6. Parallel Literacies: Transdisciplinary Reflections on Language Teaching and Learning in the Age of AI

Joshua M. Paiz
https://otcid.org/0000-0002-9322-5246

Abstract

As artificial intelligence (AI) embeds itself in modern systems, disciplinary boundaries blur, creating opportunities for transdisciplinary insights. This chapter reflects on the intersection of English language teaching and engineering education. As both a longtime English educator and a graduate student in computer science, I examine how AI has shaped my teaching and learning, revealing shared connections and challenges.

Drawing on these dual roles, I explore parallels between natural and programming languages across their syntactic, semantic, and pragmatic dimensions. I consider how AI tools like ChatGPT, Claude, and GitHub Copilot have transformed language teaching by offering feedback and scaffolding learner autonomy. At the same time, programming assistants integrated into development environments have both supported and complicated my growth as a computer language learner. Pedagogical strategies from programming—such as iterative problem-solving and debugging—also suggest useful approaches for language instruction, and vice versa.

I highlight challenges of AI integration, including overreliance, learner agency, and equitable access. Ultimately, this chapter argues that AI can bridge disciplines and foster innovative practices when used thoughtfully—as a partner that supports, rather than replaces, human critical thinking and creativity. Purposeful training remains essential to prevent learning loss and maintain pedagogical rigor.

Keywords: artificial intelligence, language teaching, computer science education, transdisciplinary

We have all spent considerable time thinking about and talking about the latest elephant in the educational room. If you've picked up this volume, there can be little doubt that it is, at least in part, because you are wrestling with that earlier-mentioned pachyderm. In case of any uncertainty, I am referring to artificial intelligence (AI), or, more specifically, generative artificial intelligence (GenAI). We educators are all grappling with, wrestling with, and coming to terms with the implications of AI. Even those of us increasingly active in the fields of computer science and engineering education are fighting against its potentially detrimental effects on teaching and learning. Granted, our focus may center more on helping humans communicate effectively with machines and, eventually, with other human users.

If such a diverse group of educators from disparate disciplinary backgrounds are having the same worries, it is worth exploring what we can learn from one another. As someone active in both applied linguistics and applied computer science, I aim to distill transdisciplinary insights in this chapter. By transdisciplinary, I refer to an integrative approach that transcends traditional boundaries, creating new frameworks by combining perspectives, methodologies, and epistemologies (Douglas Fir Group, 2016; Perrin & Kramsch, 2018). With growing calls for such work in both fields, this chapter explores how a transdisciplinary lens can help navigate the shared challenges of teaching and learning in the age of AI.

Issues That Motivated the Research

Artificial intelligence (AI) is redefining the boundaries of traditional disciplines, particularly language education and computer science. While fields like CALL (computer-assisted language learning) and MALL (mobile-assisted language learning) have bridged these domains for decades (Chinnery, 2006; Levy, 1997), tools like ChatGPT and GitHub Copilot have accelerated this convergence. Table 6.1, for instance, shows we can understand AI in educational settings as falling across different categorizations based on use-case, integration with existing tools, or underlying technology. What is important is knowing what kind of AI tool we are considering and working with, as this will dri ve our exploration of how that tool may be used in our educational practice. This shift reflects a deeper move toward transdisciplinary thinking in education, revealing both challenges and opportunities for innovation. And, we have seen early movement in this area across the educational sector as evidenced by the strong AI-focus of recent ed-tech conferences like ISTE/ASCD and the collaboration of professional organizations and government entities to drive conversations around AI in diverse educational settings (see ISTE, 2025; Paiz, 2025; UMD, 2025). We have also seen scholarly and research work in the area of AI in ELT already emerge to support extensive literature reviews that highlight a range of emerging subtopics ranging from automated feedback systems and basic chatbots to multimodal emotion recognition systems and culturally-sensitive adaptive learning environments (see Crompton et al., 2024; Sharadgah & Sa'di, 2022).

AI Category	Examples	Educational Uses
Large Language Models (LLMs)	ChatGPT, Claude, Gemini, Perplexity	Writing assistance, content generation, tutoring, research support
Code Assistants	GitHub Copilot, Android Studio (Gemini), Replit AI, Cursor	Code completion, debugging, programming support, syntax help
Integrated AI Features	MS Office Editor, Grammarly, Google Docs Smart Compose	Grammar/style checking, auto-completion, writing suggestions
Specialized Educational AI	Khan Academy's Khanmigo, Duolingo Max, Socratic	Personalized tutoring, adaptive learning, problem-solving support
Multimodal AI	DALL-E, Midjourney, Canva AI	Visual content creation, design assistance, presentation support

