

Build expertise first: why PhD training must sequence AI use after foundational skill development

Arjun Krishnan, Biomedical Informatics, University of Colorado Anschutz

Contact: arjun.krishnan@cuanschutz.edu | [@compbiologist](https://twitter.com/compbiologist)

Abstract

Generative AI tools have arrived in PhD training environments faster than principled frameworks for their use. The debate has polarized between enthusiasts who argue trainees must adopt AI immediately and critics who warn of fundamental damage to learning. Both miss the key question: not whether trainees should use AI, but when. The answer requires distinguishing two fundamentally different kinds of automation. Previous technologies like calculators, statistical software, and search engines automated mechanical execution while preserving the cognitive work that constitutes learning. Generative AI is categorically different: it automates reasoning, synthesis, and judgment themselves. This distinction matters enormously for training, because PhD education is not primarily about task completion but about developing capacity for independent scientific thought through sustained cognitive struggle. This creates what we term the verification paradox: trainees cannot meaningfully verify AI outputs because verification requires the domain expertise they are still developing. Using AI before that expertise exists bypasses the developmental process that builds it while producing polished outputs that mask the gap. The solution is sequencing. Trainees should build foundational expertise through deliberate, feedback-driven practice before using AI to augment it. The threshold is demonstrated independent mastery: the ability to complete tasks, explain reasoning, and catch errors without assistance. Once crossed, AI use becomes not just acceptable but genuinely productive. PhD programs, mentors, and professional societies urgently need community standards built on this developmental framework rather than ad hoc policies shaped by convenience. An implementation of this article's conceptual framework as a detailed guide with task-specific protocols is available at <https://doi.org/10.5281/zenodo.18452319>.

The problem we face

Universities, graduate program administrators, and mentors worldwide are grappling with an urgent question: How should PhD trainees use generative AI tools like ChatGPT, Claude, and Copilot? Students are already using these tools to write code, analyze data, draft manuscripts, and interpret results. Some complete entire analyses in minutes that would have taken days. Others feel pressure to keep up or risk falling behind.

The debate has polarized. Enthusiasts argue that AI is inevitable and students must learn to use it now (Pireci Sejdiu and Sejdiu 2025; Amini et al. 2025). Critics like Olivia Guest and Iris van Rooij warn that the inevitability is a self-serving claim peddled by the AI industry and worry that it undermines the very purpose of training (Guest et al. 2025; Suarez Estrada et al. 2025). Meanwhile, PhD programs lack principled frameworks to guide policy, and mentors struggle to advise trainees navigating these tools daily.

As an AI and machine learning (ML) researcher who leads a lab developing computational methods for biomedicine and as the Co-Director of multiple PhD training programs invested in developing the next generation of biomedical researchers, I've deeply considered the thoughtful integration of AI tools into research and education. My primary conclusion is in two parts: 1) Similar to that of many other experts, I too believe that generative AI is remarkable and qualitatively different from previous technologies in what it automates, and 2) This difference has profound implications for how and when it should be used during PhD training.

So, the issue isn't whether future scientists will use AI. They will. The question is whether they'll use it augmentatively as a tool they control, or dependently as a crutch they cannot verify. The answer hinges on developmental timing; specifically, whether trainees build foundational expertise before relying on AI support¹.

What makes generative AI different

Whenever concerns about AI in education arise, a familiar response is something like: "Every technology was initially resisted. The printing press, calculators, and search engines were all criticized as threats to learning." This defense sounds reasonable. It evokes progress and the folly of resisting inevitable change. But it fundamentally misrepresents what's different about generative AI.

A concrete example

Consider a PhD student analyzing gene expression data from an experiment comparing treated and control samples. In the pre-GenAI world, automation looked like this: the student decided which statistical test was appropriate for the experimental design (perhaps a paired t-test because samples came from the same subjects before and after treatment) and then used a statistical software to execute that test, graphing tools to plot the data based on their specifications, and a reference manager to format the citations that they selected. Each tool automated *mechanical* execution, *i.e.*, carrying out established procedures, while preserving the cognitive work. The student still had to understand when a paired versus unpaired test was needed, evaluate whether assumptions of the statistical test were met, interpret whether results supported the hypothesis, and determine how to visualize patterns to communicate findings effectively.

Today, that same student can ask ChatGPT or Claude: "Analyze this gene expression data and tell me what it means." AI can formulate an entire analysis approach, choose statistical tests, generate code, run analyses, create visualizations, interpret results, and draft a results paragraph. The student can complete the task without engaging in any of the cognitive work that the training is designed to develop.

The critical distinction: mechanical versus cognitive automation

This isn't a difference of degree but of kind. Previous technologies automated repetitive execution of established procedures. The printing press reproduced text that authors had created. Calculators executed arithmetic steps students already understood how to perform. Search engines retrieved information that humans then had to evaluate and synthesize. The common pattern is that these technologies freed humans for higher-order cognitive work. They didn't bypass learning; they supported it by handling mechanical overhead. Generative AI is categorically different because it automates cognitive processes themselves. It doesn't just execute your analysis plan; it can formulate the plan. It doesn't just reproduce your ideas; it can generate ideas. It doesn't just find information; it can synthesize arguments across sources. It doesn't just follow procedures; it can make evaluative judgments, interpret patterns, and draw conclusions. As Ronald Purser articulates in his critique of AI in education (Purser 2025): it "doesn't extend cognition—it automates it, turning thinking itself into a service."

