *Psychological science* 20(12):1507–1515.

Liu J, Xu X, Nan X, Li Y, Tan Y (2023a) " generate" the future of work through ai: Empirical evidence from online labor markets. *arXiv preprint arXiv:2308.05201* .

Liu P, Yuan W, Fu J, Jiang Z, Hayashi H, Neubig G (2023b) Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys* 55(9):1–35.

Lundqvist M, Herman P, Miller EK (2018) Working memory: delay activity, yes! persistent activity? maybe not. *Journal of neuroscience* 38(32):7013–7019.

Ma L, Xu X, He Y, Tan Y (2024) Learning to adopt generative ai. *arXiv preprint arXiv:2410.19806* .

McElheran K, Li JF, Brynjolfsson E, Kroff Z, Dinlersoz E, Foster L, Zolas N (2024) Ai adoption in america: Who, what, and where. *Journal of Economics & Management Strategy* 33(2):375–415.

Neeley T, Hinds PJ, Cramton CD (2009) Walking through jelly: Language proficiency, emotions, and disrupted collaboration in global work. *Working Paper* .

Neeley TB (2013) Language matters: Status loss and achieved status distinctions in global organizations. *Organization Science* 24(2):476–497.

Ni Y, Xu T, Li C, Zheng E, Gu B (2025) Machine behaviorism: Exploring the behavioral dynamics of large language models in decision making. *Proceedings of the 58th Hawaii International Conference on System Sciences*, 4199–4208.

Noll J, Beecham S, Richardson I (2011) Global software development and collaboration: barriers and solutions. *ACM inroads* 1(3):66–78.

Noy S, Zhang W (2023) Experimental evidence on the productivity effects of generative artificial intelligence. *Science* 381(6654):187–192.

Paas F, Renkl A, Sweller J (2003) Cognitive load theory and instructional design: Recent developments. *Educational psychologist* 38(1):1–4.

Pangakis N, Wolken S, Fasching N (2023) Automated annotation with generative ai requires validation. *arXiv preprint arXiv:2306.00176* .

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

Qiu L, Singh PV, Srinivasan K (2023) Consumer risk preferences elicitation from large language models. *Available at SSRN 4526072* .

Quinn M, Gutt D (2025) Heterogeneous effects of generative ai on knowledge seeking in online communities. *Journal of Management Information Systems* .

Raisch S, Krakowski S (2021) Artificial intelligence and management: The automation–augmentation paradox. *Academy of management review* 46(1):192–210.

Robinson P (2001) Task complexity, task difficulty, and task production: Exploring interactions in a componential framework. *Applied linguistics* 22(1):27–57.

Roussel S, Joulia D, Tricot A, Sweller J (2017) Learning subject content through a foreign language should not ignore human cognitive architecture: A cognitive load theory approach. *Learning and Instruction* 52:69–79.

Shi F, Suzgun M, Freitag M, Wang X, Srivats S, Vosoughi S, Chung HW, Tay Y, Ruder S, Zhou D, Das D, Wei J (2023) Language models are multilingual chain-of-thought reasoners. *The Eleventh International Conference on Learning Representations*, URL `https://openreview.net/forum?id=fR3wGCk-IXp`.

Song F, Agarwal A, Wen W (2023) Assessing generative ai's impact on collaborative software development productivity. *Available at SSRN 4856935* .

Sweller J (1988) Cognitive load during problem solving: Effects on learning. *Cognitive science* 12(2):257–285.

Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D, et al. (2022) Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35:24824–24837.

Xue M, Cao X, Feng X, Gu B, Zhang Y (2022) The effect of artificial intelligence (ai) on firm labor structure. *Proceedings of the 55th Hawaii International Conference on System Sciences*, 7088–7097.

Yeverechyahu D, Mayya R, Oestreicher-Singer G (2024) The impact of large language models on open-source innovation: Evidence from github copilot. *arXiv preprint arXiv:2409.08379* .

Zhang R, Duan W, Flathmann C, McNeese N, Freeman G, Williams A (2023) Investigating ai teammate communication strategies and their impact in human-ai teams for effective teamwork. *Proceedings of the ACM on Human-Computer Interaction* 7(CSCW2):1–31.

Zheng Q, Xia X, Zou X, Dong Y, Wang S, Xue Y, Shen L, Wang Z, Wang A, Li Y, et al. (2023) Codegeex: A pre-trained model for code generation with multilingual benchmarking on humaneval-x. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 5673–5684.

Zhong T, Yang Z, Liu Z, Zhang R, Liu Y, Sun H, Pan Y, Li Y, Zhou Y, Jiang H, et al. (2024) Opportunities and challenges of large language models for low-resource languages in humanities research. *arXiv preprint arXiv:2412.04497* .

Zhou D, Schärli N, Hou L, Wei J, Scales N, Wang X, Schuurmans D, Cui C, Bousquet O, Le Q, et al. (2022) Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625* .

