Barriers that Programming Instructors Face While Performing Emergency Pedagogical Design to Shape Student-AI Interactions with Generative AI Tools

Sam Lau
University of California San Diego
La Jolla, USA
lau@ucsd.edu

Kianoosh Boroojeni
Florida International University
Miami, USA
kborooje@fiu.edu

Harry Keeling
Howard University
Washington, DC, USA
hkeeling@howard.edu

Jenn Marroquin
Google
Austin, USA
zap@google.com

Abstract

Generative AI (GenAI) tools are increasingly pervasive, pushing instructors to redesign how students use GenAI tools in coursework. We conceptualize this work as *emergency pedagogical design*: reactive, indirect efforts by instructors to shape student-AI interactions without control over commercial interfaces. To understand practices of lead users conducting emergency pedagogical design, we conducted interviews (*n*=13) and a survey (*n*=169) of computing instructors. These instructors repeatedly encountered five barriers: *fragmented buy-in* for revising courses; *policy crosswinds* from non-prescriptive institutional guidance; *implementation challenges* as instructors attempt interventions; *assessment misfit* as student-AI interactions are only partially visible to instructors; and *lack of resources*, including time, staffing, and paid tool access. We use these findings to present emergency pedagogical design as a distinct design setting for HCI and outline recommendations for HCI researchers, academic institutions, and organizations to effectively support instructors in adapting courses to GenAI.

CCS Concepts

• **Human-centered computing → Empirical studies in HCI**.

Keywords

emergency pedagogical design, student-AI interaction, programming instructors, generative AI

ACM Reference Format:
Sam Lau, Kianoosh Boroojeni, Harry Keeling, and Jenn Marroquin. 2026. Barriers that Programming Instructors Face While Performing Emergency Pedagogical Design to Shape Student-AI Interactions with Generative AI Tools. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3772318.3790682

CHI '26, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/2026/04
https://doi.org/10.1145/3772318.3790682

1 Introduction

Instructors around the world are reacting to the availability of highly capable generative artificial intelligence (GenAI) tools like ChatGPT [48], Claude [25], and Gemini [51], which have raised questions around the value of formative assessments like take-home essays and programming assignments that can easily be completed by these tools [72]. Students already interact with GenAI via freely available GenAI applications, sometimes even without intending to do so because GenAI outputs are now included by default in web search results [57], word processing software [66], and code editors [30]. Since unrestricted use of GenAI tools can hinder student learning [7, 80], instructors are now finding themselves in the unenviable position of trying to structure student interactions with GenAI tools that weren't originally designed for instruction.

Drawing an analogy to emergency remote teaching where instructors needed to abruptly offer courses online in response to the COVID-19 pandemic [29], we characterize the work that instructors are now performing as *emergency pedagogical design*. In addition to all of the previous responsibilities of running a course – creating course materials, lecturing, grading, and managing course staff – instructors now suddenly need to encourage students to interact with GenAI tools in ways that help rather than hinder learning. For example, some instructors have banned GenAI usage entirely [32], while others have found it valuable to deliberately demonstrate its limitations [60]. One clear similarity between emergency remote teaching and emergency pedagogical design is that both arose in response to a disruptive event: in the former, a global pandemic, and in the latter, the creation of free-to-use, highly capable GenAI tools. One notable difference is that emergency remote teaching was generally considered to be a temporary shift in teaching modality until the COVID-19 virus was under control, but there is general consensus that GenAI tools will have a role in education even in the long-term [6]. Thus, by studying instructors' current responses to GenAI tools, we can deepen understanding of the practices that may soon be normalized in everyday instruction.

This paper examines undergraduate computing instructors as a critical early case where GenAI's impact arrived to observe emergency pedagogical design in practice. In particular, in this setting we consider computing instructors to be lead users [71] – users who are trying to design solutions for their own experienced needs. For example, computing instructors are attempting to shape student-AI interactions by creating alternative interfaces that wrap off-the-shelf GenAI tools (e.g. [35, 41]), making them a potential source of need-forecasting for instructors in other domains. This leads to our central research question: **How do computing instructors conduct emergency pedagogical design, and what barriers do they encounter?**

To address this question, we interviewed 13 computing instructors who had made substantial attempts to influence student-AI interactions by integrating GenAI into their course assignments, assessments, and infrastructure. To gauge broader patterns among a more diverse set of computing instructors, including instructors at Minority-Serving Institutions (MSIs) and Historically Black Colleges and Universities (HBCUs), we also surveyed 169 instructors about their perspectives towards integrating GenAI tools into their courses. These instructors repeatedly encountered five barriers as they engaged in emergency pedagogical design: *fragmented buy-in* for revising courses; *policy crosswinds* from non-prescriptive institutional guidance; *implementation challenges* as instructors attempt interventions; *assessment misfit* as student-AI interactions are only partially visible to instructors; and *lack of resources*, including time, staffing, and paid tool access. This study was conducted in mid-2025, about two and a half years after the initial release of well-known GenAI tools like ChatGPT and GitHub Copilot. Thus, our work captures a rare moment in time where computing instructors have been able to make initial attempts to shape student-AI interactions, but approaches have not yet converged on a set of best practices. To spur the community to use this unique opportunity to influence how instructors approach student-AI interaction, we conclude this paper with a set of open research questions for HCI (Section 5.1), recommendations for academic institutions and funders (Section 5.2), and a reflection on lessons learned from emergency remote teaching that may also apply in this context (Section 5.3).

In sum, this paper makes the following contributions:

(1) Conceptualizing emergency pedagogical design through an empirical study of computing instructors.
(2) Documenting barriers that instructors encounter as they conduct emergency pedagogical design.
(3) Highlighting implications for HCI researchers, academic institutions, and funding agencies who wish to support instructors in adapting their courses to GenAI.

2 Related Work

We organize our review of related work into two areas. We first examine the broad impact of GenAI on computing education, including instructor perceptions and reactions. We then review HCI and computing education research that has developed tools and pedagogical approaches to support productive GenAI use.

2.1 GenAI Adoption and Policy Responses in Computing Education

The rapid development of GenAI models has created both opportunities and challenges for computing education. In particular, the launch of AI models that can generate code like GitHub Copilot [24] in 2021 and ChatGPT in 2022 [48] sparked a flurry of research that found that these tools could solve problems across the undergraduate computing curricula, for example in introductory programming courses [19, 56, 62, 73, 65, 61], data structures courses [20], programming competitions [40], and software engineering courses [11]. Instructors reacted to this in the short-term by banning the use of GenAI tools in their course policies and reduced the weight of non-proctored assessments [38]. As these tools became widely accessible, however, instructors increasingly acknowledged that simply changing course policies was not enough; the content and learning goals of their courses also needed to adapt. This change in perspective was motivated by observations that students used GenAI tools regardless of policy [3, 72, 14], that GenAI coding assistants had already become commonplace in industry [47, 70], and that GenAI tools could provide new opportunities to improve programming pedagogy [54, 16, 23, 55].

Some instructors have begun to integrate GenAI directly into their courses. For instance, Vadaparty et al. redesigned an introductory programming course to incorporate GenAI from the start [69], and Benario et al. did the same for a software engineering course [26]. However, efforts like this seem to be the exception, not the norm. In a survey of computing instructors in 2024, 75% of surveyed faculty believed that their courses' learning objectives should change in response to GenAI, yet only 35% reported actually incorporating GenAI into their courses [53]. A similar pattern appears outside computer science: a global survey conducted in early 2025 reported that 88% of faculty use AI sparingly or not at all, even though 86% expect to use AI for teaching in the future [13]. Reported reasons for this gap include distrust of GenAI [8, 43, 10, 50], limited understanding of its capabilities [67], concerns about effects on learning outcomes [79, 22, 28], and concern for potential student overreliance [63, 27, 17, 13].

In contrast to this line of prior work that focused on instructor intentions or perceptions, we studied what instructors actually did in practice to change their course materials and report experienced barriers in addition to perceived barriers.

2.2 Systems that Support Learning to Program with GenAI

Since GenAI use for programming has become more common in the past few years, research on how novices learn programming with GenAI is in its early stages with both positive and negative results. For instance, Kazemitabaar et al. found that using a GenAI coding assistant can improve novices' ability to write code even on later tasks without access to GenAI [33]. However, GenAI tools impose additional metacognitive requirements on users [68], which can reinforce learning challenges for struggling students in the context of programming [52].

HCI research has proposed system designs to mitigate the potential harms of GenAI coding tools while maximizing their benefits for programming learners. For example, GenAI coding tools are often designed to quickly produce working code [31], not to help learners understand how programs work. In response, researchers have proposed GenAI tools with guardrails that offer conceptual help rather than writing code [35, 41], and that ask students to "teach" a chatbot [59]. To improve user understanding of code, Ivie adds lightweight explanations of generated code [75], Leap surfaces live runtime values [18], and WaitGPT visualizes the flow of data through a program [74].

Novices can struggle with writing effective prompts [78] and often omit important detail in their prompts when trying to generate code [42]. In contrast, systems that guide users to decompose problems can increase perceived control and ease of use [34]. Along this line of work, DBox guides learners to decompose their algorithms before writing code[44], CoLadder provides an interface to manipulate a tree of prompts and generated code [77], iGPT iteratively asks followup questions to learners to produce more complete program generation plans [76], and Prompt Problems automatically grade learner prompts by checking generated code against test cases [15]. Some systems even remove the need for prompting altogether by offering proactive suggestions [9, 36] or by offering a simplified interface to generate hints [58].

Together, this prior work advances systems that aim to improve student-AI interactions. We complement these system-centric contributions by framing instructors' adoption efforts as the process of emergency pedagogical design, and by identifying instructor needs that past work has not yet addressed in order to motivate further HCI research.

3 Methodology

To understand the characteristics of emergency pedagogical design, we conducted semi-structured interviews with 13 full-time instructors teaching undergraduate-level computing courses. We were specifically interested in understanding prior experiences of designing student-AI interactions rather than merely perceptions of possible interventions. To accomplish this, we selected interviewees who self-reported that they had made changes to their course materials because of student-AI usage, and had also taught these updated materials to students already. We filtered out instructors who had not yet made changes to their course or had only made changes to their course policies, since we felt this group of instructors had already been sufficiently studied in prior work (e.g. [38]). Interviewees were recruited via professional networks, email lists for computing instructors, and personal contacts, using snowball and purposive sampling [49] that was iterative until saturation. The interviews were conducted in May and June 2025. All interviews were held in English and participants were not financially compensated for participating in the study.

3.1 Interview Protocol

Each interview was conducted over Zoom by one researcher, lasted between 45-60 minutes, and was audio-recorded upon obtaining consent. Our interview protocol began with three background questions:

(1) What programming courses have you most recently taught?
(2) From your perspective, have GenAI tools like GitHub Copilot and ChatGPT had an impact on this course? If so, what kinds of impact have you observed?
(3) Does your course or department have a formal policy regarding student use of GenAI tools?