Why does this matter for training specifically

This distinction has profound implications for PhD training. For established researchers who already possess deep domain knowledge and analytical capabilities, AI can genuinely augment productivity. They have the

¹ This article presents the conceptual framework and core principles; a detailed implementation guide with task-specific protocols is available as a companion resource (Krishnan 2026, [doi:10.5281/zenodo.18452319](https://doi.org/10.5281/zenodo.18452319)).

expertise to guide AI appropriately, knowing which questions to ask and which approaches make sense, and to verify outputs critically by recognizing errors and assessing reasonableness. For trainees still developing those capabilities, the same tool offers something very different: a bypass around the developmental process itself. Same technology, opposite effects, depending on where the user stands in their expertise development.

Figure 1. Automation spectrum and training impact. The impact of automation depends on both WHAT is automated and WHO uses it. Automating mechanical processes (green) typically preserves learning; automating cognitive processes during training (red) risks bypassing development of essential expertise.

		User expertise level	
		Trainee (developing expertise)	Expert (established expertise)
Type of automation	Mechanical (executing established procedures)	Examples: Statistical software running chosen tests, search engines finding the right information Effect: Enables focus on cognitive skills Training impact: Positive/Neutral (preserves learning)	Examples: Automated computational pipelines, batch processing, data management tools Effect: Increases productivity
	Cognitive (reasoning, synthesis, judgment)	Examples: GenAI writing analysis code, GenAI interpreting results, GenAI synthesizing literature Effect: Bypasses skill development Training impact: NEGATIVE (prevents expertise development)	Examples: GenAI for exploring variations, AI-assisted iteration, augmented workflows Effect: May augment capability

This is why the question of developmental timing is critical, and why thoughtful integration requires understanding not just what AI can do, but what PhD training is actually trying to accomplish.

The training period has different goals

PhD training has a specific purpose that distinguishes it from both undergraduate education and professional practice: developing independent scientific experts who can navigate uncharted intellectual territory. This requires building cognitive capabilities that are fundamentally non-mechanical.

What does expertise actually look like

The goal of PhD isn't task completion, content memorization, or even productivity during the training years. It's developing the ability to formulate novel research questions from messy, ambiguous observations. To navigate abstract problems logically without algorithmic procedures to follow. To make context-dependent judgments when no clear rubric exists. To synthesize disparate information into coherent understanding. To recognize flaws in complex reasoning, both one's own and others'. To generate creative solutions to problems that don't have established answers. To develop intuition about what "correct" looks like in your domain.

These capabilities are non-mechanical precisely because no cookbook recipe exists for deploying them. Multiple valid approaches often exist for any given problem. Success requires interpretation, judgment, and adaptation to context. They demand integration across different types of knowledge rather than algorithmic application of fixed procedures.

How does expertise actually develop

These capabilities don't emerge from reading about them or watching demonstrations. They develop through sustained cognitive struggle over hundreds of hours. You build them by pushing beyond current abilities, falling short and making mistakes, and catching them yourself or with guidance from a mentor. By grappling with difficult concepts until understanding clicks into place. By developing pattern recognition through encountering variations of problems repeatedly. By wrestling with how to express complex ideas clearly. By building intuition through accumulated experience of what works and what doesn't. By learning to evaluate your own work critically before showing it to others. The "inefficiency" of this struggle is a feature, not a bug; it's the very mechanism through which learning happens. The difficulty is the point and there are no shortcuts to expertise.

The problem with cognitive automation during expertise development

When AI offers to automate the cognitive processes that constitute learning, be it formulating analytical approaches, making interpretive judgments, or synthesizing information, it offers to bypass the struggle that builds expertise and enables learning to formulate good questions, operate under ambiguity, adapt methods creatively, interpret failure constructively, and take ownership of intellectual work. It's like training navigators exclusively with GPS. They never learn to read terrain, understand their position relative to landmarks, or create routes through unmapped territory. When GPS fails or doesn't cover their destination, as inevitably happens when charting genuinely new territory, they lack the fundamental skills that independent navigation requires. This isn't hypothetical. Other professions requiring expert judgment in high-stakes situations insist on training without automation even when that automation exists and is reliable. Ship captains still learn celestial navigation despite GPS. Pilots train extensively in manual flying despite sophisticated autopilot systems. Surgeons practice basic surgical techniques extensively before touching robotic systems. Why? Because when systems fail, when edge cases arise that automated tools weren't designed for, when truly novel situations emerge, foundational expertise becomes essential. The automation makes experts more effective, but only because they already possess the expertise to use it appropriately and recognize when it's failing.

Another major problem is that, with AI, trainees can produce outputs that look sophisticated, but this sophistication creates a dangerous illusion: it instills a sense of productivity because tasks get completed and outputs get generated. The work will look competent because code will run, analyses will appear professional, and results will seem reasonable. But it masks a fundamental lack of understanding that emerges later when independent work is required, when novel problems arise where AI fails, when job interviews probe actual capabilities, or when collaborators or reviewers discover errors. The two students in **Box 1** both produce impressive outputs during training. But the critical difference emerges only when independent expertise becomes necessary. By that point, years of critical skill development may be irretrievably lost. The student who built expertise first can now use AI to work faster; the student who depended on AI lacks the foundation to work independently. This raises an obvious question: can't students simply verify AI outputs to ensure quality while maintaining efficiency? This brings us to a fundamental paradox.

Box 1. Two paths through training; and why the 'learn AI now' argument misses the point

	Month 1-3	Month 4-6	Month 7-12	Outcome
Student A	Struggles w/ choosing statistical tests for experimental design	Independently chooses & runs appropriate tests	Uses AI to speed up routine tasks	Ready for independent research career
Student B	Uses ChatGPT to analyze experimental data	Continues relying on AI for analyses	Dependency deepens	Struggles in independent research settings

A common refrain in academia and industry is: **“Students will be expected to use AI for professional work, so they should learn to use it now.”** The first part is correct; future scientists will indeed use AI extensively. But the conclusion does not follow.

