

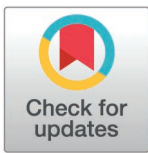
EDUCATION

Ten simple rules for effective use of generative AI for code development in environmental science

Rachel A. King^{1*}, Laurel Abowd¹, Carlo W. Broderick¹, LM Bradley¹, Max F. Czapanskiy^{2,3}, Mona M. Farnisa¹, Erica M. Ferrer¹, Carmen Galaz García^{1,2}, Darian Gill¹, Nicole M. Greco¹, Juliette Jacquemont¹, Justin A. Kadi¹, Li Kui^{1,4}, Gretchen LeBuhn^{1,5}, Liying Li¹, Abigail Meyer¹, Marisa Morse¹, Evan Patrick², Samantha Shanny-Csik^{1,2}, Nicholas A. C. Tucker¹, Zhe Wang¹, Caitlin R. Fong¹

1 National Center for Ecological Analysis and Synthesis, University of California Santa Barbara, Santa Barbara, California, United States of America, **2** Bren School of Environmental Science and Management, University of California Santa Barbara, Santa Barbara, California, United States of America, **3** College of Creative Studies, University of California Santa Barbara, Santa Barbara, California, United States of America, **4** Marine Science Institute, University of California Santa Barbara, Santa Barbara, California, United States of America, **5** Department of Biology, San Francisco State University, San Francisco, California, United States of America

* rachel.king00@gmail.com



Introduction

The explosion of generative artificial intelligence (GenAI, [Table 1](#)) into scientific disciplines is rapidly transforming how scientists conduct their research. The potential of GenAI to accelerate time consuming parts of the scientific process has led to its incorporation into nearly every aspect of research, from planning experiments to analyzing data and writing papers [1–4]. The benefits of using GenAI range from increasing productivity [5] to reducing language barriers [6] to scoping research directions [7]. However, emerging discrepancies in GenAI use across disciplines, age, career stage, demographics, and human development status [4,8] have led to concerns that instead of broadening participation, GenAI will exacerbate existing inequities in science [9]. Further concerns about the rigor and reproducibility of GenAI [10] have made some researchers skeptical about the effectiveness of using GenAI in scientific research. To avoid entrenching existing barriers and to ensure that researchers are using tools in ways that align with best research practices and ethical considerations, scientific communities must proactively develop shared norms and guidelines for the use of GenAI that are tailored to their disciplinary contexts.

The field of environmental science could benefit from advances in GenAI tools to quickly develop insights and solutions to pressing environmental challenges like wildfire resilience, climate change, and biodiversity conservation [11,12]. However, as members of the environmental science community, we note that the adoption of GenAI in our field appears highly heterogeneous and unguided. Existing recommendations for GenAI use are typically geared toward science broadly [13] or coding for software engineers [14], which may not address the particular needs of environmental science researchers. Without sufficient resources to help guide researchers in the effective use of GenAI in their work, inequities and inefficient or questionable research practices could proliferate in the field.

OPEN ACCESS

Citation: King RA, Abowd L, Broderick CW, Bradley LM, Czapanskiy MF, Farnisa MM, et al. (2026) Ten simple rules for effective use of generative AI for code development in environmental science. *PLoS Comput Biol* 22(8): e1014627. <https://doi.org/10.1371/journal.pcbi.1014627>

Editor: Stephen Kissler, University of Colorado Boulder, UNITED STATES OF AMERICA

Published: August 17, 2026

Copyright: © 2026 King et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: The authors received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

As members of an environmental science research center that serves the broader scientific community, we have observed individuals who are hesitant to use GenAI, including some co-authors, because they doubt its effectiveness, feel unsure about how to maintain core principles of reproducibility and transparency, and have concerns about the environmental and societal impacts of GenAI. To address these concerns, we brought together our community, representing researchers, software developers, data analysts, and professors, to discuss and identify 10 simple rules for effective use of GenAI in coding for environmental science.