Table 6.1. Categories of AI tools in educational settings

The Convergence of Language and Technology

Language serves as both an innate human trait and a technology, mediating human existence and computer interactions alike. Programming languages share structural similarities with natural languages, using syntax, semantics, and symbolic conventions to translate binary code into human-comprehensible forms (see Figure 6.1).

```
section .data
    num1 db 5         ; Declare first number (5)
    num2 db 10        ; Declare second number (10)
    result db 0       ; Placeholder for result

section .text
    global _start     ; Declare the entry point

_start:
    mov al, [num1]    ; Load num1 into the AL register
    add al, [num2]    ; Add num2 to the value in AL
    mov [result], al  ; Store the result in the result variable

    ; Exit the program
    mov eax, 60       ; Syscall for exit
    xor edi, edi      ; Return code 0
    syscall
```

```
#include <stdio.h>

int main() {
    // Declare variables
    int num1 = 5;         // First number
    int num2 = 10;        // Second number
    int result = 0;       // Placeholder for result

    // Add numbers
    result = num1 + num2;

    // Print the result
    printf("The result is: %d\n", result);

    return 0;             // Exit the program
}
```

Figure 6.1. Examples of computer code in Assembly (left) and C (right).

Large language models and neuro-symbolic AI systems exemplify this language-technology convergence. These systems combine neural networks' pattern recognition with symbolic reasoning's structured logic, enabling sophisticated language processing and reasoning tasks (Hitzler et al., 2022). This integration underscores language's role as the fulcrum between humans and machines, necessitating transdisciplinary approaches to leverage AI's capabilities responsibly in education.

My journey from English language teaching to computer science illuminated these parallels firsthand. Despite failing as a Computer Science undergraduate student in 2004, the AI-education intersection compelled my return to the field. Programming's linguistic echoes—structured syntax, meaningful symbols, communication rules—revealed that both natural and programming languages are fundamentally tools for communication, one mediating human relationships, the other human-machine interaction.

Contemporary Educational Challenges

AI integration presents three primary challenges for education. First, AI tools' growing prevalence requires significant pedagogical adjustments. Microsoft Office has included AI since 2013 (Raja, 2015), and platforms like Android Studio now feature AI assistance by default (Android, 2024). While these tools offer personalized learning and enhanced feedback (Tajik, 2025; Zhao, 2024), they risk fostering over-reliance and diminishing critical thinking (Alqahtani & Wafula, 2024). Second, AI serves dual roles as both teaching assistant and potential disruptor. While AI provides personalized feedback and automates routine tasks (Gökçearslan et al., 2024), concerns about algorithmic bias, equitable access, and learner autonomy remain paramount (Crawford, 2020; Pasquale, 2020). Third, literacy in the AI era must expand beyond traditional skills to encompass AI literacy—understanding AI tools, data ethics, and computational thinking

(Celik, 2023). This expanded literacy enables critical evaluation of AI-generated information and informed decision-making (Walter, 2024).

These challenges underscore the urgency of preparing learners and educators for an AI-driven future through transdisciplinary approaches that bridge language education and computer science, a potential path forward that I will explore throughout this chapter. However, comprehensively addressing these challenges will take concerted and collaborative disciplinary effort (see Paiz et al., 2025).

Context of the Research

Personal/Professional Context

This chapter emerges from my dual professional and academic journey—one that spans applied linguistics and applied computer science. As an English language educator, I have long emphasized the importance of structured learning environments that encourage students to take ownership of their knowledge development. This belief aligns with sociocultural theories of learning, which highlight the role of guided interaction and scaffolding in developing expertise (Lantolf & Thorne, 2006; Vygotsky, 1986). My transition into computer science was driven by shifts in international education and the increasing presence of AI in both language teaching and engineering education (Johnson, 2024; Paiz et al., 2025). What began as an effort to explore alternative career paths quickly evolved into an inquiry into the intersections between these fields—an exploration of the transdisciplinary nature of AI-mediated teaching and learning.

Educational Technology Landscape

The technological landscape of education is evolving rapidly with AI integration. While programming AI tools like GitHub Copilot are embedded directly into the creation process, shifting focus from syntax to systems thinking, language learning AI tools tend to function as practice aids that supplement rather than transform instruction (Ghimire & Edwards, 2024; Zhao, 2024). This fundamental difference raises key questions about how AI-mediated learning varies across disciplines.

Theoretically, this work is informed by sociocognitive approaches to second language acquisition (SLA), which emphasize the role of interaction, cognition, and context in shaping learning experiences (Atkinson, 2011; Lantolf & Thorne, 2006). These perspectives align with key pedagogical principles in programming education, computational thinking, and problem decomposition that help learners approach complex problems methodically (Guzdial, 2015; Wing, 2017). By examining these frameworks in parallel, I seek to explore how AI mediates both linguistic and computational literacy, and how insights from one domain might inform pedagogical strategies in the other.

This chapter, therefore, lays the groundwork for exploring the role of AI in shaping learner autonomy, the pedagogical affordances of AI across disciplines, and the implications of AI integration for education more broadly. Through a comparative, transdisciplinary reflection, I examine how these themes intersect, drawing on personal teaching and learning experiences, AI tool interactions, and cross-disciplinary insights to inform both research and practice.