The argument conflates two distinct things: using AI tools and using them effectively. Effective AI use in research requires the ability to guide the tool toward appropriate approaches, recognize when its outputs are wrong, and work independently when it fails. These capabilities are built through the same foundational development that AI use during training would bypass. A trainee who reaches independence having outsourced cognitive work to AI has learned neither the domain expertise nor the verification skills required to wield the tool responsibly.

The timing argument also misunderstands the nature of AI tool adoption. Unlike domain expertise, which takes years of deliberate practice to build, proficiency with AI interfaces is rapid. A researcher with strong analytical foundations can achieve productive fluency with a new AI tool within days. What cannot be compressed is the development of the expertise that makes that fluency valuable. The student who builds foundational skills first and adopts AI tools later loses nothing competitively and gains the foundations that make AI use effective rather than dependent.

Finally, the “learn it now” argument implicitly assumes that AI tools are stable targets worth investing training time in. But the specific tools will change rapidly; the reasoning and verification skills that allow effective use of any tool will not. Building the latter is the durable investment.

The machinal bypass trap

Deanna Kaplan, Roman Palitsky, and Charles Raison identify this pattern as ‘machinal bypass’, *i.e.*, using AI not to support human thinking but to sidestep it altogether (Kaplan et al. 2025). The phenomenon describes a recognizable vulnerability: “when we feel stressed, overwhelmed, or doubtful of our capabilities, we look for manageable fixes that help avoid these uncomfortable feelings”. For PhD students facing sustained difficulty, AI offers exactly such a fix. Rather than deliberating through a scientific problem with incomplete data and imperfect explanations, students can ask “a language model to fill in the blanks with a logical conclusion”. The work gets done, the discomfort gets avoided, but the developmental process gets bypassed entirely.

The verification paradox

The standard response to concerns about AI dependence sounds reassuringly simple: “Just verify the output, like you would verify any collaborator’s work.” This seems reasonable. After all, science requires verification regardless of source. What makes AI different? The answer reveals a fundamental paradox at the heart of AI use during PhD training: verification requires the very expertise trainees are trying to develop.

Joe Cheng and Sara Altman capture this problem precisely (Cheng and Altman 2025): AI tools for complex tasks “will make mistakes” but are “good enough...to lull you into a false sense of security, making you think you don’t need to review the code it generates.” Their analogy is apt: AI is a set of fins that can make you swim faster and stronger, not a life vest. It will not keep you from drowning if you don’t already know how to swim. For PhD students still learning to swim—still developing the background knowledge, analytical expertise, and technical skills required to verify complex work—AI doesn’t augment capability; it creates dependence.

The circular dependency

Here’s the problem: you cannot verify what you cannot do yourself. You can’t spot errors you haven’t learned to recognize. You don’t know what “correct” should look like in complex scenarios without prior experience. You lack the pattern recognition to notice subtle problems. You’re missing the domain knowledge to catch context-specific mistakes. You haven’t developed the intuition about reasonableness that comes from doing

similar analyses many times. These issues create a circular trap: using AI before you can verify it means training yourself to produce unverified work.

Recent empirical work from Anthropic (the company behind Claude) bears this out directly. Judy Shen and Alex Tamkin found that AI use impaired conceptual understanding, code reading, and debugging abilities, with the largest skill gaps emerging in debugging (Shen and Tamkin 2026)—precisely the capability that requires understanding when and why code fails. Notably, AI use didn’t uniformly harm learning: participants who used AI to build comprehension—asking follow-up questions, requesting explanations, posing conceptual questions while coding independently—retained more than those who delegated coding to AI. This distinction maps cleanly onto our framework: the former reflects AI use after or alongside developing understanding; the latter bypasses it entirely.

Why is verification particularly difficult with AI

The challenge has both practical and expertise dimensions. Practically, AI generates output faster than humans can carefully review it. When fifty lines of analysis code appear in a few seconds, thoroughness feels like inefficiency. The speed creates psychological pressure to move on rather than verify line-by-line. The volume of output makes comprehensive review exhausting. And the polished appearance—professional formatting, confident explanations, code that runs without errors—suggests correctness even when fundamental problems lurk beneath the surface. Faced with this issue, professionals across industries deal with AI-generated “workslop” (Niederhoffer et al. 2025) either by completely eliminating evaluation or by doing extra work finding and correcting errors. Naturally, this issue is deeper and starker for PhD students in training. They genuinely cannot perform the verification that AI outputs require. Consider what verification actually demands for that gene expression analysis discussed earlier.

A concrete verification scenario

AI has generated a complete statistical analysis: code, tests, visualizations, and interpretation. To verify this, the trainee must answer several questions: *Is this statistical test appropriate for the experimental design?* This requires understanding the difference between paired and unpaired tests, when parametric versus non-parametric approaches apply, and how the experimental structure (same subjects measured twice versus different subjects in each group) determines which test is valid. *Are the statistical assumptions met?* This requires knowing what assumptions (of normality, equal variances, and independence) underlie the chosen test and how to evaluate whether the data satisfy them. *Is multiple testing correction appropriate?* This requires understanding when and why corrections like Bonferroni or FDR apply, and whether the correction method matches the scientific question. *Does the interpretation match the actual results?* This requires the ability to read statistical output correctly, distinguish between statistical significance and biological importance, and recognize when conclusions overreach what the data support. *Are there confounding variables unaccounted for?* This requires knowledge of the experimental system and the ability to think critically about what else might explain the observed patterns. *Is the biological interpretation reasonable?* This requires domain expertise to evaluate whether the proposed mechanism makes sense given what’s known about the system.