Audience and process

We developed this guide for members of the environmental science community who use GenAI or are curious about using GenAI to write code and conduct data analyses while maintaining a high level of scientific rigor in their work. We all have backgrounds in various domains of environmental science and code on a regular basis, but we use GenAI to varying degrees (rarely to daily). Through a series of facilitated discussions, we identified important lessons learned through our own use of GenAI and subsequently streamlined these into 10 rules. We refined these rules through literature review and co-writing. Given the rapid development of research on GenAI, we frequently include references to pre-print articles and non-peer-reviewed material, such as reports or news articles, to provide timely perspectives on this topic. Our use of these sources should not unduly impact the key points of our narrative because rules were developed through direct user experience.

These rules focus on technical considerations and practices specific to GenAI use, especially where it creates novel challenges or requires new approaches, rather than broader best practices related to scientific rigor, open science, or coding. The rules further assume that an individual has determined that GenAI use is both appropriate and allowable for the identified task and do not describe the ethical considerations underlying that decision. Guidance on the ethical use of GenAI represents an active and complementary area for future research that requires interdisciplinary expertise beyond the scope of this paper [15]. However, we do discuss potential consequences of GenAI use that should be considered to avoid perpetuating social and environmental inequities.

Rules

The rules emerging from our discussion generally fall into one of three categories, similar to those identified in Bridgeford et al. [14]: before, during, and after coding with GenAI. To best apply these rules, users should first understand that most GenAI systems operate through pattern recognition and not through true reasoning [21]. As a result, any code or statistical methods suggested by GenAI are the result of learned patterns in training data, not explicit testing of assumptions, verification of correctness, or awareness of context-specific constraints. This context helps users critically evaluate outputs; for example, recognizing that suggested models may violate statistical assumptions, or that code may appear plausible but incorrect. It also provides important context for ethical considerations. GenAI systems are trained on existing

Table 1. Definitions of key terms.

Term	Definition
Agentic AI	A class of AI system designed to perform complex, multi-step tasks by planning, delegating, and iteratively executing actions using tools and/or models to execute user goals. Agentic AI systems typically combine an LLM with external tools, memory, and control logic that enable goal-directed behavior and adaptive task decomposition, often with varying degrees of human oversight.
AI Agent	An AI agent is an instance of an agentic system that uses a large language model (or similar model) to interpret user instructions, select and use tools, and carry out sequences of actions within a defined environment. AI agents operate under explicit constraints and permissions, and their behavior is typically bounded by system rules, available tools, and human approval mechanisms rather than full autonomy or independent learning.
Context File	A plain text file (e.g., AGENTS.md) that provides persistent background information and instructions on how a GenAI tool or AI agent should respond to prompts and generate output.
Context Window	Similar to working memory, this determines how much text an LLM can remember at one time before needing to summarize it [16].
Generative Artificial Intelligence (GenAI)	A particular class of artificial intelligence algorithms trained to produce new content that can take the form of text, audio, video and imagery [17].
GenAI model	The underlying trained system (or checkpoint) used to generate outputs in a GenAI tool.
GenAI tool	The platform through which one interacts with a GenAI model, ranging from chatbots like ChatGPT [18] or Gemini [19] to interactive development environments (IDEs) like Cursor [20].
Large Language Model (LLM)	A type of deep learning model designed to respond to and generate natural language and other types of content. These models are trained on huge quantities of data to learn underlying patterns they then use to produce content based on what is statistically most likely to follow a prompt [16].

<https://doi.org/10.1371/journal.pcbi.1014627.t001>

data and methods, therefore outputs reflect prior work without clear attribution, and propagate biases present in their training data [22]. Users should remain mindful of issues related to intellectual ownership, proper citation, and responsible use of generated content.

To ground each rule in practice, we draw on an existing data science project as a running example of each rule in [Table 2](#): the Wildfire Resilience Index (WRI), an open-access tool designed to measure community and landscape resilience to wildfire. The WRI integrates heterogeneous data streams including satellite imagery, land-cover and vegetation traits, and socioeconomic variables, into a spatially explicit, multi-domain composite index intended to inform resource managers, decision makers, and communities. Developed by a six-person analytic team with subject matter expertise contributed by a broader working group spanning multiple institutions, the project required exactly the kind of complex, multi-contributor data synthesis GenAI tools are increasingly designed to support. We note that any tools referenced in the manuscript serve as examples and are not recommendations for any particular product.