The purpose of these background questions was to establish the context of our interviewees' instruction as of mid-2025, approximately 2.5 years after the initial release of popular GenAI tools like ChatGPT and GitHub Copilot. These questions led into the primary open-ended scenario which took up the majority of our interviews:

> Please open and screen-share course materials from a memorable assignment, assessment, or moment in your course where you incorporated GenAI tools. Could you explain how you wanted students to interact with GenAI in this part of the course?

As part of our protocol, we prepared followup questions to understand instructor motivation for making this change to their course materials, their experiences implementing and evaluating this change, and student reactions. We also asked interviewees followup questions to help elicit more thorough responses.

3.2 Rationale for Our Interview Protocol

Our protocol design was based on several theoretical considerations. Since we were specifically interested in what instructors actually did (in contrast to what they wanted to do), we drew from critical incident technique [21], a qualitative method for collecting and analyzing events that had substantial impact on outcomes. In our context, this means that we spent the majority of the interview asking instructors to reflect on the most memorable interventions that they performed in their courses, with the expectation that talking about vivid experiences would encourage interviewees to share more relevant details. Note that we did not define "memorable" as "positive"; we were equally interested in unsuccessful experiences attempting to design student-AI interactions and made this clear to our interviewees as part of our protocol.

We also grounded our conversations around concrete pieces of course materials, such as take-home assignments or exam questions. This part of the protocol was inspired by the cognitive walkthrough [45] methodology in HCI, which recommends discussing a shared artifact to help elicit greater quality and quantity of observations. To reduce the risk of interviewees fixating on minute details of their course materials, we also asked interviewees to make higher-level reflections after each line of questioning.

Lastly, when designing individual interview questions, we applied best practices to reduce bias, such as using neutral wording and avoiding jargon [1]. In our protocol, we also did not mention specific AI tools to avoid priming or anchoring biases, but if participants brought up a specific tool of their own accord, we asked followup questions about that tool.

3.3 Interview Participant Backgrounds

Our interviewee backgrounds are summarized in Table 1. We recruited 13 participants (5 non-male). Most of our participants taught in the United States (10/13), while others taught in Canada (1/13), New Zealand (1/13), and the United Kingdom (1/13). All participants held PhD degrees and were either tenure-line faculty (8/13) or nontenure-line faculty (5/13). With one exception that taught in an engineering department, all interviewees taught in computer science departments. Interviewees taught undergraduate courses at both introductory and upper levels, including introductory programming, data structures, algorithms, software engineering, and cybersecurity. There was considerable range in the number of years of full-time teaching experience (5–42 years, μ=17 years). Participants primarily taught at public PhD-granting universities (10/13),

Table 1: We interviewed 13 computing instructors to understand their experiences conducting emergency pedagogical design for their courses. Gender = Male (M), Female (F), Non-binary (NB). Years = number of years of experience as a full-time instructor. Each course listed in the "Courses discussed" column is marked with (L) to denote a lower-division course or (U) for an upper-division course.

ID	Gender	Country	Years	Type of university	Courses discussed
P01	F	US	12	Public PhD-granting	Algorithms (U)
P02	M	US	26	Private undergrad-only	Intro to programming (L)
P03	M	Canada	5	Public PhD-granting	Data structures (L)
P04	M	US	6	Public PhD-granting	Intro to programming (L), Software eng. (U)
P05	M	US	23	Public PhD-granting	Intro to programming (L)
P06	NB	US	11	Private PhD-granting	Intro to programming (L)
P07	M	US	28	Private PhD-granting	Software design (U)
P08	F	US	9	Public PhD-granting	Intro to programming (L)
P09	M	New Zealand	12	Public PhD-granting	Intro to programming (L)
P10	F	US	22	Public PhD-granting	Software engineering (U)
P11	F	US	9	Public PhD-granting	Intro to programming (L)
P12	M	US	14	Public PhD-granting	Web development (U)
P13	M	UK	42	Public PhD-granting	Cybersecurity (L)

while some taught at private PhD-granting universities (2/13) and undergraduate-only institutions (1/13).

3.4 Survey

Since our interviewees were selected specifically because they had already conducted interventions to improve student-AI interactions, we recognized that our interviews would likely emphasize perspectives of instructors who wanted to keep including GenAI usage as part of their courses. In order to reduce the impact of this limitation, in parallel with our interviews we conducted a survey to gather perspectives of computing faculty more broadly, including faculty that had not changed their course materials in response to GenAI usage. The survey was designed to be completed within 10 minutes and asked participants questions about their personal views, their perceptions of colleagues, and basic teaching demographics such as average number of courses taught per year and number of students taught per year.

The survey was distributed via email lists for computing faculty, discussion forums, and personal outreach. To capture viewpoints from more diverse faculty, we specifically contacted email lists for faculty at Minority-Serving Institutions (MSIs) [39] and Historically Black Colleges and Universities (HBCUs) [4]. A total of 169 faculty responded to the survey. More than half of these faculty taught at MSIs (51%), and a sizable proportion taught at HBCUs (17%). Most faculty either taught 1-2 courses per academic term (37%) or 3-4 courses per academic term (41%), although a small minority taught more than 5 courses per term (17%). Because the survey relied on convenience and snowball sampling across lists with unknown membership sizes, and because respondents could skip any item, we could not calculate a response rate; instead, we treat the results as descriptive of respondents rather than as estimates of wider faculty attitudes.

3.5 Data Analysis

During each interview, the researcher who conducted the interview took timestamped notes. Then, a second researcher independently listened to each audio recording and took their own set of notes. Both researchers met weekly to discuss their notes together. After the interviews were completed, we iteratively came up with themes using an inductive approach [12]. Both researchers independently annotated interview excerpts at the level of short utterances or question-answer turns, depending on where a shift in meaning occurred. We then compared annotations in weekly meetings. Coding proceeded as a negotiated process rather than through formal inter-rater reliability statistics: disagreements were discussed case by case, and codes were revised until both researchers reached consensus. This process unfolded over five weekly iterations, during which we merged, split, and renamed codes until the codebook stabilized. After convergence, we re-reviewed earlier excerpts to ensure consistent application. This iterative process led us to center each theme on barriers instructors encountered while attempting to alter student-AI interactions.

To analyze the survey data, two researchers computed counts and proportions for Likert-scale questions and applied an inductive approach to generate themes for open-ended responses. For open-ended items, both researchers independently labeled sentences within each response. We compared labels in weekly meetings and resolved disagreements through discussion until agreement was reached. Coding stabilized after three iterations, after which we re-reviewed earlier responses for consistency and grouped the agreed-upon codes into the themes reported in the results. Although we allowed survey participants to leave any question blank if they chose, we found that all of our Likert-scale questions had response rates over 90% (between 152-161 responses out of 169 submissions). As such, when reporting Likert-scale question results, we omit the number of responses for ease of reading.

3.6 Study Scope and Limitations

We purposively recruited interviewees who had already made changes to student-AI interactions beyond policy-only edits. This focus may bias the interviews toward instructors with more time, staffing, or institutional support and under-represent instructors who chose not to engage or were unable to do so. By asking for memorable incidents and successful or attempted interventions, the data may overweight vivid cases and underweight routine or abandoned efforts. Accounts are retrospective self-reports and thus subject to recall and social-desirability biases, so we treat quotations as situated evidence rather than measurements. Despite efforts to recruit broadly, interviews were conducted in English and were predominantly with U.S.-based faculty at research-intensive, PhD-granting institutions; transferability to community colleges, teaching-focused institutions, and non-U.S. contexts (including non-English instruction) may be limited.

The survey used convenience and snowball distribution via email lists, forums, and personal outreach rather than probability sampling. We did not weight responses and allowed missing items, so we cannot estimate population parameters or make inferential claims beyond respondents. This approach likely introduced self-selection biases: instructors with strong views about GenAI, whether positive or negative, may have been more motivated to respond. Although we sought to include a more diverse range of responses from MSIs and HBCUs, the sample is nonetheless not nationally representative. Because of these limitations, in this paper we treat survey results as descriptive of our respondents and use percentages mainly to compare or contrast with interview patterns rather than as prevalence estimates.

Because all interviewees taught programming-focused courses and the survey focused on computing faculty, our findings are more directly relevant within computing education and may not generalize to other domains. We attempted to frame our findings so that they have the potential to appear in other disciplines, but we acknowledge that instructors in writing-intensive, humanities, or social science contexts may face distinct challenges that our data do not capture. As such, transferability of these findings beyond computing requires caution, and we view this work as an early case in one discipline where GenAI impacted teaching practices rapidly.

4 Results

In this section we present a definition and properties of emergency pedagogical design as instructors adapted their courses to account for student GenAI usage (Section 4.1). During this design work, instructors faced five barriers that repeatedly arose during our analysis: fragmented buy-in (Section 4.2); policy crosswinds (Section 4.3); implementation challenges (Section 4.4); assessment misfit (Section 4.5); and lack of resources (Section 4.6).

Both the properties of emergency pedagogical design and the five barriers reported in this section were derived from the same inductive thematic analysis described in Section 3. During coding, we noted two kinds of recurring patterns: cross-cutting characteristics of the design setting that we refer to as properties of emergency pedagogical design, and recurring challenges that we grouped into five barriers. The properties characterize the conditions under which instructors worked, while the barriers describe the constraints that shaped their attempts to change student-AI interactions.

4.1 Overview of Emergency Pedagogical Design

In the context of our instructors reacting to student usage of GenAI tools, we define *emergency pedagogical design* as the work that instructors do to adjust their courses in order to improve student interactions with GenAI. In synthesizing interview accounts, we observed several cross-cutting patterns that described the broader conditions under which instructors tried to adjust their courses. We refer to these as properties of emergency pedagogical design because they characterize the design setting that emerged from our inductive analysis rather than specific challenges or solutions. We observed the following properties:

- **Reactive rather than proactive:** Instructors did not anticipate the capabilities or extent to which students would use GenAI tools without instructor intervention. Since the instructors in our study had been teaching existing courses, instructors generally had to retrofit course materials that were created before GenAI tools were widely used rather than write new materials from scratch that were designed to account for GenAI usage.
- **Indirect rather than direct:** Interaction designers typically have the ability to make direct changes to the interface they are designing. In contrast, instructors lack the ability to directly update interfaces of GenAI tools, since these tools are maintained by corporations rather than academic institutions. In addition, GenAI output is increasingly embedded within everyday interfaces (e.g., web search [57] and code editors [30]), making it challenging for instructors to anticipate all the ways that students can trigger GenAI usage, whether intentionally or inadvertently. Without the ability to change GenAI interfaces directly, instructors indirectly influence student interactions with GenAI through learning materials, course policies, assessment strategies, and course infrastructure.
- **Ambient evidence rather than orchestrated evaluations:** Rather than in-depth user studies, instrumentation, or A/B tests, instructors inferred effects from informal signals such as office-hour conversations, anecdotes from course staff, help-queue and autograder patterns, forum threads, and shifts in attendance.
- **Timely iteration rather than deferred certainty:** Instructors felt that they couldn't wait for cumulative research evidence or stable tools to emerge; they instead wanted to act quickly, making minimally viable policy and curricular changes and then iterating based on local signals.