Research Questions
Given the increasing convergence of AI, language education, and programming pedagogy, this study explores the following primary research questions:

1. How does AI mediate teaching and learning experiences across language and programming education?
2. How can insights from programming education inform language teaching and vice versa?

These questions are further supported by secondary inquiries that examine specific pedagogical challenges:

- What role does AI play in scaffolding learner autonomy?
- How do AI tools support or hinder the development of language competencies in educational contexts?
- What are the broader pedagogical implications of AI integration for teaching and learning?

By addressing these questions, this chapter aims to provide a transdisciplinary analysis of AI's role in shaping pedagogical practice, learner agency, and knowledge construction across disciplines.

Research Method: A Transdisciplinary Autoethnographic Approach
This study adopts a reflexive autoethnographic methodology (Ellis et al., 2011) that integrates theoretical perspectives from sociocognitive SLA (Atkinson, 2011; Lantolf & Thorne, 2006) and computational problem-solving frameworks (Guzdial, 2015; Wing, 2017). By combining Lawrence and Nagashima's (2020) technological auto-documentation protocols with Hunt's (2008) critical incident analysis, I explore the intersections of AI-mediated learning across linguistic and computational domains.

Data Collection Procedures
Data collection involved three interrelated processes: (1) multimodal artifact analysis of teaching materials and computational artifacts, documenting 127 ChatGPT interactions for language instruction and 84 AI-assisted coding sessions; (2) an 18-month reflexive journal capturing critical learning moments in both domains; and (3) transcribed screen recordings of AI sessions examining prompt engineering patterns and decision-making processes. These data streams addressed both research questions: (1) examining AI's mediation of teaching/learning (RQ1) and (2) revealing transferable cross-disciplinary insights (RQ2), particularly regarding autonomy scaffolding and competency development.

Data Analysis
To analyze these data sources, I employed a grounded theory approach (Yazan, 2019) following three stages: (1) open coding to identify emergent themes, (2) axial coding to map themes onto theoretical constructs from SLA and computational thinking, and (3) selective coding to synthesize patterns of transdisciplinary AI mediation. Building on Hunt's (2008) critical incident analysis, I examined 23 pivotal AI interactions that significantly influenced my learning process. The analytical framework reflects the sociocognitive foundations of this research, drawing on Lantolf and Thorne's (2006) scaffolding concepts and Shibata's (2012) distributed cognition model to examine how AI both facilitates and constrains learner autonomy across disciplines.

Results and Discussion

In this section, I present the central findings of my autoethnographic exploration into the intersections of language teaching and computer programming. The discussion is organized into four subsections that explore key dimensions of this experience. First, I will explore the parallels between language and programming, focusing on syntax, semantics, pragmatics, and learning progressions. Then, I will discuss how AI shapes and supports (or hinders) my learning experiences. Next, I will seek to highlight pedagogical strategies emerging from the convergence of these domains, and I will conclude by examining the challenges and limitations that arise from relying on technology as both a scaffold and potential disruptor. Through these interconnected discussions, I aim to illuminate the transdisciplinary nature of AI's role in shaping teaching and learning across two seemingly disparate fields.

Before exploring these findings in detail, it's worth revisiting the central questions that guided this autoethnographic investigation: (1) How does AI mediate teaching and learning experiences across language and programming education, and (2) how can insights from programming education inform language teaching and vice versa? The experiences and insights that follow directly illuminate these questions and their supporting inquiries about learner autonomy, competency development, and broader pedagogical implications. Through the four interconnected themes explored below, I demonstrate how AI tools both facilitated and complicated my learning across these domains while revealing new possibilities for transdisciplinary education.

AI-mediated Learning Experiences

One of the most immediate effects I encountered was AI's ability to offer rapid, targeted feedback on both my academic writing and my coding projects. In the context of a literature review, ChatGPT suggested rephrasings for clunky sentences, mirroring the kind of stylistic refinements that I regularly taught in my English language classes. Meanwhile, in a database management project that I was helping supervise, GitHub Copilot auto-completed Structured Query Language (SQL) queries in ways I wouldn't have initially considered and in a fraction of the time that it would have taken me to devise the query if it involved complex queries. These AI "scaffolds" accelerated my revisions, revealing the potential for immediate formative feedback (Alqahtani & Wafula, 2024). Yet I sometimes found myself trusting the suggestions too readily, underlining how AI support can both enhance and complicate the learning process.