Table 1. AI errors that require domain expertise to catch. Each of these errors appears in polished, professional-looking output with confident explanations. Detection requires the domain expertise that trainees are developing through PhD training. Using AI before developing this expertise means training yourself to produce unverified work.

Error category	Specific example	Why trainees can't catch it	Consequences
Hallucinated citations	AI cites "Smith et al. 2023, Nature" which doesn't exist, or misrepresents what paper actually shows	Trainees haven't read all literature; assume AI is accurate; don't verify every citation in original sources	False information enters your work; reviewers catch it; credibility damaged; retraction risk
Inappropriate statistical methods	AI suggests t-test for data requiring paired test, or uses parametric test when assumptions violated	Requires understanding experimental design, test assumptions, when each test applies. This is expertise that's being developed	Wrong conclusions; failed replications; reviewer rejection; wasted experimental effort
Logical code errors	Code runs without errors but filters data incorrectly (e.g., ">" instead of ">=", wrong column name)	Requires knowing what correct output should look like; careful manual checking; experience with edge cases	Subtle errors in results; conclusions based on wrong data; discovered only if results seem obviously odd
Confirmation bias	AI chooses analysis approach that supports expected hypothesis; ignores contradictory evidence	Requires scientific skepticism, awareness of multiple approaches, willingness to question convenient results	False discoveries; publication bias; contributes to reproducibility crisis
Missing domain caveats	AI interpretation ignores important technical limitations (detection limits, saturation, batch effects)	Requires deep knowledge of measurement approaches, experimental systems, technical constraints of methods	Overstated conclusions; misinterpretation of biology; misleading claims
Misrepresented statistics	AI reports statistical results it didn't actually calculate, or misinterprets p-values, confidence intervals	Requires understanding what statistics mean, ability to verify calculations independently	False confidence in results; incorrect interpretation; statistical errors in publication
Wrong baseline comparisons	AI compares to inappropriate control group or ignores experimental blocking factors	Requires understanding experimental design, what constitutes proper control, how blocking works	Wrong conclusions about treatment effects; confounded results

Cycles within cycles

Each of these verification steps requires specific expertise that PhD training is designed to build. The trainee early in their PhD cannot perform this verification because they're still developing the knowledge required to do so. The circular problem is inescapable and can spawn more vicious cycles. As identified by a Microsoft paper about knowledge work (Lee et al. 2025), the low self-confidence that is likely to stem from limited skills and knowledge may lead students to rely more heavily on AI, potentially diminishing their critical engagement and independent problem-solving. On the other hand, the rapid completion of tasks and production of polished outputs could lead to an illusion of competence, which might crumble suddenly and painfully when the graduated trainee faces professional situations like novel problems and unseen failures that expose the missing foundation. The consequences extend beyond skill gaps. Suqing Wu and colleagues (Wu et al. 2025) found that transitioning from GenAI collaboration to independent work led to significant decreases in intrinsic motivation and increases in boredom, even as participants reported a greater sense of control. The implication for PhD training is sobering: students who collaborate with AI during foundational tasks may not only lack the skills for independent work; they may also lose the motivation to do it.

The asymmetry is unavoidable

This asymmetry is not about the technology itself but about developmental stage. The same tool augments capability in those who already possess it and undermines development in those still building it. Which brings us to a solution.

A developmental framework for sequencing AI use thoughtfully

The verification paradox has a solution, but it requires recognizing that the question isn't whether to use AI; it's when and how. The answer lies in sequencing: build expertise first, then use AI to augment that expertise.

The core principle: Expertise before augmentation

The framework is straightforward in concept, though it demands discipline in practice. During foundational skill development, trainees should complete cognitive work without AI assistance; not as a fixed number of trials, but until independent mastery is demonstrated. Research on deliberate practice and cognitive load makes clear that what matters is the quality of engagement: structured struggle, feedback, and iteration, not the accumulation of attempts (Ericsson and Harwell 2019). The threshold is functional, not numerical: can the trainee complete the task successfully without assistance, explain their reasoning clearly, and catch their own errors? Until that bar is reached, the developmental process requires protection from bypass.

During this phase, AI use should be strictly instructive. After genuinely attempting to understand a concept and reaching an impasse, a trainee might use AI dialogue to clarify a specific confusion. After wrestling with a problem, they might discuss their approach to identify flaws in reasoning. The key distinction is that AI serves as a Socratic interlocutor probing understanding, not as an oracle delivering answers.

Once independent mastery is established, AI use becomes not just appropriate but genuinely productive. At this stage, AI can accelerate iteration and exploration, serving as a critique and verification aid rather than a primary solution generator. The trainee's expertise is what makes this productive: they can guide AI toward appropriate approaches, recognize when outputs are wrong, and work independently when AI fails or proves inadequate for novel problems. This is the difference between augmentation and dependence.