Taken together, these properties portray emergency pedagogical design as reactive, indirect, and iteration-driven work conducted under uncertainty and with limited visibility into student behavior. In the following subsections, we elaborate on five recurring barriers that constrained how instructors carried out emergency pedagogical design within this setting, summarized in Table 2.

Table 2: Overview of the five barriers that emerged from our analysis of interview and survey data. The barriers summarize recurring constraints computing instructors encountered while engaging in emergency pedagogical design.

Barrier	Description	Representative Quote
Fragmented Buy-In (Section 4.2)	Instructors have limited personal bandwidth and faced mixed reactions from colleagues in their departments.	"For years, I've been trying to get my department to at least talk about getting students to use GenAI better! And we really haven't, amazingly enough." (P07)
Policy Crosswinds (Section 4.3)	Policies around GenAI could be dramatically different for individual courses, which led to student confusion.	"I have to keep telling my students that using ChatGPT is not cheating in my class. They're scared to use ChatGPT in front of me because other classes don't allow it." (P01)
Implementation Challenges (Section 4.4)	Lacking established pedagogical guidelines, instructors implemented experimental, bespoke approaches to shape student-AI interactions.	"Honestly, I'm still trying to figure it out, but I'm confident that introducing GenAI isn't something we should postpone. The longer we wait, the more bad habits students can accrue." (P08)
Assessment Misfit (Section 4.5)	The efficacy of interventions on both student-AI interactions and student learning was difficult to measure, even when telemetry data was available.	"I have hundreds of student logs [with my course's custom chatbot], but I don't have the time to analyze them all." (P04)
Lack Of Resources (Section 4.6)	Instructors had to work within strict time, staffing, and financial constraints which limited their ability to try new approaches.	"Without [external] financial support, these changes simply wouldn't have happened." (P09)

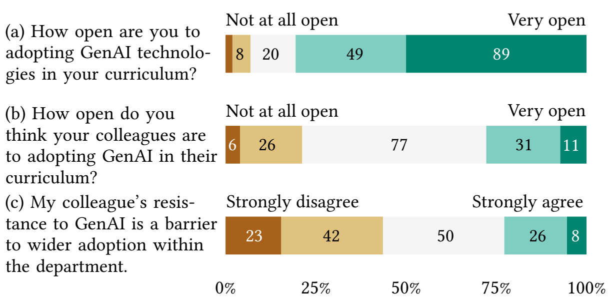

Figure 1: Although a strong majority of survey respondents considered themselves open to adopting GenAI technologies (a), only a minority felt their colleagues were equally open (b). However, only a few instructors perceived colleague resistance to GenAI as a barrier (c), suggesting that ambivalence was more common than direct opposition. Instructors responded on a 5-point scale. The number of responses for each option is displayed within each bar.

4.2 Fragmented Buy-In

We use *buy-in* to refer to two levels of commitment: an instructor's own motivation to revise course materials to shape student-AI interactions, and departmental support from colleagues that enables and rewards those revisions. We make this distinction because our survey results indicated a clear gap between instructors' own openness to adopting GenAI and their perceptions of departmental support. While 81% of respondents rated themselves as Open or Very Open to adopting GenAI technologies, only 28% said the same of their colleagues (Fig. 1a,b). Although 23% Agreed or Strongly Agreed that colleague resistance was a barrier, most respondents selected neutral options, suggesting ambivalence rather than direct opposition (Fig. 1c). These patterns indicate that instructors often felt individually willing to revise courses but did not perceive a shared departmental commitment to this work.

Since we selected for interviewees that had already made concrete changes to their course materials in response to student GenAI usage, a limitation of our interviews is that they only surface perspectives of instructors who had strong personal buy-in for making course changes. Even so, several interviewees described an initial period of reluctance or skepticism (P03-P06, P09-P10, P12). For example, one instructor mentioned that "I'd been trying to ignore [GenAI tools] for as long as I possibly could" by using proctored quizzes to deter students from using GenAI on assignments (P06). Despite having course policies forbidding GenAI, this instructor eventually realized that students used GenAI to complete assignments anyway, and then did poorly on proctored assessments – a pattern echoed by other interviewees (P04, P08, P12, P13). On reflection, P06 shared that "punishing them [via a proctored assessment] after the fact is not particularly useful or effective. So how can I get them to stop doing things they shouldn't be doing?" Other instructors cited a desire to curb potential "bad habits" when students used GenAI without instructor guidance (P07, P08).

Beyond personal buy-in, departmental buy-in was also fragmented. Interviewees reported that they were in a small minority within their departments who were meaningfully integrating AI into coursework (P01, P05, P07, P08, P11), a pattern that aligned

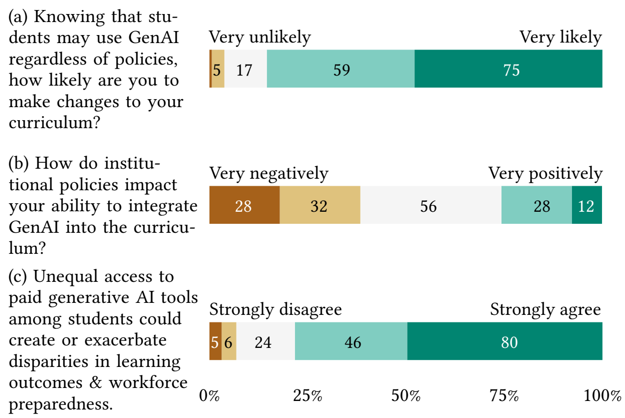

Figure 2: Surveyed instructors felt motivation to update their curriculum in response to student usage of GenAI (a). Institutional constraints had mixed perceived impact (b), yet unequal access to paid GenAI tools was a widely held concern (c). Instructors responded on a 5-point scale. The number of responses for each option is displayed within each bar.

with our survey data. Some even faced opposition from other faculty in their department about trying to include AI interactions in coursework (P05, P10-P12). The mixed reactions by other faculty in their departments reduced informal support, slowed coordination across multi-section courses, and made it harder to share materials or run aligned evaluations. This isolation also constrained scope: all of our interviewees piloted course-specific changes, but were not yet able to attain broader alignment with other courses via shared policies, common assignment patterns, or cross-course scaffolds.

Key Insights: Together, limited personal bandwidth and weak departmental support narrowed what was feasible. Instructors tended to adopt small, local changes they could own, rather than larger designs that required coordination, new tooling, or sustained evaluation.

4.3 Policy Crosswinds

Since a growing number of institutions have started to issue guidelines about GenAI usage in coursework [46], our survey asked computing instructors about the effects of these top-down policies on their adoption of GenAI. Knowing that students may use GenAI tools regardless of policies forbidding them, 85% of respondents said they were Likely or Very Likely to change their curriculum (Figure 2a). By contrast, when asked how institutional constraints or policies affected their ability to integrate GenAI, only 26% reported a Positive or Very Positive impact (Figure 2b). Respondents also expressed concerns about the uneven enforceability of institution-level policies, noting in open-ended responses that policies rarely specified how GenAI should be incorporated into assignments, how to interpret allowed versus disallowed uses, or how students' access to paid tools should be handled.

Likewise, nearly all interviewees reported that neither their department nor institution provided prescriptive policies governing GenAI use in teaching as a result of fragmented buy-in (P01-P08, P10-P13). In the absence of top-down policies, instructors had to decide their own policies on a per-course basis. As a result, practices diverged, sometimes dramatically. As one instructor summarized: "From a student perspective, it's the wild west. Some courses allow GenAI usage, some don't." (P07). Instructors pointed out that this caused "lots of confusion" (P01) for students who had to navigate different policies for every course that they took. Although several instructors mentioned departmental meetings to discuss GenAI policies, discussions had not yet converged shared guidelines or consistent approaches (P01, P02, P04, P05, P07, P10, P12). These inconsistencies also resulted in equity implications that current policies often did not address. For example, none of our interviewees reported course policies that distinguished between access to paid versus unpaid tools or between standalone chatbots and GenAI embedded in everyday software (e.g., code editor assistants or web search). The difference between paid and unpaid tools in particular was salient among survey respondents: 78% of responses Agreed or Strongly Agreed with the statement, "Unequal access to paid generative AI tools among students could create or exacerbate disparities in learning outcomes and workforce preparedness" (Figure 2c).

As one notable counterexample, P09 described a university "two-lane" policy that distinguished proctored from unproctored assessments. For proctored assessments, instructors could set GenAI rules; for unproctored assessments like take-home programming exercises, "students can use whatever tools they choose, and instructors cannot prohibit GenAI use" (P09). This instructor voiced support for this approach as a pragmatic way to align policy with what could be reasonably enforced outside proctored settings.

Key Insights: Inconsistent or absent guidance produces divergent course policies. Students must track different rules across classes, and course policies rarely account for paid versus unpaid tools or GenAI embedded in common software, raising equity concerns.

4.4 Implementation Challenges

Instructors wanted to shape how students used GenAI, but their leverage was indirect. They could adjust policies, task design, workflows, and course infrastructure, but not the commercial tools students actually used. We observed a spectrum of redesigns. At the lightest end, some instructors kept existing bans but permitted GenAI in specific programming labs or assignments (P03, P06, P13). At the most extensive end, others told students to use GenAI across most assignments, integrating it into routine student practice (P01, P02, P05, P08, P11). Some instructors even built custom interfaces for students that used GenAI tools as part of their implementation (P04, P09).

One way to frame instructor actions is to describe them as interventions that are intended to encourage certain student behaviors with GenAI tools. We highlight selected examples to demonstrate the diversity of instructor approaches:

- *Prompting as decomposition (P01).* Intervention: verbally encouraged students to use GenAI to generate small, concrete

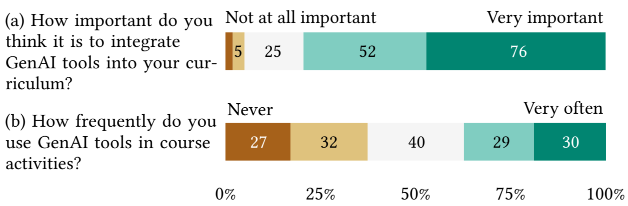

Figure 3: Surveyed instructors felt it was important to integrate GenAI tools into their curriculum (a), but fewer instructors reported actually doing so (b). Instructors responded on a 5-point scale. The number of responses for each option is displayed within each bar.

inputs for algorithms. Intended behavior: prompt at the level of subproblems rather than paste full problem statements.