Indeed, my reliance on AI tools soon raised questions about learner agency, both my own and that of the students I teach. On one hand, freeing cognitive resources by offloading routine checks to ChatGPT or Copilot allowed me to focus on higher-order tasks, such as discourse coherence in writing or overall system design in coding (Wing, 2017). On the other hand, there were moments when I realized I was no longer pushing myself to explore alternatives, and certainly moments where my command over programming principles could be called into question, such as during an exam where I froze on a question about designing a certain data structure that I had implemented multiple times in course homework assignments. If the professor had not been gracious with directed prompting and forced me to think through the problem, I certainly would have failed it. This tension echoes sociocognitive insights that learners benefit most from guided exploration and co-cognition with their interlocutors, but may risk stagnation if over-reliant on the affordances in their ecosocial space (see Atkinson, 2011; Lantolf & Thorne, 2006).

The advantages of AI—personalized feedback, rapid iteration, and immediate error-flagging—were evident, but so were the pitfalls. In one instance, ChatGPT produced a misleading explanation of a data structures algorithm, requiring me to verify its accuracy against course materials. Similarly, Copilot once suggested code that ran but introduced hidden inefficiencies—overly long compute times and excessive resource consumption. These incidents reinforced findings that while AI can expedite learning, inaccuracies or biases in outputs require vigilant user oversight (Mishara, 2024; Paiz, 2024). Experiencing these affordances and constraints in both language and programming revealed a common need for effective prompt engineering, reflective practice, and critical evaluation (Guzdial, 2015). Whether refining an argumentative paragraph or debugging a Python script, awareness of AI's strengths and limitations underscored how transdisciplinary approaches can optimize learning and safeguard autonomy in an era increasingly shaped by intelligent systems.

Coding Principles and Language Pedagogy

This research raised for me the need to explore how the intersection of coding principles and advanced language pedagogy can yield innovative approaches to teaching and learning. This became even more apparent when I was assigned to teach an English for academic purposes (EAP) class for engineering students. Previously my teaching had been focused on general English language instruction. By integrating systematic debugging, iterative design, peer-review techniques, and the transfer of concepts across programming and language writing, I illustrate how each domain's strengths can inform the other. In doing so, I highlight four key strategies—borrowing logic structures for text organization, applying editorial methodologies in code, foregrounding iterative problem-solving, and adopting a debugging mindset—that underscore the potential of transdisciplinary exchange.

Transfer of Programming Concepts to Language Writing

In preparing a new book chapter on AI in higher education—a process steeped in complex rhetorical moves—I realized that programming's systematic logic could inform my revision strategies. For instance, I asked ChatGPT to identify "breakpoints" in my manuscript where the narrative flow seemed to falter, treating these points like debugging flags in code (Guzdial, 2015). The AI suggested reorganizing certain paragraphs around more coherent transitions, akin to restructuring functions for clarity. By pinpointing segments with inconsistent arguments, ChatGPT essentially mirrored how compilers catch syntax errors, reminding me that iterative refinement can apply just as effectively to high-level scholarly writing. For example, in an early draft I wrote: "AI tools have become common in classrooms. These tools can restructure how students think about revision. Revision is central to academic success, but it is often taught as a one-step process." ChatGPT flagged the middle sentence as misplaced, noting that I introduced "revision" before clearly establishing its importance. It suggested moving the definition of revision forward, producing a smoother flow: "AI tools have become common in classrooms. Revision, a process central to academic success, is often taught as a one-step task. These tools can restructure how students think about revision." By isolating the misplaced idea and repositioning it, the AI effectively highlighted a structural 'syntax error,' much like a compiler would in code. The change clarified my argument and made the progression of ideas more coherent.

Application of Language Teaching Principles to Programming

Conversely, principles from advanced academic writing, such as maintaining a strong authorial stance (Hyland, 2009) and engaging with counterarguments, proved surprisingly useful in crafting more readable code. During a recalled AI chat, I asked how to adapt peer-review approaches from my editorial work to a group coding assignment. ChatGPT recommended “code reviews” that emulate scholarly peer feedback, stressing the importance of not just identifying bugs but also challenging underlying logic. This communicative dynamic, common in writing workshops, resonated in collaborative programming environments.

Role of Iterative Problem-Solving

Whether refining a theoretical framework or debugging a recursive function, I encountered similar cycles of trial, error, and revision. On the drive home one day, I put ChatGPT into voice mode and discussed these parallels with it to help think through potential applications to my teaching practice. It pointed out that just as I draft multiple versions of a conclusion to ensure it aligns with my argument, I run and re-run code to verify performance. In both cases, each iteration brought me closer to a polished final product (Wing, 2017). By incorporating iteration, development, and appropriate scaffolds into classroom practice, we can better support students’ growth in linguistic proficiency. For example, in an intermediate presentations class we might start with an assignment on outlining, where learners produce a rough outline of their talk based on a given prompt. After receiving targeted feedback either from the instructor or peers, they revise the outline, fine-tuning main points and transitions much like they would refine function calls or algorithmic steps in a coding environment. Each iteration adds clarity and substance, ensuring that by the time students finalize their speaking notes and practice delivery, they have gone through multiple cycles of planning, revising, and polishing mirroring the systematic improvements inherent to iterative software design. This may feel similar to how we already scaffold writing tasks as iterative design requires learners to also *revisit* and *revise* their work multiple times rather than moving through a single linear sequence. Each cycle of outline-creation, feedback, and revision drives deeper engagement with both content and language use, prompting learners to test, refine, and eventually automate new skills (Wing, 2017). By foregrounding these repeated rounds of improvement rather than a one-shot “draft and done” process akin to what we see in traditional in-class essay exams that are making a come back (Shibu, 2025), iterative design offers a systematic framework for continuous development—a methodology that mirrors the multiple passes of debugging and optimization in coding, yet is distinctly applied to language learning.