Why does this sequencing work

This approach develops verification expertise before creating dependence on AI. You build the pattern recognition and domain knowledge required to spot errors. You develop the judgment needed to evaluate whether approaches make sense. You gain the experience to know what 'reasonable' looks like. This developmental process, as Sebastian Raschka notes (Raschka 2025), is inherently cyclical: doing manually, studying expert work, getting feedback, and iterating. Empirical evidence from a study of AI impact on coding skill formation supports this sequencing, showing that *how* participants used AI, not simply *whether* they used it, determined learning outcomes (Shen and Tamkin 2026). Those who used AI to support active comprehension retained knowledge; those who delegated tasks to AI did not. The interaction patterns that harmed learning most (full delegation, progressive reliance, iterative AI debugging) are precisely those our framework cautions against during foundational skill development. AI can make this cycle faster and more expansive when you use it as a partner and AI augmentation becomes safe and productive. You can verify what it produces, guide it toward better approaches when it goes astray, and work independently when it fails or doesn't exist for your particular problem. The result: scientists who wield AI as a tool they control rather than

depend on. Scientists who can work on novel problems where AI hasn't been trained. Scientists who can evaluate others' AI-assisted work because they possess independent expertise.

Box 2. Productive AI uses during training after building foundational skills

1. PRACTICING PROFESSIONAL COMMUNICATION

Presentation practice:

- After creating presentation content yourself, use AI voice mode to simulate Q&A
- Example prompt: "You are a faculty member in [field]. I'll present my research plan. Ask challenging questions about rationale, methodology, and limitations. One question at a time."
- Why this works: Content is yours; AI provides iteration practice that's impractical with busy mentors
- Skill developed: Articulation, handling challenges, thinking on your feet

Comprehensive exam preparation:

- After mastering exam material, use voice mode to practice answering questions
- Example: "You're my committee member with expertise in [area]. Test my understanding with progressively difficult questions. After each answer, critique my clarity and suggest improvements."
- Why this works: Knowledge is yours; AI simulates high-stakes interaction for practice
- Skill developed: Explaining complex concepts, defending ideas, handling pressure

Networking/collaboration preparation:

- After understanding your work deeply, practice discussing it with AI role-playing as potential collaborators
- Example: "You're Dr. [X] whose work on [Y] I'd like to discuss. Respond as they might, ask about my work, and afterward give feedback on how I could be more concise."
- Why this works: Builds confidence in professional interactions through safe practice
- Skill developed: Concise communication, finding collaboration angles

2. INSTRUCTIVE DIALOGUE

Socratic questioning:

- After attempting analysis yourself, engage in dialogue to test understanding
- Example: "I chose a paired t-test for this before/after design. Ask me questions to probe whether I truly understand when and why this test is appropriate."
- Why this works: You've already thought through the problem; AI helps identify gaps
- Skill developed: Deep understanding, recognizing knowledge gaps

Debugging conceptual understanding:

- After getting stuck on a concept, discuss your confusion after trying to resolve it yourself
- Example: "I've read about multiple testing correction but I'm confused about when to apply Bonferroni vs. FDR. Here's what I understand [explain]. What am I missing?"
- Why this works: You've engaged with the material first; AI helps clarify specific confusions
- Skill developed: Conceptual clarity, asking good questions

Code review dialogue:

- After writing code yourself, walk through it with AI to identify issues
- Example: "I'll explain my code line by line. Point out logical errors, inefficiencies, or places where my reasoning is unclear."
- Why this works: You wrote the code; AI helps you think through edge cases and logic
- Skill developed: Code review skills, debugging reasoning

3. TECHNICAL ASSISTANCE

Installation/configuration help:

- Troubleshooting environment setup, package compatibility, system configuration
- Why this works: Purely mechanical; doesn't bypass learning the actual analysis
- Example: "I'm getting this error installing [package]. What system dependencies might I be missing?"

Syntax learning:

- Learning new programming syntax for operations you already understand conceptually
- Why this works: You know what you want to do; AI shows how in new language
- Example: "I want to filter this dataframe for rows where column A > 5 AND column B is 'treated'. How do I write this in pandas?"

Error message interpretation:

- Understanding cryptic error messages after reading them carefully yourself
- Why this works: Helps decode technical jargon without bypassing debugging process
- Example: “I’m getting this error [paste]. I’ve checked my file path and data types. What else could cause this?”

4. ITERATION & EXPLORATION**Generating variations:**

- After creating initial version, explore alternatives quickly
- Example: After writing clear paragraph, ask “Suggest 3 alternative ways to phrase this key sentence while maintaining accuracy”
- Why this works: Core content is yours; AI helps explore expression options
- Skill developed: Recognizing multiple valid approaches, stylistic range

Critique and refinement:

- After completing draft, get structured feedback
- Example: “Critique this *Methods* paragraph for clarity, completeness, and reproducibility. Where would a reader be confused?”
- Why this works: You created content; AI provides additional perspective
- Skill developed: Self-editing, anticipating reader needs

Productive uses share key features

Notice what the productive uses in **Box 2** have in common. They’re all deployed after substantive intellectual work has been completed independently. The presentation content is fully developed before using AI to practice Q&A. The code is written before using AI to review logic. The concepts are studied before using AI to test understanding. These uses support iteration and practice rather than initial generation. They require critical evaluation of AI feedback rather than blind acceptance. They augment human learning opportunities (like getting feedback from a busy mentor) rather than replacing cognitive work. They build on developing expertise rather than bypassing it. Detailed, task-specific implementation protocols for computational data analysis, literature review and synthesis, manuscript and proposal writing, figure generation, and communication practice are available in a comprehensive implementation guide (Krishnan 2026, [doi:10.5281/zenodo.18452319](https://doi.org/10.5281/zenodo.18452319)).