- *Tasks that resist one-shot answers (P03).* Intervention: designed a lab with a malformed CSV and multiple valid approaches. Intended behavior: recognize that GenAI outputs can be unreliable and require validation.
- *Guardrailed conceptual help (P04).* Intervention: deployed a custom chatbot that guided reasoning but avoided giving code. Intended behavior: have students write code themselves while using GenAI for conceptual support.
- *Underspecified requirements and judgment (P06).* Intervention: included a problem where a critter's score must converge but "it may wiggle around a bit." Intended behavior: practice converting an ambiguous specification to a precise prompt.
- *Refactoring with intent (P07).* Intervention: showed that GenAI refactors better when students name target designs (e.g., "replace the if statements with an abstract interface"). Intended behavior: motivate learning of foundational concepts to work productively with GenAI.
- *Plan before you generate (P05, P08).* Intervention: required a design document before asking GenAI to implement. *Intended behavior:* keep planning and decomposition as student responsibility rather than outsourcing the whole project.
- *Explanations as first-class work (P09).* Intervention: built a pipeline that autograded code produced from student explanations of code via an AI tool. Intended behavior: position GenAI as a tool for learning activities (explaining, testing), not just writing code.
- *GenAI as a component, not a destination (P12).* Intervention: had students call GenAI as an API within a program. Intended behavior: view GenAI as one element in a larger system rather than a standalone chatbot.

We interpret the wide range of instructor responses to GenAI usage as evidence of the perceived instructor need to intervene even without clear guidelines from policy or research on effective student-AI interactions. Interviewees found navigating this uncertainty challenging. As one instructor said, "we have to figure it out on our own" (P08).

Survey results suggest that our interviewees are not enthusiastic outliers about teaching GenAI usage in their coursework: 80% rated integrating GenAI into computing curriculum as Important or Very Important (Figure 3a). However, fewer survey respondents had already begun this process, as only 37% used GenAI tools during course activities Often or Very Often (Figure 3b).

Key Insights: Lacking direct control over commercial tools and operating under sparse guidance, instructors needed to work through policies, task design, workflow requirements, and custom wrappers to shape *how* students used GenAI rather than *whether* they used it.

4.5 Assessment Misfit

We use "assessment" in two senses that instructors treated as related but distinct: (1) behavioral alignment: whether students used GenAI in the ways instructors intended; and (2) learning without scaffolds: whether students could explain, trace, and write code unaided. In most reported cases, interviewees used proctored assessments as a way to check whether students were interacting with GenAI tools in intended ways. However, understanding student behavior prior to a summative assessment was challenging because instructors only had partial visibility into day-to-day tool use.

Challenges of understanding student-AI interactions. Instructors observed an assignment–exam gap: students performed well on auto-graded or take-home work yet struggled on proctored tasks (P04, P06, P08, P10, P12, P13). This motivated them to examine student-AI interactions more closely and attempt to make changes. P04 reported a memorable example during the quarter where he first introduced his custom chatbot. On a mid-quarter skill demo requiring students to write a short Python function from scratch, "One third of my students got a 0. This was a class of 450 students, so 150 students got a 0, which was very concerning to me." P04 had originally planned to introduce students to GitHub Copilot and allow them to use Copilot for the remainder of the course, but decided to completely backtrack and not introduce Copilot at all because he was concerned about student knowledge of programming basics.

P08 described a related pattern: a confidence–competence gap. Students who had succeeded on take-home work "were very insistent that they understood the material, yet when I asked them followup questions about their code, they could not respond." P08 suspected these students were using copy-paste workflows and limited practice with writing or tracing code from scratch but could not verify these behaviors across the full class. P08's example illustrates a shared challenge with assessing student-AI interactions. Because student-AI interactions occurred in commercial tools outside course infrastructure by default, instructors lacked reliable telemetry: they could not see which tools were used, how prompting evolved, or whether course guidelines were bypassed. As a result, instructors inferred effects from ambient signals – office hours, staff anecdotes, autograder patterns, and proctored performance – rather than from planned, instrumented evaluations.

Even in cases where this telemetry was possible in theory, understanding student behavior in practice was difficult. Since students used his custom chatbot, P04 had access to logs of student interactions. However, he noted that analyzing hundreds of conversation logs was impractical during the term since it would take too much time away from routine teaching work. P07 shared an experience building a similar custom chatbot, only to find that students found the chatbot less helpful than commercial tools and chose to use commercial tools instead.

Challenges of assessing student knowledge. Other instructors spent considerable effort redesigning their course's assignments and grading criteria. For example, P02 shifted most assignment credit to brief, weekly "stand-up" meetings where students demonstrated and discussed their own code. Submitted code counted for only 15–20% of the grade; these weekly meetings carried the bulk of credit instead. Stand-up meetings targeted three behaviors: communicating the program; explaining how data flows through the program; and predicting outcomes under small changes. P06 adopted similar standup meetings, noting that these meetings quickly surfaced students who were struggling and deterred integrity problems. A shared challenge was ensuring that grading was consistent across all students. In this case, both P02 and P06 felt the need to create detailed, step-by-step instructions for their course staff on how to grade every assignment that used stand-up meetings.

In an algorithms course, P01 awarded full credit for "good-faith effort" on weekly problem sets and scaled the amount of feedback with the amount of visible work. This grading emphasized engagement and positioned feedback as the main incentive, while summative written exams still assessed individual reasoning.

P13 reoriented some assignments so that a small portion involved obtaining GenAI output, with most credit tied to students' critiques of that output (e.g., identifying inconsistencies and justifying revisions), shifting assessed value from production to evaluation.

Tensions and open questions. These redesigns raised four unresolved issues. First, *validity*: do orals, critiques, and explanation items measure the target constructs (e.g., program comprehension)? Second, *reliability*: instructors noted questions about TA calibration for orals and alignment between AI-graded and human-graded judgments but did not report formal checks. Third, *feasibility*: weekly standup meetings and detailed assignment feedback scaled only with substantial staffing effort. Fourth, *equity*: when students differ in access to paid tools or in comfort with spoken English, assessment changes may create uneven burdens.

Finally, several instructors questioned whether current practices provide valid evidence of learning. P10, who manages program-level assessment in her department, noted that colleagues' claims about GenAI "working" rested on perceptions and student enthusiasm rather than on validated measures, and emphasized the need to assess learning outcomes at the program level. Across interviews, instructors acted under time pressure and partial observability; they relied on ambient evidence rather than orchestrated evaluations when judging whether students both used GenAI as intended and learned the target skills.

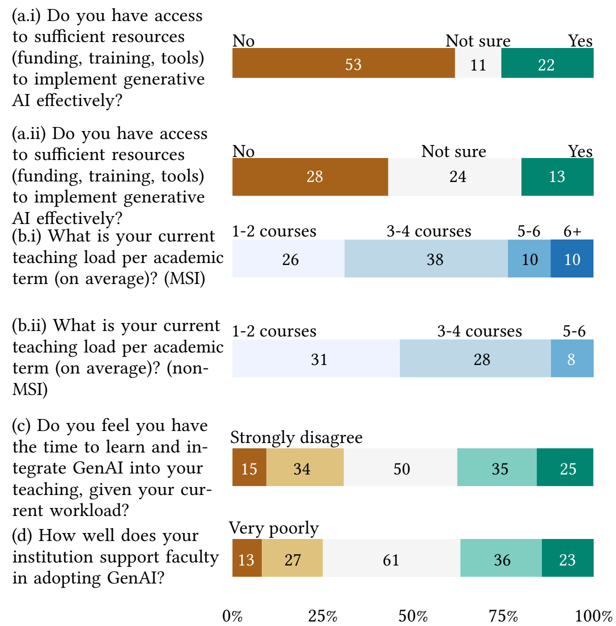

Figure 4: A substantial number of our survey respondents expressed that they did not have the resources they needed to implement GenAI, which was especially prominent for MSI instructors (a.i, a.ii). MSI instructors were also more likely to have heavier teaching loads (b.i, b.ii). Instructors generally perceived a lack of sufficient time to learn and integrate GenAI (c), and that their institutions did not support this work (d). Instructors responded on a 5-point scale. The number of responses for each option is displayed within each bar.

Key Insights: Instructors operated with partial observability, inferring behavior from informal, ambient signals. Credit on course assignments tended to shift from correctness toward explanation, communication, and judgment. These moves raise open questions about validity, reliability, feasibility, and equity, and point to a need for ways to capture both how students use GenAI and what they can do without it.

4.6 Lack of Resources

Adapting courses for GenAI required resources that many surveyed computing instructors lacked. When asked whether they had sufficient resources like funding, training, and tools to implement GenAI effectively, 53% responded No. This pattern was more pronounced among MSI instructors: 62% responded No, compared to 43% of non-MSI instructors (Figure 4a.i, a.ii). Teaching load may help explain this difference. MSI instructors were more likely to teach three or more courses per term (70% vs. 54% for non-MSIs) (Figure 4b.i, b.ii). In the most extreme cases, 10 respondents reported teaching six or more courses per academic term; all were from MSIs.

Instructors also expressed a lack of time: when asked if they had time to learn and integrate GenAI given their current workload, 62% of surveyed instructors responded Strongly Disagree, Disagree, or Neither (Figure 4c). When asked how well their institution supports faculty in adopting emerging technologies like GenAI, a nearly identical 63% of faculty responded Very Poorly, Poorly, or Neither (Figure 4d).

To understand the kinds of resources actually required to adapt a course, we examined the resources that our interviewees had when conducting emergency pedagogical design. We found that our interviewees dedicated substantial amounts of time, staff hours, and sometimes even their own money in the process of emergency pedagogical design. For example, bespoke infrastructure required ongoing development and maintenance. As mentioned in Section 4.4, P04 deployed a custom chatbot and P09 built an AI-graded explanation workflow; both of these systems required design, implementation, bug fixes, and upkeep during the term. These two instructors reported that these systems were only possible with financial support to hire software engineers and PhD students.

Whole-course redesign also demanded time. Only two interviewees decided to revise a majority of their course materials (P05, P08). To accomplish this, both interviewees adopted course materials that were built by others rather than developing them from scratch, noting that it would be infeasible to revise an entire course's worth of curriculum on their own.

Alternative assessments often meant a shift away from autograded work towards qualitative work, which also required more course staff to manage. As mentioned in Section 4.5, P02 reorganized his course around weekly stand-up meetings with every student. To handle this for his course of 300 students, he needed "a lot of TAs!" (P02). In practice, this course needed to recruit and train over 50 course staff members to cover approximately 300 students, a 6:1 staff to student ratio. Others emphasized the hand-grading burden for creative work, especially in courses that traditionally relied on autograder tests (P05, P08, P13).