Importance of Debugging Approaches

Finally, debugging served as an instructive metaphor for addressing issues in text and code. When ChatGPT flagged a disjointed section of my chapter, I “stepped through” each claim to isolate the flawed logic much like running a debugger line by line. Here, I actually engaged in a form of mock debate with the AI system to create greater context for the revision session and to ensure that I received more actionable recommendations from the system to revise and refine my work. This process encouraged deeper insight rather than hasty patchwork fixes, echoing best practices in both error correction and advanced academic revision. Indeed, a parallel process emerged in a computer science class, where I would regularly use Claude to refine my data model, as I had never designed a non-relational (NoSQL) database before (see Fitzgerald et al., 2008; Vuogan & Li 2023). I also used it to help refine my user interface design, having at that time not yet taken any course work that explicitly focused on human-computer interfaces.

From a language-teaching perspective, adopting this "debugging mindset" means treating each sentence and paragraph like a line of code that needs to be "stepped through" for logical, structural, and stylistic integrity. Rather than relying on quick, surface-level fixes, teachers can encourage students to engage in more deliberate, debate-like revision sessions, whether with peers or AI tools, where each claim is interrogated for cohesion and clarity. This iterative approach promotes deeper engagement with the text, nudging learners to explore why certain phrasing or organization might break down. In so doing, educators help students move beyond patchwork edits toward a more robust understanding of how to craft and sustain complex written arguments.

Challenges and Limitations

In my own journey, I noticed a tendency to lean on AI tools for quick fixes like refining an awkward sentence or auto-completing a coding function. While this often saved time, it risked eroding deeper problem-solving abilities. As other researchers have warned (Mishara, 2024), an over-dependence on AI can short-circuit the cognitive processes essential to authentic learning and skill mastery. This then means that ensuring that learners (including myself) remain in control calls for explicit strategies to encourage critical evaluation of AI outputs. In my writing practice, this translated to re-checking ChatGPT's suggestions against primary sources. While in coding tasks I manually tested Copilot-generated solutions to confirm correctness. By framing AI as a scaffold rather than a final authority, educators can foster metacognitive awareness and self-regulation (Lantolf & Thorne, 2006).

Another critical concern became abundantly clear while I was working with the language teachers in Bahrain's Ministry of Education during a U.S. Department of State-sponsored educational exchange that I participated in during the spring of 2024. This trip would see me leading 25 training seminars for K-12, university, and vocational educators introducing the notion of critical AI literacy. The issue that came up time and again was that the widening digital divide for those without reliable internet or advanced computing resources will come to have more profound impacts in our classrooms and the workplace (Walter, 2024). Learners in under-resourced environments may miss out on AI's benefits entirely, deepening existing achievement gaps. Conversely, students from working class backgrounds may be at a disadvantage entering the labor market compared to students who can pay for powerful AI interview tutors or for access to more performant models to help craft better application documents. Even among those with access, uneven AI literacy can further complicate effective AI integration, underscoring the importance of institutional and systemic support. Ultimately, AI should complement and not replace hands-on practice and reflective thinking. Whether composing an academic paper or debugging a JavaScript function, learners need opportunities to grapple with complexity on their own (Wing, 2017). Preserving this balance helps maintain the integrity of the learning experience while still reaping AI's many advantages.

Implications for Policy, Practice, and Future Research

Building on the reflections, findings, and discussions presented thus far, this final section addresses three core areas where the insights from this transdisciplinary inquiry can be operationalized. Here I consider the policy implications of integrating AI into educational contexts, explore practice implications for classroom strategies and approaches, and offer future research directions, identifying fruitful avenues and emerging questions at the intersection of AI, language pedagogy, and computer science education.

A recurring theme throughout my experiences was the dual role of AI as both an accelerator of innovation and a potential disruptor of foundational learning. From a policy standpoint, educational institutions and governmental bodies should develop clear guidelines that define when and how AI tools can be used in academic settings. Much like established policies for academic integrity (Pasquale, 2020), these new guidelines would specify acceptable use cases; for instance, whether AI-generated code snippets or AI-edited writing are permissible for graded assignments and under what conditions. By articulating these standards, policymakers can help educators navigate ethical quandaries, protect learners from over-dependence, and ensure that AI complements rather than supplants the development of core competencies (Mishara, 2024).