Accountability and transparency

Thoughtful integration requires transparency, which is fundamental to scientific accountability and reproducibility. So, when you use AI tools, document the specifics like you would when using any other scientific tool. Note the name and version, date of use, prompts you provided, outputs generated, and crucially, your verification steps. Track errors you found and how you corrected them. This record protects you by enabling reproducibility and helping you learn from patterns, not by surveilling your process. Save interaction records so others could reproduce your work. Discuss AI use with mentors and collaborators on shared projects so they understand what tools contributed to shared outputs. In addition to enabling reproducibility, this documentation, when reviewed, helps you identify patterns in when AI helps versus hinders. And it protects you. If questions arise about your work, clear documentation of what you did and how you verified it demonstrates scientific rigor.

Not paternalism, but pedagogy

Some might view these guidelines as paternalistic restrictions. They’re not. Training already involves extensive scaffolding and appropriate sequencing of challenges. We require prerequisite courses before allowing students into advanced ones because foundational knowledge enables productive engagement with complex

material. We insist on preliminary and comprehensive examinations before dissertation research because demonstrating breadth and depth of knowledge ensures students can contextualize their work appropriately. We expect multiple revision rounds with mentor feedback before manuscript submission because scientific writing develops through iterative improvement. These aren't arbitrary barriers. They're pedagogical structures that support development of expertise. The AI use framework fits this same pattern: it sequences tool use to align with capability development, ensuring students build the expertise required to use powerful tools effectively rather than dependently. The goal is clear: train scientists who can effectively wield AI because they possess the expertise to guide and verify it, not scientists who depend on AI because they lack that foundational knowledge².

The need for community standards

Expecting individual labs to develop ad hoc policies isn't scalable and is unlikely to result in sufficiently incisive and comprehensive recommendations³. PhD programs should develop explicit guidelines for AI use, tailored to developmental stages within training. These policies should focus on expertise development rather than blanket restrictions. They should distinguish between uses that support learning and uses that bypass it. They should be updated as research and empirical evidence about AI's impact on learning accumulates. And importantly, they should be shared across institutions to establish community standards. We need field-wide discussion about PhD (and medical) training in the AI era. Professional societies should convene working groups. Funding agencies should consider whether training grants require AI use policies. Journals might request disclosure of AI use in methods training just as they now request disclosure in manuscript preparation. The research community should collaborate on this challenge just as it has collaborated on other methodological and ethical issues.

The stakes for scientific expertise

AI in research is inevitable. Tools will only become more powerful and more integrated into scientific workflows. That's precisely why thoughtful training matters so much right now. The question before us isn't whether future scientists will use AI extensively. They will. The question is whether they'll use it effectively as experts wielding powerful tools, or dependently as novices relying on systems they cannot fully evaluate. Effective AI use requires expertise to guide it appropriately, which entails knowing which questions to ask, which approaches make sense, and when outputs seem suspicious. It requires knowledge to verify outputs critically by recognizing errors, assessing reasonableness, and catching subtle problems. It requires the ability to work independently when AI fails, when it doesn't exist for novel problems, when systems go down or produce nonsensical results. Trainees cannot develop these capabilities by outsourcing cognitive work to AI during the very period when expertise development should be the primary goal.

² This approach also respects trainees as adults capable of making informed choices. As a co-director of graduate training programs, I believe in giving students comprehensive resources—infrastructure, mentorship, transparent reasoning about pedagogical choices—and then trusting them to lead their own PhD journey. Some students, fully informed about the risks of pervasive AI use during foundational skill development, will choose to use it anyway. That's their right. The goal of training programs isn't to prevent all suboptimal choices, but to ensure students understand the developmental consequences of their decisions and to maintain standards around what is ethical, legal, and policy-compliant.

³ To be clear, individual labs will, and should, make choices that fit their specific contexts and values. In my own group, I've implemented specific policies because I'm accountable for our collective output. But the framework I've outlined here is meant as a foundation for community discussion, not a mandate.

What's at stake for individuals

For trainees, the choice between building expertise first versus relying on AI early has career-defining implications. It's the difference between developing distinctive capabilities that make you valuable as a collaborator and employee, versus generic skills that AI itself provides. Between tackling novel problems that advance your field, versus limiting yourself to problems AI has been trained to solve. Between developing a scientific identity and voice that reflects deep understanding, versus producing outputs that sound authoritative but lack foundation. These differences may not be apparent during training when supportive environments and AI availability mask gaps. They become starkly visible when trainees enter independent careers and face expectations of autonomous expertise. "AI won't replace humans; humans who use AI will replace humans who do not"; so goes the common slogan, but it needs revision. It's humans with deep expertise and judgment who can direct, critique, and complement AI that will replace humans who can't.

What's at stake for science

The implications extend to the scientific enterprise broadly. We risk worsening the reproducibility crisis through unverifiable analyses. When neither authors nor reviewers can fully verify AI-assisted work, errors propagate unchecked through the literature. We risk degrading review quality. As peer reviewers face the same verification challenges that trainees do, subtle errors slip through. We risk shifting scientific culture from transparent demonstration of reasoning ("show your work and justify your choices") to opaque acceptance of polished outputs ("accept what's shown and move on"). Most fundamentally, we risk producing a generation of researchers who can produce sophisticated outputs they cannot fully verify, evaluate, or extend independently. Who can generate analyses but cannot assess their appropriateness. Who can write code that runs but cannot debug when it fails. Who can cite literature that AI has summarized but cannot evaluate the original arguments critically.

Fork in the road

We stand at a decision point. One path leads to training scientists who wield AI as a tool they control, using it to work faster and explore more thoroughly once they've built foundational expertise. The other path leads to producing researchers dependent on AI they cannot fully evaluate, capable of impressive outputs but lacking the independent expertise required when tools fail, limitations arise, or novel problems emerge. The difference between these outcomes hinges on a deceptively simple principle: whether we allow and encourage trainees to develop expertise first before becoming dependent on augmentation.