Direct costs varied. Some instructors reported that their institutions had agreements with GenAI companies to provide the latest models at no cost to their students (P02, P13). Others spent their own money out-of-pocket in order to support student-AI usage for custom AI tools (P06, P12). Instructors also pointed to external funds to offset API or engineering costs (e.g., P04, P08, P09).

As counterexamples to this pattern, some instructors reached their goals with off-the-shelf tools and minimal course changes. For example, P01 primarily guided students to use free off-the-shelf tools to generate example inputs for algorithms (P01). Others taught explicit prompting practices without building bespoke systems (e.g., P03, P07).

What is notable about our interviewees is that they taught at most two courses per academic term, and many had advantages such as partnerships, external funds, curriculum materials, or the ability to hire many course staff members. This pattern is striking: substantial attempts to change student-AI interaction were made by instructors with lighter teaching loads and access to staffing or funding. These accounts highlight the kinds of resources – time, staffing, funding, and infrastructure – that enabled our interviewees to make substantial course changes. Yet our survey data suggests that most computing instructors, especially those at MSIs who reported heavier teaching loads and less institutional support, do not have comparable resources. Because we view our survey sample as a more representative of the constraints faced by computing faculty, this gap indicates that many computing instructors may be unable to adopt the kinds of course adaptations described by our interviewees without additional support.

Key Insights: Instructors who successfully adapted courses for GenAI tended to have advantages like lower teaching load, staffing, and funding. Instructors who lacked these resources, especially those at MSIs, found it more difficult to effectively engage in emergency pedagogical design.

5 Discussion

Our findings position emergency pedagogical design as a distinct design setting for HCI. Emergency pedagogical design arises when instructors must shape student-AI interactions under four conditions: reactive timing, indirect control over commercial tools, partial observability of day-to-day behavior, and time pressure to act. Without the time needed to conduct large-scale evaluations, instructors chose to rely on ambient evidence gathered from informal student observations. We expect that emergency pedagogical design will become less necessary as GenAI tools stabilize, policies align across courses, and validated measures of student-AI behavioral alignment are established. In the meantime, however, there is a rare opportunity for HCI research, academic institutions, and funding agencies to impact how instructors approach student-AI interactions in the years to come.

5.1 Open Questions for HCI Research

As outlined by the five barriers of our research findings (Section 4), building effective courses is a challenging design problem made even more complicated as instructors are now trying to influence student-AI interactions. Some barriers, especially Implementation Challenges (Section 4.4) and Assessment Misfit (Section 4.5), suggest open questions for future HCI research to support instructors in emergency pedagogical design which we pose here for the research community.

- Defining traits of effective student-AI interactions: Our interviewees expressed a desire to improve how students interact with GenAI tools, and often had clear ideas about undesirable behaviors (such as copy-pasting code), but struggled to articulate what constitutes more effective or desirable student-AI interaction. How can we define a taxonomy of student-AI interactions? How can HCI research help instructors translate learning outcomes into specific, desirable, and undesirable types of student-AI interactions?
- Understanding effects of Policy Crosswinds (Section 4.3): Students frequently encounter divergent policies regarding GenAI usage across different courses. How do students respond to inconsistent policies, such as one course requiring AI suggestions to be disabled while another encourages their use? What factors influence whether students adhere to these varying policies?
- Designing desirable course-aligned tools: Since students can easily switch to general-purpose chatbots, what affordances

and incentives make course-specific or course-aligned tools the preferred option for students without relying on coercion?
- Understanding student-AI behaviors with commercial tools: Instructors typically have only partial visibility into how students use commercial GenAI tools, relying on proxies to infer behavior, such as scores on a proctored quiz. How reliable are current instructor proxies for understanding student-AI usage? Can we design more accurate proxies that support early detection of undesirable interactions while also preserving student privacy? How can these proxies inform instructor decision-making?
- Understanding student-AI behaviors with bespoke tools: Instructors who develop custom wrappers around GenAI tools theoretically have greater insight into student interactions, but often find it difficult to analyze large volumes of interaction data. How can tools help instructors make sense of student-AI interactions that are directly observable (e.g., through concept induction [37])? How can practices from machine learning monitoring be adapted to educational contexts without requiring instructors to manage extensive infrastructure or acquire new technical expertise?
- Supporting adoptability for instructors: Given that instructors face significant time, staffing, and financial constraints, how can we design tools and interventions that are easy to adopt, require minimal disruption to existing course materials, and remain low-cost for instructors?
- Making use of ambient evidence: Instructors often rely on informal signals such as office hour conversations and staff reports to gain insight into student behavior. How can tools help capture a greater quantity and quality of ambient data, and support instructors in acting on this information? How closely can we approximate the benefits of detailed user studies using interactions that already occur, such as office hours and online forum posts?
- Preserving student privacy: Many approaches to understanding student-AI interactions require collecting additional data about student behaviors, which risks capturing sensitive information. What types of aggregates and visualizations can provide instructors with useful visibility (such as usage summaries, failure modes, and guardrail breaches) without exposing sensitive content? What consent and governance models are appropriate for educational settings?

5.2 Recommendations for Academic Institutions and Funders

When we began recruiting interviewees for this study, we initially expected that it would be straightforward to find computing instructors who had updated their course materials in response to GenAI, given that more than two years had passed since the release of prominent tools like ChatGPT and GitHub Copilot. However, we quickly discovered that our assumption was misplaced: it was far easier to find instructors who had only revised their course policies, rather than those who had made substantive updates to assignments, assessments, or instructional content. One possibility is that while the conversation around GenAI in education is

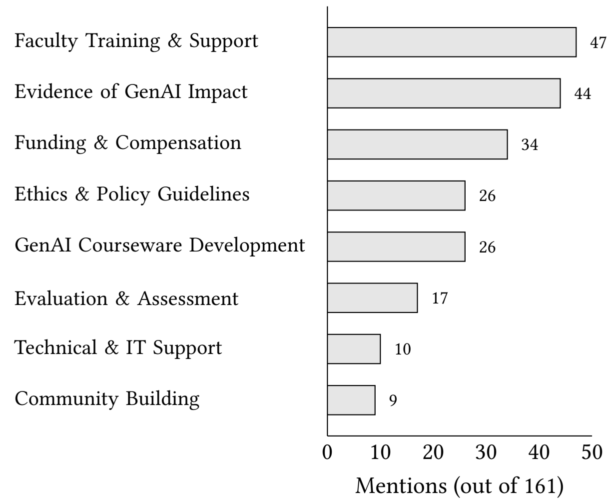

Figure 5: Responses to the question "In your opinion, what kinds of support would be most helpful to address faculty resistance to teaching GenAI?" Survey respondents were allowed to choose up to three options from a pre-defined list. Eight most common responses shown, with counts displayed to the right of each bar.

widespread, the actual integration of GenAI into course materials remains relatively rare.

Our interview findings further revealed that the work of emergency pedagogical design is both time-consuming and largely unrecognized by institutions. This burden is exacerbated by the absence of established pedagogical guidelines and best practices for GenAI instruction, requiring instructors to craft custom solutions often from scratch. For example, none of our interviewees reported direct support such as teaching relief, additional course staff, or extra compensation for the substantial work they performed to redesign materials or assessments. This lack of institutional recognition and support meant that instructors shouldered the burden of adaptation on top of their routine teaching responsibilities, often working in isolation and without formal acknowledgment. Survey results extended these findings: instructors expressed a desire to improve student-AI interactions by updating their course materials, yet reported that they were prevented from doing so by resource constraints, a challenge especially pronounced at MSIs with heavier teaching loads and more limited access to staff or funding (Section 4.6). These patterns raise equity concerns, as there is a real risk that well-resourced institutions and instructors will be able to adapt quickly and effectively to the challenges and opportunities of GenAI, while less-resourced institutions, particularly those serving underrepresented student groups, may fall further behind. If only the most privileged institutions can afford to update their curricula and assessments, GenAI could reinforce existing disparities in student outcomes and preparedness.

Not all of the issues surfaced by our study can be addressed solely through HCI research or technical interventions; barriers like Fragmented Buy-in (Section 4.2), Policy Crosswinds (Section 4.3), and Lack of Resources (Section 4.6) are also structural or institutional in nature. In light of these findings, we offer several recommendations for academic institutions and funders to more effectively support instructors as they design better student-AI interactions. Academic institutions and departments should acknowledge that emergency pedagogical design represents additional work that instructors are taking on while still needing to manage their routine teaching responsibilities. We suggest reframing policy changes such as blanket bans on student GenAI usage as a form of emergency pedagogical design: instructors who impose such bans are still trying to reduce undesired student-AI interactions via course policy and face similar challenges in assessing whether these policies have actually succeeded in influencing student behavior. In this reframing, nearly all instructors are engaged in emergency pedagogical design of some kind, and we recommend that academic institutions reward instructors who wish to make improvements for faculty across their departments and schools. Departments should make establishing shared policies a priority to reduce the potential harms of policy crosswinds, especially on students. Institutions should provide resources for instructors who are leaders within their institutions and communities, potentially in the form of teaching reliefs, the ability to hire additional course staff, and stipends for hosting training workshops for other faculty. Where less resources are available, institutions can recognize even individual, small-scale interventions and facilitate faculty share-outs to mitigate fragmented buy-in.

Another notable observation from our interviews was that few interviewees were aware of how other instructors were making changes to their curriculum and assessing the effectiveness of these changes outside of their home departments. To address this, we recommend funding organizations to support efforts to create communities of practice where instructors can share preliminary work, curriculum and approaches. This could fulfill a pressing need: when survey respondents were asked what kinds of support would be most helpful to address faculty resistance to teaching GenAI, faculty training and support was the most requested resource, followed closely by evidence of GenAI impact (Figure 5). Although academic research venues remain an important way to share peer-reviewed experiences, the length of the publication cycle is long, and instructors can benefit even from informal discussions about work-in-progress approaches. Since HCI research prioritizes novelty and academic departments prioritize local impact, we encourage funders to provide incentives for HCI researchers to produce tools and techniques that can be easily adopted at scale, and for instructors to broadcast and receive credit for innovative approaches to emergency pedagogical design that can be more agile than publishing peer-reviewed papers. All of these recommendations and suggestions reflect a continued need for administrators, department chairs, and funders to support faculty as they engage in emergency pedagogical design.