Policymakers and administrators must also provide institution-wide frameworks that keep teachers and students focused on meaningful learning outcomes. For example, universities might adopt a “human-in-the-loop” principle requiring that any AI-generated material be reviewed and revised by students themselves (Walter, 2024). Similarly, accreditation bodies could mandate that instructors include critical AI literacy components, such as prompt engineering, bias detection, and data ethics, into curricula across disciplines (Tajik, 2025; Lam et al., 2024). These requirements would help safeguard the rigor of educational programs, ensuring that AI-driven learning aligns with broader institutional goals like fostering critical thinking and ethical scholarship (Crawford, 2020).

Finally, professional development (PD) opportunities are essential if educators are to harness AI responsibly (Paiz et al., 2025). My own journey highlighted the steep learning curve in understanding how AI tools operate and how best to integrate them into classroom instruction. Workshops or certification courses focused on AI pedagogical strategies—encompassing everything from designing AI-friendly assignments to teaching prompt engineering—would empower teachers to adapt resources like ChatGPT and GitHub Copilot for various linguistic and computational tasks. Moreover, PD offerings should emphasize reflexive practices, encouraging educators to assess their own reliance on AI, mitigate biases in AI-generated feedback, and remain vigilant about data privacy (Alqahtani & Wafula, 2024).

Taken together, these policy-level strategies can help educational systems leverage AI’s benefits while minimizing potential drawbacks. By embedding critical AI competencies into institutional policies, schools and universities can proactively address ethical, logistical, and pedagogical challenges, thereby setting the stage for more equitable and transformative learning experiences.

At the classroom level, my autoethnographic experiences suggest that intentional, scaffolded use of AI tools can enhance both language and programming education (Guzdial, 2015; Lantolf & Thorne, 2006). For instance, teachers might design assignments where learners first attempt a writing prompt or coding exercise independently, then consult an AI to compare approaches, identify errors, or generate alternative solutions. The key is to structure such activities so that students actively evaluate AI outputs—looking for inaccuracies, testing efficiency, and reflecting on the underlying logic or language choices (Raymond, 2003). This approach guards against passive AI consumption and fosters deeper cognitive engagement.

To keep learners at the center of the process, instructors can embed metacognitive checkpoints in AI-mediated tasks. For example, in an EAP setting, requiring students to articulate the “why” behind each AI-suggested revision encourages them to go beyond surface-level corrections. Likewise, in programming labs, learners might briefly journal about why the AI-generated code works (or fails), thereby internalizing algorithmic thinking. Such reflective practices are particularly vital given the risk of complacency when AI readily supplies solutions (Wing, 2017). Educators can use rubrics emphasizing not only correctness but also reflection on process, ensuring that AI remains an aid rather than a crutch.

My experiences highlight how AI can spur learners to examine their assumptions and test new ideas, provided it is introduced as a dialogic partner rather than an oracle. Teachers could, for instance, arrange “Socratic seminars” with AI involvement, prompting students to ask structured follow-up questions, challenge the AI’s reasoning, and compare multiple AI-sourced perspectives. In coding projects, small groups might iteratively refine AI-generated functions while analyzing computational complexity and resource usage. When educators situate AI within an inquiry-based framework (Atkinson, 2011), they encourage students to develop habits of questioning, checking evidence, and refining their work—skills that are essential for both language learning and computational problem-solving.

Paradoxically, while AI can generate text or code swiftly, it can also bolster creative thinking by freeing learners from routine mechanical tasks (Styve et al., 2024). For example, if a language-learner no longer struggles with basic grammar checks, they may devote more effort to thematic development or rhetorical flair. In programming, if Copilot streamlines repetitive boilerplate tasks, students can focus on higher-level design patterns and problem-solving strategies. To capitalize on this potential, instructors can design “creative expansion” activities where, after accepting an AI’s initial suggestions, learners must modify or enrich them to reflect personal style, advanced rhetorical moves, or optimized algorithmic structures. This not only preserves originality but also encourages meta-creative processes.

Taken as a whole, these practice implications underscore the importance of structured engagement, reflection, and creative adaptation in AI-mediated classrooms. By designing assignments that promote critical thinking and encourage learner autonomy, educators can make technology a catalyst for deeper, more meaningful learning experiences across both language and programming contexts.

While this work lays foundational insights, many unanswered questions remain. For instance, how do different types of AI models (e.g., large language models vs. more specialized tutoring systems) affect learning outcomes in multilingual classrooms vs. monolingual programming labs (Zhao, 2024)? Does the scaffolding effect differ for students at varying levels of proficiency in language or coding? Multi-site comparative studies spanning different institutions and learner demographics could shed light on how context-specific variables shape the efficacy of AI interventions.