What makes this choice difficult is at the core of the appeal of machinal bypass (Kaplan et al. 2025): scientific work involves uncomfortable vulnerability. Not knowing feels stressful. Creating something new risks rejection. Wrestling with incomplete information and imperfect understanding requires authentic emotional investment. AI offers escape from all of this by providing confident answers instead of uncertainty, polished outputs instead of vulnerable creation. But the discomfort isn't a flaw in the training system to be optimized away. It's the mechanism through which scientific capability and identity develop. With, the discomfort AI eliminates isn't incidental to engagement; it's what sustains it. Along with evidence that AI collaboration might erode intrinsic motivation (Wu et al. 2025), the question is whether we preserve this necessary struggle or allow AI to bypass it.

To be precise, the question isn't "Should we use AI in research?" The question is "Should we automate the cognitive work and motivation that constitutes learning?" For the training period specifically, when expertise development must be the priority, the answer must be no; not from resistance to technology, but from commitment to what produces scientists capable of advancing it.

What this requires is tractable. PhD programs and institutions can articulate explicit, staged AI use policies that tie tool access to demonstrated competency rather than imposing blanket restrictions or granting uncritical, wholesale access (such as via ChatGPT EDU). For e.g., training programs can build the “expertise before augmentation” principle into individual development plans, comprehensive exam design, and rotation expectations. Professional societies can convene working groups to develop field-specific standards before institutional inertia settles on defaults that prioritize convenience over development. Funding agencies supporting training grants are well-positioned to require explicit AI use policies as a condition of award. The framework outlined here, and the detailed practical guide accompanying it (Krishnan 2026, [doi:10.5281/zenodo.18452319](https://doi.org/10.5281/zenodo.18452319)), aims to provide the conceptual and practical foundation for these conversations.

The same logic extends across STEM PhD training wherever independent analytical judgment is the goal. The verification paradox and expertise asymmetry described here are not unique to biomedicine, though their discipline-specific expression varies. Educators in related contexts, including undergraduate research experiences and research-focused master’s programs, should find the core principles transferable and worth adapting.

Some inefficiencies are essential. The difficulty is the mechanism. The struggle is the point. Only by recognizing this can we train scientists who will use AI effectively rather than dependently; scientists who will drive the next generation of discovery rather than being limited by the patterns AI has already learned.

Acknowledgements

I’m grateful to Chad Myers for sharing his lab’s AI use principles in mid-2024 and instigating me to write one for my group, which ultimately provoked a lot of thinking and reading about this topic. Many thanks to Stephen Turner for reading a version of this manuscript and providing insightful notes, which helped improve this article. This work was supported by NSF 2328140 grant to AK.

AI use statement

This manuscript (and the supplementary guidelines) was developed with assistance from Claude (Anthropic) in ways that align with the expertise-before-augmentation framework the article advocates. The core argument, conceptual framework, empirical evidence selection, and scientific conclusions originated from the author based on his experience in research training and synthesis of published studies and articles. AI was used strictly for iterative refinement after substantive intellectual work was completed independently. The author maintains full responsibility for all intellectual contributions and any errors. For further transparency and training purposes, **Box 3** provides more details (and examples) of AI use, the distinction between augmentative use (by someone with established expertise) and dependent use (as a substitute for expertise development), and the verification process.

Box 3. Disclosure of AI use: specific example uses, critical distinctions, and verification process

SOME SPECIFIC USES
<div><div>1. Structural and editorial feedback (after complete drafts were written)</div><div><ul style="list-style-type: none">Identifying repetitive sections and suggesting consolidationsFlagging tonal inconsistencies and proposing revisionsRecommending tighter phrasing for overly verbose passages</div><div><i>Example prompt:</i> “I’ve written a full draft of this article. Analyze it for repetition, identify where the argument could be tightened, and suggest specific cuts while preserving the core conceptual framework.”</div></div>

Example prompt: “Flag the draft for academic/jargon’y language that might be a barrier for a broader audience (scientists, educators, students, interested public)”

Example prompt: “I want to make sure that the following points come through in the article: [three-part response to proficiency refrain]. What is the best place to integrate them?”

Author decision: The author rejected AI’s suggestion to create a standalone section early in the article, and instead chose to integrate these points into **Box 1**, recognizing this would maintain better narrative flow while still ensuring the arguments were present.

AI suggestion (as part of a proposed revision): Add the statement ‘Professional scientists often navigate using established maps and validated tools’ under ‘Training to map uncharted terrain’. Also potentially remove this whole section as it is likely weak.

Author decision: Retained the section and substantially rewrote based on his experience-based detailed view on the PhD→Professional scientist shift that better captures the developmental progression while maintaining the navigational metaphor.

AI contribution: As part of condensing the text, Claude suggested converting paragraphs about automation and student narrative into more easily digestible display items, and designed their content and structure (*Figure 1: Automation matrix* and *Box 1: Two students narrative*). Also, upon prompting to create miniature versions of the common AI errors and productive use cases listed in the supplementary *Practical Guide*, Claude created *Table 1: AI errors* and *Box 2: Productive uses*. These were then edited and refined by the author.

2. Title and framing development (after core argument was established)

- Generating alternative title options that make the central thesis explicit
- Suggesting abstract teaser language that captures the main argument concisely
- Gaming out different framings: chronological, conceptual-first, narrative-based, paradox-centered, and provocative

Example prompt: “The current title is descriptive but passive. Suggest active titles that present the opinion I’m putting forward rather than just naming the topic.”

AI suggestion: Conceptual-first framing because it leads with the most powerful conceptual contribution.