5.3 Learning from Emergency Remote Teaching

Less than a decade ago, instructors also had to make sudden sweeping changes to their teaching in response to COVID-19. What are some lessons learned from emergency remote teaching that might help address the challenges that computing instructors now face when conducting emergency pedagogical design? In our view, one of the more helpful perspectives taken during COVID-19 was drawing a clear distinction between online learning, where a course is specifically designed for an online modality, and emergency remote teaching, where a course is offered online "in a hurry, with bare minimum resources, and scant time" [29]. In this paper, we argue for a similar distinction. While decades of research have examined how to integrate AI into pedagogy (for example, via intelligent tutoring systems [5]), this research has also shown that effectively using AI often takes forethought and careful design. As was the case for emergency remote teaching, we view the lack of resources and time that instructors now face in emergency pedagogical design (Section 4.6) as an indicator that emergency pedagogical design should be treated as distinct from designing AI-integrated courses.

Another outcome of emergency remote teaching was that instructors became more familiar with techniques for online learning and blended learning [64] and found them valuable even after a return to in-person instruction [2]. If this pattern also holds for emergency pedagogical design, we might expect that Fragmented Buy-in (Section 4.2) and Policy Crosswinds (Section 4.3) will become less influential as more instructors will learn about and use GenAI tools for instruction. However, this will be made more possible as we solve current problems in implementation (Section 4.4) and assessment (Section 4.5) of student-AI interactions, and thus we believe that HCI research has a critical role to play in discovering effective design practices for the future.

6 Conclusion

We studied how computing instructors adapted courses to widespread GenAI use through interviews (n=13) and a survey (n=169), framing this work as *emergency pedagogical design*: reactive, indirect efforts to shape student-AI interactions under time pressure and partial observability. We characterized the properties of emergency pedagogical design and five barriers that instructors face in this work: fragmented buy-in, policy crosswinds, implementation challenges, assessment misfit, and lack of resources. We positioned emergency pedagogical design as a design setting for HCI where iteration relies on ambient evidence. Looking ahead, we encourage HCI researchers, tool builders, institutions, and funding agencies to help shift from emergency responses to routine practice by providing instructors with new knowledge and tools to improve student-AI interactions, more robust evaluation methods, time and recognition for instructor efforts, and financial support for access and maintenance.

Acknowledgments

We thank Google for their generous support for this project. Large language models were used to revise the text for clarity, spelling, and grammar.

References

[1] William C. Adams. 2015. Conducting Semi-Structured Interviews. In *Handbook of Practical Program Evaluation*. (1st ed.). Kathryn E. Newcomer, Harry P. Hatry, and Joseph S. Wholey, (Eds.) Wiley, (Aug. 10, 2015), 492–505. ISBN: 978-1-118-89360-9 978-1-119-17138-6. doi:10.1002/9781119171386.ch19.

[2] Olasile Babatunde Adedoyin and Emrah Soykan. 2023. Covid-19 pandemic and online learning: the challenges and opportunities. *Interactive Learning Environments*, 31, 2, (Feb. 17, 2023), 863–875. doi:10.1080/10494820.2020.1813180.

[3] Rudaiba Adnin, Atharva Pandkar, Bingsheng Yao, Dakuo Wang, and Maitraye Das. 2025. Examining Student and Teacher Perspectives on Undisclosed Use of Generative AI in Academic Work. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (CHI '25). Association for Computing Machinery, New York, NY, USA, (Apr. 25, 2025), 1–17. isbn: 979-8-4007-1394-1. doi:10.1145/3706598.3713393.

[4] Walter R. Allen, Joseph O. Jewell, Kimberly A. Griffin, and De'Sha S. Wolf. 2007. Historically Black colleges and universities: Honoring the past, engaging the present, touching the future. *The Journal of Negro Education*, 263–280. Retrieved Sept. 8, 2025 from https://www.jstor.org/stable/40034570?casa_token=M5oD_NMQm14AAAAA:PJJ_rV9zx5AIuAVx_kfLfaP5xFBbcUfwPpbuKnl02ngTb-y2gc6DDsVeHVqGMwBFFM_y0tx8LvJF81ht3Ts6HQPMwcqV55obsWrBFYlEbo7iVivpOH-D JSTOR: 40034570.

[5] John R. Anderson, Albert T. Corbett, Kenneth R. Koedinger, and Ray. Pelletier. 1995. Cognitive Tutors: Lessons Learned. *Journal of the Learning Sciences*, 4, 2, (Apr. 1995), 167–207. doi:10.1207/s15327809jls0402_2.

[6] David Baidoo-Anu and Leticia Owusu Ansah. 2023. Education in the Era of Generative Artificial Intelligence (AI): Understanding the Potential Benefits of ChatGPT in Promoting Teaching and Learning. *Journal of AI*, 7, 1, (Dec. 31, 2023), 52–62. doi:10.61969/jai.1337500.

[7] Hamsa Bastani, Osbert Bastani, Alp Sungu, Haosen Ge, Ozge Kabakcı, and Rei Mariman. 2024. Generative AI can harm learning. *Available at SSRN*, 4895486. Retrieved Sept. 11, 2025 from https://atelierdesfuturs.org/wp-content/uploads/2025/07/ssrn-4895486.pdf.

[8] Rodrigo Borela, Meryem Yilmaz Soylu, Jeonghyun Lee, and Nimisha Roy. 2025. What Computing Faculty Want: Designing AI Tools for High-Enrollment Courses Beyond CS1. In *Proceedings of the 2025 ACM Conference on International Computing Education Research V.2* (ICER '25). Association for Computing Machinery, New York, NY, USA, (Aug. 3, 2025), 32–33. isbn: 979-8-4007-1341-5. doi:10.1145/3702653.3744327.

[9] Valerie Chen, Alan Zhu, Sebastian Zhao, Hussein Mozannar, David Sontag, and Ameet Talwalkar. 2025. Need Help? Designing Proactive AI Assistants for Programming. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (CHI '25). Association for Computing Machinery, New York, NY, USA, (Apr. 25, 2025), 1–18. isbn: 979-8-4007-1394-1. doi:10.1145/3706598.3714002.

[10] Seongyune Choi, Yeonju Jang, and Hyeoncheol Kim. 2023. Influence of Pedagogical Beliefs and Perceived Trust on Teachers' Acceptance of Educational Artificial Intelligence Tools. *International Journal of Human–Computer Interaction*, 39, 4, (Feb. 25, 2023), 910–922. doi:10.1080/10447318.2022.2049145.

[11] Rudrajit Choudhuri, Dylan Liu, Igor Steinmacher, Marco Gerosa, and Anita Sarma. 2024. How Far Are We? The Triumphs and Trials of Generative AI in Learning Software Engineering. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (ICSE '24). Association for Computing Machinery, New York, NY, USA, (Apr. 12, 2024), 1–13. isbn: 979-8-4007-0217-4. doi:10.1145/3597503.3639201.

[12] Juliet Corbin and Anselm Strauss. 2014. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage publications. Retrieved Sept. 9, 2025 from https://books.google.com/books?hl=en&lr=&id=MaKWBQAAQBAJ&oi=fnd&pg=PP1&dq=Basics+of+qualitative+research&ots=QuFfxO86R2&sig=O0OXLODEDuvoXghgizYbM93XEvs.

[13] Digital Education Council. 2025. What Faculty Want: Key Results from the Global AI Faculty Survey 2025. (Jan. 28, 2025). Retrieved Aug. 18, 2025 from https://www.digitaleducationcouncil.com/post/what-faculty-want-key-results-from-the-global-ai-faculty-survey-2025.

[14] Mark Allen Cu and Sebastian Hochman. 2023. Scores of Stanford students used ChatGPT on final exams. (Jan. 22, 2023). Retrieved Sept. 2, 2025 from https://stanforddaily.com/2023/01/22/scores-of-stanford-students-used-chatgpt-on-final-exams-survey-suggests/.

[15] Paul Denny, Juho Leinonen, James Prather, Andrew Luxton-Reilly, Thezyrie Amarouche, Brett A. Becker, and Brent N. Reeves. 2024. Prompt Problems: A New Programming Exercise for the Generative AI Era. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSE 2024: The 55th ACM Technical Symposium on Computer Science Education. ACM, Portland OR USA, (Mar. 7, 2024), 296–302. isbn: 979-8-4007-0423-9. doi:10.1145/3626252.3630909.

[16] Paul Denny et al. 2024. Computing Education in the Era of Generative AI. *Communications of the ACM*, 67, 2, (Feb. 2024), 56–67. doi:10.1145/3624720.

[17] Areej ElSayary. 2024. An investigation of teachers' perceptions of using ChatGPT as a supporting tool for teaching and learning in the digital era. *Journal of Computer Assisted Learning*, 40, 3, (June 2024), 931–945. doi:10.1111/jcal.12926.

[18] Kasra Ferdowsi, Ruanqianqian (Lisa) Huang, Michael B. James, Nadia Polikarpova, and Sorin Lerner. 2024. Validating AI-Generated Code with Live Programming. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (CHI '24). Association for Computing Machinery, New York, NY, USA, (May 11, 2024), 1–8. isbn: 979-8-4007-0330-0. doi:10.1145/3613904.3642495.

[19] James Finnie-Ansley, Paul Denny, Brett A. Becker, Andrew Luxton-Reilly, and James Prather. 2022. The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming. In *Proceedings of the 24th Australasian Computing Education Conference* (ACE '22). Association for Computing Machinery, New York, NY, USA, (Feb. 14, 2022), 10–19. isbn: 978-1-4503-9643-1. doi:10.1145/3511861.3511863.

[20] James Finnie-Ansley, Paul Denny, Andrew Luxton-Reilly, Eddie Antonio Santos, James Prather, and Brett A. Becker. 2023. My AI Wants to Know if This Will Be on the Exam: Testing OpenAI's Codex on CS2 Programming Exercises. In *Proceedings of the 25th Australasian Computing Education Conference*. ACE '23: Australasian Computing Education Conference. ACM, Melbourne VIC Australia, (Jan. 30, 2023), 97–104. isbn: 978-1-4503-9941-8. doi:10.1145/3576123.3576134.

[21] John C. Flanagan. 1954. The critical incident technique. *Psychological bulletin*, 51, 4, 327. Retrieved Sept. 8, 2025 from https://psycnet.apa.org/record/1955-01751-001.

[22] Behzad Foroughi, Madugoda Gunaratnege Senali, Mohammad Iranmanesh, Ahmad Khanfar, Morteza Ghobakhloo, Nagaletchimee Annamalai, and Bita Naghmeh-Abbaspour. 2024. Determinants of Intention to Use ChatGPT for Educational Purposes: Findings from PLS-SEM and fsQCA. *International Journal of Human–Computer Interaction*, 40, 17, (Sept. 1, 2024), 4501–4520. doi:10.1080/10447318.2023.2226495.

[23] Diana Franklin, Paul Denny, David A. Gonzalez-Maldonado, and Minh Tran. 2025. Generative AI in Computer Science Education: Challenges and Opportunities. *Elements in Generative AI in Education*, (Apr. 2025). isbn: 9781009581738 9781009581691 9781009581707. doi:10.1017/9781009581738.