One pressing need is for longitudinal research that tracks learners over multiple terms or academic years. We need to be able to assess, as a discipline, whether or not learners become more independent and reflective over time, or whether reliance on AI steadily increases. A longitudinal lens could illuminate whether early AI exposure translates into sustained competence, or whether new cycles of dependency emerge as AI capabilities evolve.

My experience underscores the value of cross-pollination between language education and computer science. Future research might involve co-teaching, where instructors from both disciplines design integrated modules, such as a collaborative project requiring both an essay and a working prototype, with AI providing feedback in each domain. Researchers could measure outcomes like transfer of problem-solving strategies, improvements in rhetorical awareness, or heightened computational thinking. By blending qualitative (e.g., reflective journals, interviews) and quantitative (e.g., performance metrics, usage logs) methods, such studies could yield comprehensive insights into how AI facilitates or complicates transdisciplinary learning.

Finally, the rapid pace of AI innovation raises new ethical, technical, and pedagogical questions. How might generative AI shift as models become more context-aware, potentially blurring lines between "human" and "machine" authorship even further (Crawford, 2020; Liu & Lin, 2018)? What design principles can ensure AI remains transparent in its suggestions—allowing learners to distinguish system logic from their own? As different cultures, languages, and coding paradigms interact with AI, will new forms of digital literacy emerge that we have yet to conceptualize (Paiz, 2024)? Addressing these issues will demand not only technical advances, but also renewed collaboration across linguistics, computer science, sociology, and education.

Moving forward, the challenge is to balance AI's transformative potential with a steadfast commitment to genuine learning—one that values critical thinking, learner agency, and ethical responsibility. The policy, practice, and research directions outlined here can serve as guiding pillars for educators, policymakers, and scholars aiming to harness AI's power while mitigating its pitfalls. By fostering intentional design, reflective engagement, and cross-disciplinary scholarship, we can ensure that AI remains a robust ally to human creativity and inquiry, rather than a substitute for them.

References

Alqahtani, N., & Wafula, Z. (2024). Artificial intelligence integration: Pedagogical strategies and policies at leading universities. *Innovative Higher Education*, 1-20. https://doi.org/10.1007/s10755-024-09749-x

Android. (2024). AI-assisted coding. *Android Developers*. Retrieved from: https://developer.android.com/studio/preview/gemini/ai-code-completion

Atkinson, D. (2011). *Alternative approaches to second language acquisition*. Routledge.

Celik, I. (2023). Exploring the determinants of artificial intelligence (Ai) literacy: Digital divide, computational thinking, cognitive absorption. *Telematics and Informatics*, *83*, 102026.

Chinnery, G. M. (2006). Going to the MALL: Mobile assisted language learning. *Language Learning & Technology, 10*(1), 9-16.
Crawford, K. (2020). *The atlas of AI: Power, politics, and the planetary costs of artificial intelligence*. Yale University Press.
Crompton, H., Edmett, A., Ichaporia, N., & Burke, D. (2024). AI and English language teaching: Affordances and challenges. *British Journal of Educational Technology*, *55*(6), 2503-2529. https://doi.org/10.1111/bjet.13460
Douglas Fir Group. (2016). A transdisciplinary framework for SLA in a multilingual world. T*he Modern Language Journal, 100*(S1), 19-47. https://doi.org/10.1111/modl.12301
Ellis, C., Adams, T. E., & Bochner, A. P. (2011). Autoethnography: An overview. *Forum: Qualitative Social Research, 12*(1) https://doi.org/10.17169/fqs-12.1.1589 .
Fitzgerald, S., Lewandowski, G., McCauley, R., Murphy, L., Simon, B., Thomas, L., & Zander, C. (2008). Debugging: Finding, fixing and flailing, a multi-institutional study of novice debuggers. *Computer Science Education*, 18(2), 93-116. https://doi.org/10.1080/08993400802114508.
Ghimire, A., & Edwards, J. (2024). Coding with AI: How are tools like ChatGPT being used by students in foundational programming courses. In A. M. Olney, I. A. Chounta, Z. Liu, O. C. Santos, & I. I. Bittencourt (Eds.), *Artificial Intelligence in Education*. *AIED 2024.* Lecture Notes in Computer Science, vol 14830. Springer. https://doi.org/10.1007/978-3-031-64299-9_20
Gökçearslan, Ş., Tosun, C., & Erdemir, Z. G. (2024). Benefits, challenges, and methods of artificial intelligence (AI) chatbots in education: A systematic literature review. *International Journal of Technology in Education*, *7*(1), 19-39. https://doi.org/10.46328/ijte.600
Guzdial, M. (2015). *Learner-centered design of computing education: Research on computing for everyone*. Morgan & Claypool Publishers.
Hitzler, P., Eberhart, A., Ebrahimi, M., Sarker, M. K., & Zhou, L. (2022). Neuro-symbolic approaches in artificial intelligence. *National Science Review*, *9*(6), nwac035.
Hunt, A. (2008). Critical incident technique. In P. J. Lavrakas (Ed.), *Encyclopedia of survey research methods* (pp. 127-129). SAGE.
Hyland, K. (2009). *Academic discourse: English in a global context*. Continuum.
ISTE. (2025). *ISTELive 2025 Conference.* https://conference.iste.org/2025/
Johnson, M. (2024). Generative AI and CS education. *Communications of the ACM, 67*(4), 23–24. https://doi.org/10.1145/3632523
Lam, K., et al. (2024) A framework for assurance audits of algorithmic systems. *FaccT '24: Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency* (pp. 1078-1092). Association for Computing Machinery. https://doi.org/10.1145/3630106.3658957
Lantolf, J. P., & Thorne, S. L. (2006). *Sociocultural theory and second language learning*. Oxford.
Lawrence, G., & Nagashima, Y. (2020). Auto-documentation in technological research. *Qualitative Inquiry, 26*(7), 801-812.
Levy, M. (1997). *Computer-assisted language learning: Context and conceptualization.* Oxford University Press.
Liu, K., & Lin, Z. (2018). Ethical AI research protocols. *ACM Transactions on Computing Education, 18*(4), 1-22. https://doi.org/10.1145/3634685