Author decision: Chose chronological/historical framing because of alignment with the practical guide, which was written first, and because what felt more natural.

3. Citation integration strategy (after reference list was finalized)

- Proposing more optimal locations within the manuscript structure for incorporating empirical citations
- Drafting integration language that weaves references naturally into existing prose

Example prompt: “Here are my notes from three original sources that I’m citing in this article. Give me integration ideas for each, along with succinct text based on my notes.”

4. Social media dissemination (after manuscript was finalized)

- Drafting Bluesky thread and LinkedIn post following established format and style from previous posts by the author
- Adapting technical content for broader audiences

Example prompt: “I’ve attached a previous Twitter thread I wrote for a recent paper. Follow that format and style to create posts for this new work on both Bluesky and LinkedIn.”

CRITICAL DISTINCTIONS IN HOW AI WAS USED

- **AI did not generate the verification paradox concept, the mechanical vs. cognitive automation framework, the sequencing principle, or any other core intellectual contribution.** These originated from the author’s analysis of the problem, reading of the empirical literature, and experience in PhD training.
- **AI did not select which empirical studies to cite or determine their relevance.** The author independently identified all references based on extensive reading about relevant topics.
- **The author exercised independent editorial judgment throughout.** Multiple AI suggestions were rejected when they weakened the argument, introduced unnecessary complexity, or didn’t match the intended voice.
- **All claims, interpretations, and recommendations were verified independently by the author.** All AI outputs were treated as draft material requiring verification and revision, not authoritative content.

VERIFICATION PROCESS

Every AI-generated suggestion was extensively evaluated for: (1) factual accuracy by checking against the author's knowledge of primary sources, (2) logical coherence with the overall argument, (3) appropriateness of tone and register for the target venue, and (4) consistency with the author's intended meaning. Suggestions were revised, rejected, or substantially modified based on this evaluation.

References

- Amini, Lisa, Henry F. Korth, Nita Patel, Evan Peck, and Ben Zorn. 2025. "Empowering the Future Workforce: Prioritizing Education for the AI-Accelerated Job Market." arXiv:2503.09613. Preprint, arXiv, March 3. <https://doi.org/10.48550/arXiv.2503.09613>.
- Cheng, Joe, and Sara Altman. 2025. "Databot Is Not a Flotation Device." *Posit*, August 28. <https://posit.co/blog/databot-is-not-a-flotation-device/>.
- Ericsson, K. Anders, and Kyle W. Harwell. 2019. "Deliberate Practice and Proposed Limits on the Effects of Practice on the Acquisition of Expert Performance: Why the Original Definition Matters and Recommendations for Future Research." *Frontiers in Psychology* 10 (October). <https://doi.org/10.3389/fpsyg.2019.02396>.
- Guest, Olivia, Marcela Suarez, Barbara Müller, et al. 2025. "Against the Uncritical Adoption of 'AI' Technologies in Academia." Preprint, Zenodo, September 5. <https://doi.org/10.5281/ZENODO.17065099>.
- Kaplan, Deanna M., Roman Palitsky, and Charles L. Raison. 2025. "The 'Machinal Bypass' and How We're Using AI to Avoid Ourselves." *Proceedings of the National Academy of Sciences* 122 (51): e2518999122. <https://doi.org/10.1073/pnas.2518999122>.
- Lee, Hao-Ping (Hank), Advait Sarkar, Lev Tankelevitch, et al. 2025. "The Impact of Generative AI on Critical Thinking: Self-Reported Reductions in Cognitive Effort and Confidence Effects From a Survey of Knowledge Workers." *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA), CHI '25, April 25, 1–22. <https://doi.org/10.1145/3706598.3713778>.
- Niederhoffer, Kate, Gabriella Rosen Kellerman, Angela Lee, Alex Liebscher, Kristina Rapuano, and Jeffrey T. Hancock. 2025. "AI-Generated 'Workslop' Is Destroying Productivity." *Generative AI. Harvard Business Review*, September 22. <https://hbr.org/2025/09/ai-generated-workslop-is-destroying-productivity>.
- Pireci Sejdiu, Nora, and Sejdi Sejdiu. 2025. "The Quiet Transformation of Higher Education in the AI Era." *Open Research Europe* 5 (August): 249. <https://doi.org/10.12688/openreseurope.20715.1>.
- Purser, Ronald. 2025. "AI Is Destroying the University and Learning Itself." *Current Affairs*, December 1. <https://www.currentaffairs.org/news/ai-is-destroying-the-university-and-learning-itself>.
- Raschka, Sebastian. 2025. "The State Of LLMs 2025: Progress, Problems, and Predictions." Sebastian Raschka, PhD, December 30. <https://magazine.sebastianraschka.com/p/state-of-llms-2025>.
- Shen, Judy Hanwen, and Alex Tamkin. 2026. "How AI Impacts Skill Formation." arXiv:2601.20245. Preprint, arXiv, January 28. <https://doi.org/10.48550/arXiv.2601.20245>.
- Suarez Estrada, Marcela, Müller Barbara, Olivia Guest, and Iris van Rooij. 2025. *Critical AI Literacy: Beyond Hegemonic Perspectives on Sustainability*. June 12. <https://doi.org/10.5281/ZENODO.15677840>.
- Wu, Suqing, Yukun Liu, Mengqi Ruan, Siyu Chen, and Xiao-Yun Xie. 2025. "Human-Generative AI Collaboration Enhances Task Performance but Undermines Human's Intrinsic Motivation." *Scientific Reports* 15 (1): 15105. <https://doi.org/10.1038/s41598-025-98385-2>.