[24] Nat Friedman. 2021. Introducing GitHub Copilot: your AI pair programmer. The GitHub Blog. (June 29, 2021). Retrieved May 14, 2025 from https://github.blog/news-insights/product-news/introducing-github-copilot-ai-pair-programmer/.

[25] Sharon Goldman. 2023. OpenAI rival Anthropic introduces Claude, an AI assistant to take on ChatGPT. VentureBeat. (Mar. 14, 2023). Retrieved May 2, 2025 from https://venturebeat.com/ai/google-funded-anthropic-introduces-claude-chatgpt-rival-through-chat-and-api/.

[26] Jamie Gorson Benario, Jenn Marroquin, Monica M. Chan, Ernest D.V. Holmes, and Daniel Mejia. 2025. Unlocking Potential with Generative AI Instruction: Investigating Mid-level Software Development Student Perceptions, Behavior, and Adoption. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSE TS 2025: The 56th ACM Technical Symposium on Computer Science Education. ACM, Pittsburgh PA USA, (Feb. 12, 2025), 395–401. isbn: 979-8-4007-0531-1. doi:10.1145/3641554.3701859.

[27] Summit Haque and Christopher Hundhausen. 2025. Generative AI Access, Usage, and Perceptions: An Empirical Comparison of Computing Students In The United States and Bangladesh. In *Proceedings of the 2025 ACM Conference on International Computing Education Research V.1*. ICER 2025: ACM Conference on International Computing Education Research. ACM, Charlottesville USA, (Aug. 3, 2025), 109–124. isbn: 979-8-4007-1340-8. doi:10.1145/3702652.3744231.

[28] Emma Harvey, Allison Koenecke, and Rene F. Kizilcec. 2025. "Don't Forget the Teachers": Towards an Educator-Centered Understanding of Harms from Large Language Models in Education. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (CHI '25). Association for Computing Machinery, New York, NY, USA, (Apr. 25, 2025), 1–19. isbn: 979-8-4007-1394-1. doi:10.1145/3706598.3713210.

[29] Charles Hodges, Stephanie Moore, Barb Lockee, Torrey Trust, and Aaron Bond. 2020. The difference between emergency remote teaching and online learning. *Educause review*, 27, 1, 1–9. Retrieved Sept. 2, 2025 from https://brill.com/downloadpdf/display/book/9789004702813/BP000029.pdf.

[30] Burke Holland. 2024. Announcing a free GitHub Copilot for VS Code. (Dec. 18, 2024). Retrieved Sept. 2, 2025 from https://code.visualstudio.com/blogs/2024/12/18/free-github-copilot.

[31] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. SWE-bench: Can Language Models Resolve Real-World GitHub Issues? (Nov. 11, 2024). arXiv: 2310.06770 [cs]. Retrieved June 21, 2025 from http://arxiv.org/abs/2310.06770. Pre-published.

[32] Arianna Johnson. 2023. ChatGPT In Schools: Here's Where It's Banned—And How It Could Potentially Help Students. Forbes. (Jan. 18, 2023). Retrieved Sept. 11, 2025 from https://www.forbes.com/sites/ariannajohnson/2023/01/18/chatgpt-in-schools-heres-where-its-banned-and-how-it-could-potentially-help-students/.

[33] Majeed Kazemitabaar, Justin Chow, Carl Ka To Ma, Barbara J. Ericson, David Weintrop, and Tovi Grossman. 2023. Studying the effect of AI Code Generators on Supporting Novice Learners in Introductory Programming. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI '23: CHI Conference on Human Factors in Computing Systems. ACM, Hamburg Germany, (Apr. 19, 2023), 1–23. isbn: 978-1-4503-9421-5. doi:10.1145/3544548.3580919.

[34] Majeed Kazemitabaar, Jack Williams, Ian Drosos, Tovi Grossman, Austin Zachary Henley, Carina Negreanu, and Advait Sarkar. 2024. Improving Steering and Verification in AI-Assisted Data Analysis with Interactive Task Decomposition. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. UIST '24: The 37th Annual ACM Symposium on User Interface Software and Technology. ACM, Pittsburgh PA USA, (Oct. 13, 2024), 1–19. ISBN: 979-8-4007-0628-8. doi:10.1145/3654777.3676345.

[35] Majeed Kazemitabaar, Runlong Ye, Xiaoning Wang, Austin Zachary Henley, Paul Denny, Michelle Craig, and Tovi Grossman. 2024. CodeAid: Evaluating a Classroom Deployment of an LLM-based Programming Assistant that Balances Student and Educator Needs. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. CHI '24: CHI Conference on Human Factors in Computing Systems. ACM, Honolulu HI USA, (May 11, 2024), 1–20. ISBN: 979-8-4007-0330-0. doi:10.1145/3613904.3642773.

[36] Anjali Khurana, Xiaotian Su, April Yi Wang, and Parmit K Chilana. 2025. Do It For Me vs. Do It With Me: Investigating User Perceptions of Different Paradigms of Automation in Copilots for Feature-Rich Software. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (CHI '25). Association for Computing Machinery, New York, NY, USA, (Apr. 25, 2025), 1–18. ISBN: 979-8-4007-1394-1. doi:10.1145/3706598.3713431.

[37] Michelle S. Lam, Janice Teoh, James A. Landay, Jeffrey Heer, and Michael S. Bernstein. 2024. Concept Induction: Analyzing Unstructured Text with High-Level Concepts Using LLooM. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. CHI '24: CHI Conference on Human Factors in Computing Systems. ACM, Honolulu HI USA, (May 11, 2024), 1–28. ISBN: 979-8-4007-0330-0. doi:10.1145/3613904.3642830.

[38] Sam Lau and Philip Guo. 2023. From "Ban It Till We Understand It" to "Resistance is Futile": How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot. In *Proceedings of the 2023 ACM Conference on International Computing Education Research V.1*. ICER 2023: ACM Conference on International Computing Education Research. ACM, Chicago IL USA, (Aug. 7, 2023), 106–121. ISBN: 978-1-4503-9976-0. doi:10.1145/3568813.3600138.

[39] Xiaojie Li and C. Dennis Carroll. 2007. Characteristics of minority-serving institutions and minority undergraduates enrolled in these institutions: Postsecondary education descriptive analysis report (nces 2008-156). *National Center for Education Statistics*. Retrieved Sept. 8, 2025 from https://eric.ed.gov/?id=ED499114.

[40] Yujia Li et al. 2022. Competition-level code generation with AlphaCode. *Science*, 378, 6624, (Dec. 9, 2022), 1092–1097. doi:10.1126/science.abq1158.

[41] Mark Liffiton, Brad E Sheese, Jaromir Savelka, and Paul Denny. 2023. CodeHelp: Using Large Language Models with Guardrails for Scalable Support in Programming Classes. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*. Koli Calling '23: 23rd Koli Calling International Conference on Computing Education Research. ACM, Koli Finland, (Nov. 13, 2023), 1–11. ISBN: 979-8-4007-1653-9. doi:10.1145/3631802.3631830.

[42] Francesca Lucchetti, Zixuan Wu, Arjun Guha, Molly Q. Feldman, and Carolyn Jane Anderson. 2024. Substance Beats Style: Why Beginning Students Fail to Code with LLMs. (Oct. 15, 2024). arXiv: 2410.19792 [cs]. Retrieved Aug. 19, 2025 from http://arxiv.org/abs/2410.19792. Pre-published.

[43] Wenhan Lyu, Shuang Zhang, Tingting Chung, Yifan Sun, and Yixuan Zhang. 2025. Understanding the practices, perceptions, and (dis)trust of generative AI among instructors: A mixed-methods study in the U.S. higher education. *Computers and Education: Artificial Intelligence*, 8, (June 1, 2025), 100383. doi:10.1016/j.caeai.2025.100383.

[44] Shuai Ma, Junling Wang, Yuanhao Zhang, Xiaojuan Ma, and April Yi Wang. 2025. DBox: Scaffolding Algorithmic Programming Learning through Learner-LLM Co-Decomposition. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (CHI '25). Association for Computing Machinery, New York, NY, USA, (Apr. 25, 2025), 1–20. ISBN: 979-8-4007-1394-1. doi:10.1145/3706598.3713748.

[45] Thomas Mahatody, Mouldi Sagar, and Christophe Kolski. 2010. State of the Art on the Cognitive Walkthrough Method, Its Variants and Evolutions. *International Journal of Human-Computer Interaction*, 26, 8, (July 30, 2010), 741–785. doi:10.1080/10447311003781409.

[46] Nora McDonald, Aditya Johri, Areej Ali, and Aayushi Hingle Collier. 2025. Generative artificial intelligence in higher education: Evidence from an analysis of institutional policies and guidelines. *Computers in Human Behavior: Artificial Humans*, 3, 100121. Retrieved Dec. 4, 2025 from https://www.sciencedirect.com/science/article/pii/S2949882125000052.

[47] Ivan Mehta. 2025. A quarter of startups in YC's current cohort have codebases that are almost entirely AI-generated. TechCrunch. (Mar. 6, 2025). Retrieved Sept. 9, 2025 from https://techcrunch.com/2025/03/06/a-quarter-of-startups-in-ycs-current-cohort-have-codebases-that-are-almost-entirely-ai-generated/.

[48] OpenAI. 2022. Introducing ChatGPT. (Nov. 30, 2022). Retrieved May 2, 2025 from https://openai.com/index/chatgpt/.

[49] Lawrence A. Palinkas, Sarah M. Horwitz, Carla A. Green, Jennifer P. Wisdom, Naihua Duan, and Kimberly Hoagwood. 2015. Purposeful Sampling for Qualitative Data Collection and Analysis in Mixed Method Implementation Research. *Administration and Policy in Mental Health and Mental Health Services Research*, 42, 5, (Sept. 2015), 533–544. doi:10.1007/s10488-013-0528-y.

[50] Hyanghee Park and Daehwan Ahn. 2024. The Promise and Peril of ChatGPT in Higher Education: Opportunities, Challenges, and Design Implications. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (CHI '24). Association for Computing Machinery, New York, NY, USA, (May 11, 2024), 1–21. ISBN: 979-8-4007-0330-0. doi:10.1145/3613904.3642785.

[51] Sundar Pichai and Demis Hassabis. 2023. Introducing Gemini: our largest and most capable AI model. Google. (Dec. 6, 2023). Retrieved Sept. 8, 2025 from https://blog.google/technology/ai/google-gemini-ai/.