Zhou E, Lee D (2024) Generative artificial intelligence, human creativity, and art. *PNAS nexus* 3(3):pgae052.

Zoph B, Yuret D, May J, Knight K (2016) Transfer learning for low-resource neural machine translation. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1568–1575.

Appendix. Tables and Figures

Table A1 Learning Effects Analysis

	Accuracy	Completion Time
English Proficiency	43.165***	-49.012***
	(4.594)	(4.818)
Metadata Prompting	42.131***	-46.243***
	(3.490)	(3.627)
Test2	2.642	0.357
	(3.824)	(3.938)
English Proficiency ×	-40.395***	45.731***
Metadata Prompting	(5.992)	(6.189)
English Proficiency ×	-2.991	-0.230
Test2	(6.474)	(6.708)
Metadata Prompting ×	-5.797	-2.584
Test2	(5.248)	(5.264)
English Proficiency ×	8.225	3.668
Metadata Prompting × Test2	(9.032)	(9.052)
Controls	Yes	Yes
Observations	1,347	1,347
Adjusted R-squared	0.539	0.572

Notes: Huber-White heteroskedasticity-consistent errors in parentheses
* p < 0.10, ** p < 0.05, *** p < 0.01
Control variables include coding ability, undergraduate grades, AI experience, gender, education, experience, and task fixed effects

Table A2 Alternative Accuracy Measures

	Original Result	First Attempt	Second Attempt	Maximum
English Proficiency	41.224***	40.178***	42.271***	41.260***
	(3.319)	(3.165)	(3.671)	(3.282)
Metadata Prompting	39.473***	39.301***	39.645***	39.273***
	(2.621)	(2.520)	(2.894)	(2.576)
English Proficiency ×	-36.664***	-36.553***	-36.775***	-36.170***
Metadata Prompting	(4.512)	(4.341)	(4.990)	(4.435)
Controls	Yes	Yes	Yes	Yes
Observations	1,347	1,347	1,347	1,347
Adjusted R-squared	0.539	0.553	0.494	0.546

Notes: Huber-White heteroskedasticity-consistent errors in parentheses
* p < 0.10, ** p < 0.05, *** p < 0.01
Control variables include coding ability, undergraduate grades, AI experience, gender, education, experience, task fixed effects, and test fixed effects.
Column (1) replicates column (3) of Table 5.

Table A3 Sample Selection Analysis

	Accuracy	Completion Time
English Proficiency	41.331***	-49.103***
	(3.327)	(3.422)
Metadata Prompting	39.079***	-47.903***
	(2.642)	(2.644)
English Proficiency ×	-36.216***	48.023***
Metadata Prompting	(4.553)	(4.553)
Controls	Yes	Yes
Observations	1,312	1,312
Adjusted R-squared	0.538	0.578

Notes: Huber-White heteroskedasticity-consistent errors in parentheses
* p < 0.10, ** p < 0.05, *** p < 0.01
Sample excludes all observations for individuals who abandoned in Test 2
Control variables include coding ability, undergraduate grades, AI experience, gender, education, experience, and task fixed effects

Table A4 Alternative Model Specifications

	Linear	Probit	Poisson	Neg. Binomial
	Abandonment		Completion Time	
	(1)	(2)	(3)	(4)
English Proficiency	-0.939***	-2.601***	-0.897***	-0.910***
	(0.195)	(0.575)	(0.062)	(0.064)
Metadata Prompting	-0.807***	-2.385***	-0.937***	-0.944***
	(0.142)	(0.462)	(0.061)	(0.062)
English Proficiency ×	1.085***	3.190***	0.835***	0.848***
Metadata Prompting	(0.246)	(0.810)	(0.107)	(0.108)
Controls	Yes	Yes	Yes	Yes
Observations	965	965	1,347	1,347
Adjusted/Pseudo R^2	0.076	–	–	–
χ^2	–	76.826	1756.760	1735.076

Notes: Huber-White heteroskedasticity-consistent errors in parentheses
* p < 0.10, ** p < 0.05, *** p < 0.01
Control variables include coding ability, undergraduate grades, AI experience, gender, education, experience, and task fixed effects

Table A5 Post-hoc Tukey Test Results for Group Comparisons

Comparison	Mean Difference	Standard Error	p-value
Panel A: Preferred Future Use			
Medium vs Low	-0.366***	0.100	0.001
High vs Low	-0.662***	0.108	0.000
High vs Medium	-0.296**	0.110	0.021
Panel B: Confidence in Prompting			
Medium vs Low	-0.317***	0.095	0.003
High vs Low	-0.747***	0.103	0.000
High vs Medium	-0.430***	0.105	0.000
Panel C: Expected Effectiveness			
Medium vs Low	-0.289**	0.104	0.015
High vs Low	-0.596***	0.112	0.000
High vs Medium	-0.307**	0.115	0.021

Notes: * $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$
All measures use 5-point scale (1 = prefer regular prompting, 5 = prefer metadata prompting)
Low: bottom tercile; Medium: middle tercile; High: top tercile