Mishara, P. (2024). The ethical implications of AI in education: Privacy, bias, and accountability. *Journal of Informatics Education and Research, 4*(2). https://doi.org/10.52783/jier.v4i2.1827
Paiz, J. M. (2025). Harnessing AI to revolutionize global English language teaching. *U.S. Department of State Focus Session - ISTELive2025/ASCD*. San Antonio, TX.
Paiz, J. M. (2024). Critical AI literacy & ELT: Teaching in an AI-rich world. *IATEFL Language Technology SIG Newsletter* (Spring 2024). IATEFL.
Paiz, J. M., Tonecelli, R., & Kostka, I. (2025). *Artificial intelligence, real teaching: A guide to AI in ELT*. University of Michigan Press.
Pasquale, F. (2020). *The new laws of robotics.* Harvard University Press.
Perrin, D., & Kramsch, C. (Eds.). (2018). Transdisciplinarity in applied linguistics (Focus Issue). *AILA Review, 31*. https://doi.org/10.1075/aila.31
Raja, S. (2015). *Computers make the best listeners: A history and experiment on natural language processing* [Unpublished Masters Thesis]. Florida Atlantic University.
Raymond, E. S. (2003). *The art of Unix programming*. Addison-Wesley Professional.
Sharadgah, T. A., & Sa'di, R. A. (2022). A systematic review of research on the use of artificial intelligence in English language teaching and learning (2015-2021): What are the current effects? *Journal of Information Technology Education: Research*, *21*, 337-377. https://doi.org/10.28945/4999
Shibata, H. (2012). Distributed cognition in programming. *Journal of CS Education, 29*(2), 45-67. https://doi.org/10.1145/2787622.2787715
Shibu, S. (2025). College professors are turning to an old-school product from a family-owned business to combat AI cheating. *Entrepreneur.* https://www.entrepreneur.com/business-news/college-professors-turn-back-to-blue-books-to-combat-chatgpt/492450
Styve, A., Virkki, O. T., & Naeem, U. (2024). Developing critical thinking practices interwoven with generative AI usage in an introductory programming course. *2024 IEEE Global Engineering Education Conference (EDUCON)*, 01–08. https://doi.org/10.1109/educon60312.2024.10578746
Tajik, A. (2025). Exploring the role of AI-driven dynamic writing platforms in improving EFL learners' writing skills and fostering their motivation. *Research Square*. https://doi.org/10.21203/rs.3.rs-5788599/v1
UMD. (2025). *AI Ready Maryland AI in Education Summit*. Canton, MD.
Vuogan, L., & Li, S. (2023). Examining the effectiveness of peer feedback in second language writing: A meta-analysis. *TESOL Quarterly*, *57*(4), 1115-1138. https://doi.org/10.1002/tesq.3178.
Vygotsky, L. S. (1986). *Thought and language* (A. Kozulin, Translator). MIT Press
Walter, Y. (2024). Embracing the future of Artificial Intelligence in the classroom: The relevance of AI literacy, prompt engineering, and critical thinking in modern education. *International Journal of Educational Technology in Higher Education, 21(*1). https://doi.org/10.1186/s41239-024-00448-3
Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology, 25*(2), 7-14. https://doi.org/10.17471/2499-4324/922
Yazan, B. (2019). An autoethnography of a language teacher educator. *Teacher Education Quarterly*, *46*(3), 34-56.

Zhao, D. (2024). The impact of AI-enhanced natural language processing tools on writing proficiency: An analysis of language precision, content summarization, and creative writing facilitation. *Education and Information Technologies.* https://doi.org/10.1007/s10639-024-13145-5