[52] James Prather, Brent Reeves, Juho Leinonen, Stephen MacNeil, Arisoa S. Randrianasolo, Brett Becker, Bailey Kimmel, Jared Wright, and Ben Briggs. 2024. The Widening Gap: The Benefits and Harms of Generative AI for Novice Programmers. (May 27, 2024). arXiv: 2405.17739 [cs]. Retrieved July 19, 2024 from http://arxiv.org/abs/2405.17739. Pre-published.

[53] James Prather et al. 2025. Beyond the Hype: A Comprehensive Review of Current Trends in Generative AI Research, Teaching Practices, and Tools. In *2024 Working Group Reports on Innovation and Technology in Computer Science Education*. ITiCSE 2024: Innovation and Technology in Computer Science Education. ACM, Milan Italy, (Jan. 22, 2025), 300–338. ISBN: 979-8-4007-1208-1. doi:10.1145/3689187.3709614.

[54] James Prather et al. 2023. The Robots Are Here: Navigating the Generative AI Revolution in Computing Education. In *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education*. ITiCSE 2023: Innovation and Technology in Computer Science Education. ACM, Turku Finland, (Dec. 22, 2023), 108–159. ISBN: 979-8-4007-0405-5. doi:10.1145/3623762.3633499.

[55] Nishat Raihan, Mohammed Latif Siddiq, Joanna C.S. Santos, and Marcos Zampieri. 2025. Large Language Models in Computer Science Education: A Systematic Literature Review. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSE TS 2025: The 56th ACM Technical Symposium on Computer Science Education. ACM, Pittsburgh PA USA, (Feb. 12, 2025), 938–944. ISBN: 979-8-4007-0531-1. doi:10.1145/3641554.3701863.

[56] Brent Reeves, Sami Sarsa, James Prather, Paul Denny, Brett A. Becker, Arto Hellas, Bailey Kimmel, Garrett Powell, and Juho Leinonen. 2023. Evaluating the Performance of Code Generation Models for Solving Parsons Problems With Small Prompt Variations. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1* (ITiCSE 2023). Association for Computing Machinery, New York, NY, USA, (June 30, 2023), 299–305. ISBN: 979-8-4007-0138-2. doi:10.1145/3587102.3588805.

[57] Elizabeth Reid. 2024. Generative AI in Search: Let Google do the searching for you. Google. (May 14, 2024). Retrieved Sept. 2, 2025 from https://blog.google/products/search/generative-ai-google-search-may-2024/.

[58] Lianne Roest, Hieke Keuning, and Johan Jeuring. 2024. Next-Step Hint Generation for Introductory Programming Using Large Language Models. In *Proceedings of the 26th Australasian Computing Education Conference*. ACE 2024: Australian Computing Education Conference. ACM, Sydney NSW Australia, (Jan. 29, 2024), 144–153. ISBN: 979-8-4007-1619-5. doi:10.1145/3636243.3636259.

[59] Kantwon Rogers, Michael Davis, Mallesh Maharana, Pete Etheredge, and Sonia Chernova. 2025. Playing Dumb to Get Smart: Creating and Evaluating an LLM-based Teachable Agent within University Computer Science Classes. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (CHI '25). Association for Computing Machinery, New York, NY, USA, (Apr. 25, 2025), 1–22. ISBN: 979-8-4007-1394-1. doi:10.1145/3706598.3713644.

[60] Kevin Roose. 2023. Don't Ban ChatGPT in Schools. Teach With It. *The New York Times. Technology*, (Jan. 12, 2023). Retrieved Sept. 11, 2025 from https://www.nytimes.com/2023/01/12/technology/chatgpt-schools-teachers.html.

[61] Jaromir Savelka, Arav Agarwal, Marshall An, Chris Bogart, and Majd Sakr. 2023. Thrilled by Your Progress! Large Language Models (GPT-4) No Longer Struggle to Pass Assessments in Higher Education Programming Courses. In *Proceedings of the 2023 ACM Conference on International Computing Education Research V.1*. ICER 2023: ACM Conference on International Computing Education Research. ACM, Chicago IL USA, (Aug. 7, 2023), 78–92. ISBN: 978-1-4503-9976-0. doi:10.1145/3568813.3600142.

[62] Jaromir Savelka, Arav Agarwal, Christopher Bogart, Yifan Song, and Majd Sakr. 2023. Can Generative Pre-trained Transformers (GPT) Pass Assessments in Higher Education Programming Courses? In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1* (ITiCSE 2023). Association for Computing Machinery, New York, NY, USA, (June 30, 2023), 117–123. ISBN: 979-8-4007-0138-2. doi:10.1145/3587102.3588792.

[63] Judy Sheard, Paul Denny, Arto Hellas, Juho Leinonen, Lauri Malmi, and Simon. 2024. Instructor Perceptions of AI Code Generation Tools - A Multi-Institutional Interview Study. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSE 2024: The 55th ACM Technical Symposium

on Computer Science Education. ACM, Portland OR USA, (Mar. 7, 2024), 1223–1229. isbn: 979-8-4007-0423-9. doi:10.1145/3626252.3630880.

[64] Jitendra Singh, Keely Steele, and Lovely Singh. 2021. Combining the Best of Online and Face-to-Face Learning: Hybrid and Blended Learning Approach for COVID-19, Post Vaccine, & Post-Pandemic World. *Journal of Educational Technology Systems*, 50, 2, (Dec. 2021), 140–171. doi:10.1177/00472395211047865.

[65] Adish Singla. 2023. Evaluating ChatGPT and GPT-4 for Visual Programming. In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 2* (ICER '23). Vol. 2. Association for Computing Machinery, New York, NY, USA, (Sept. 13, 2023), 14–15. isbn: 978-1-4503-9975-3. doi:10.1145/3568812.3603474.

[66] Jared Spataro. 2023. Introducing Microsoft 365 Copilot – your copilot for work. The Official Microsoft Blog. (Mar. 16, 2023). Retrieved Sept. 2, 2025 from https://blogs.microsoft.com/blog/2023/03/16/introducing-microsoft-365-copilot-your-copilot-for-work/.

[67] Mei Tan and Hari Subramonyam. 2024. More than Model Documentation: Uncovering Teachers' Bespoke Information Needs for Informed Classroom Integration of ChatGPT. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (CHI '24). Association for Computing Machinery, New York, NY, USA, (May 11, 2024), 1–19. isbn: 979-8-4007-0330-0. doi:10.1145/3613904.3642592.

[68] Lev Tankelevitch, Viktor Kewenig, Auste Simkute, Ava Elizabeth Scott, Advait Sarkar, Abigail Sellen, and Sean Rintel. 2024. The Metacognitive Demands and Opportunities of Generative AI. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. CHI '24: CHI Conference on Human Factors in Computing Systems. ACM, Honolulu HI USA, (May 11, 2024), 1–24. isbn: 979-8-4007-0330-0. doi:10.1145/3613904.3642902.

[69] Annapurna Vadaparty, Daniel Zingaro, David H. Smith Iv, Mounika Padala, Christine Alvarado, Jamie Gorson Benario, and Leo Porter. 2024. CS1-LLM: Integrating LLMs into CS1 Instruction. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*. ITiCSE 2024: Innovation and Technology in Computer Science Education. ACM, Milan Italy, (July 3, 2024), 297–303. isbn: 979-8-4007-0600-4. doi:10.1145/3649217.3653584.

[70] Jordan Novet Vanian Jonathan. [n. d.] Satya Nadella says as much as 30% of Microsoft code is written by AI. CNBC. Retrieved Sept. 9, 2025 from https://www.cnbc.com/2025/04/29/satya-nadella-says-as-much-as-30percent-of-microsoft-code-is-written-by-ai.html.

[71] Eric Von Hippel. 1986. Lead Users: A Source of Novel Product Concepts. *Management Science*, 32, 7, (July 1986), 791–805. doi:10.1287/mnsc.32.7.791.

[72] James D. Walsh. 2025. Everyone Is Cheating Their Way Through College. Intelligencer. (May 7, 2025). Retrieved Sept. 2, 2025 from https://nymag.com/intelligencer/article/openai-chatgpt-ai-cheating-education-college-students-school.html.

[73] Michel Wermelinger. 2023. Using GitHub Copilot to Solve Simple Programming Problems. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (SIGCSE 2023). Association for Computing Machinery, New York, NY, USA, (Mar. 3, 2023), 172–178. isbn: 978-1-4503-9431-4. doi:10.1145/3545945.3569830.

[74] Liwenhan Xie, Chengbo Zheng, Haijun Xia, Huamin Qu, and Chen Zhu-Tian. 2024. WaitGPT: Monitoring and Steering Conversational LLM Agent in Data Analysis with On-the-Fly Code Visualization. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. (Oct. 13, 2024), 1–14. arXiv: 2408.01703 [cs]. doi:10.1145/3654777.3676374.

[75] Litao Yan, Alyssa Hwang, Zhiyuan Wu, and Andrew Head. 2024. Ivie: Lightweight Anchored Explanations of Just-Generated Code. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. (May 11, 2024), 1–15. arXiv: 2403.02491 [cs]. doi:10.1145/3613904.3642239.

[76] Thomas Y. Yeh, Karena Tran, Ge Gao, Tyler Yu, Wai On Fong, and Tzu-Yi Chen. 2025. Bridging Novice Programmers and LLMs with Interactivity. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSE TS 2025: The 56th ACM Technical Symposium on Computer Science Education. ACM, Pittsburgh PA USA, (Feb. 12, 2025), 1295–1301. isbn: 979-8-4007-0531-1. doi:10.1145/3641554.3701867.

[77] Ryan Yen, Jiawen Stefanie Zhu, Sangho Suh, Haijun Xia, and Jian Zhao. 2024. CoLadder: Manipulating Code Generation via Multi-Level Blocks. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. UIST '24: The 37th Annual ACM Symposium on User Interface Software and Technology. ACM, Pittsburgh PA USA, (Oct. 13, 2024), 1–20. isbn: 979-8-4007-0628-8. doi:10.1145/3654777.3676357.

[78] J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. Why Johnny Can't Prompt: How Non-AI Experts Try (and Fail) to Design LLM Prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI '23: CHI Conference on Human Factors in Computing Systems. ACM, Hamburg Germany, (Apr. 19, 2023), 1–21. isbn: 978-1-4503-9421-5. doi:10.1145/3544548.3581388.

[79] Cynthia Zastudil, Magdalena Rogalska, Christine Kapp, Jennifer Vaughn, and Stephen MacNeil. 2023. Generative ai in computing education: Perspectives of students and instructors. In *2023 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–9. Retrieved Aug. 5, 2025 from https://ieeexplore.ieee.org/abstract/document/10343467/.

[80] Chunpeng Zhai, Santoso Wibowo, and Lily D. Li. 2024. The effects of over-reliance on AI dialogue systems on students' cognitive abilities: a systematic review. *Smart Learning Environments*, 11, 1, (June 18, 2024), 28. doi:10.1186/s40561-024-00316-7.