Before

Rule 1. Understand your domain (or take the time to learn it)

Effective use of GenAI tools requires understanding a task well enough to clearly articulate it, which draws on both subject matter expertise and knowledge of appropriate methods. When users lack the necessary expertise, they should make a deliberate effort to build it or collaborate with subject matter experts to ensure they can guide GenAI toward correct implementations. The risks of using GenAI without this foundation range from relatively innocuous time wasting seen in junior

Table 2. Examples of how the rules were applied to a real data science project, the Wildfire Resilience Index (WRI; <https://www.wildfireindex.org/>).

Rule	Application
Rule 1. Understand your domain (or take the time to learn it)	The project team was led by a PhD scientist with over 10 years of professional experience, with additional oversight of a senior scientist and a 10-member working group that provided the expertise required for such a complex project. The lead project scientist was responsible for every major decision and engaged in code review and output validation. No key decisions were made by AI.
Rule 2. Use a GenAI tool tailored to your task	Early in the project, team members mostly used general-purpose chatbots for broad exploration. As the project matured into a multi-contributor codebase spanning R and Python pipelines, the team shifted to an AI-native IDE (Cursor [20]) to maintain consistency across scripts, manage dependencies, and support automated refactoring.
Rule 3. Set up your GenAI platform for your task	The set-up of GenAI tools evolved as the WRI team gained experience in their use and project complexity increased. We moved from using reusable project descriptions and chatbot system prompts toward more structured context management systems that maintained project information across sessions. Specifically, the team created a shared context file describing the project's spatial extent, the structure of the index, data source conventions, and preferred coding style in both R and Python. Providing this background at the start of each session meant that AI tools generated code consistent with existing pipeline conventions rather than producing generic solutions that required extensive reworking. As the project matured, we started using OpenClaw [59] to build a shared workspace that maintains project context across sessions. Rather than re-briefing the AI at the start of each coding session, the team co-developed a set of structured context files iteratively with AI: SOUL.md encodes the project's scientific values and how the agent should reason about wildfire resilience; IDENTITY.md is a quick-reference card with project name, role, and metadata; AGENTS.md defines procedural rules for common workflows like data ingestion and domain score aggregation; USER.md captures individual researcher roles and preferences so outputs are tailored to each contributor; TOOLS.md documents the R and Python environments and key packages; HEARTBEAT.md schedules recurring checks such as verifying pipeline integrity or flagging when new climate or land cover data becomes available; and BOOTSTRAP.md runs once at setup to configure the agent's identity for the project. These files enabled agents to start each session knowing the project's spatial extent, domain structure, coding conventions, and team norms without any manual re-briefing.
Rule 4. Give GenAI tools only as much access as needed	The team worked on a shared high-performance computing system where multiple projects and users share compute resources. When using AI agents, the team scoped agent access to only the relevant project directories and excluded sensitive paths (e.g., home directories, credential stores, and shared system locations) using configuration-based controls such as ignore rules and restricted workspace settings. While fully isolated virtual machine environments could further reduce system-level risk, they are not used in this context due to the need for direct access to high-performance computing resources, shared file systems, and cluster tooling required for large-scale analysis. Instead, the team relied on least-privilege scoping within the live environment, ensuring the agent could only read and modify files necessary for the task while limiting broader system access. This reduced the risk of unintended changes to unrelated parts of the system while preserving access to essential compute infrastructure.
Rule 5. Adhere to principles of data privacy and security	The team works exclusively with publicly available data such as satellite imagery, census variables, climate model outputs, and species distribution data and therefore faces minimal data privacy risk when using GenAI tools. If extensions to this work require sensitive data, such as Indigenous community data, private community member data, unpublished agency datasets, or fine-grained socioeconomic variables or survey data, the team would adjust the workflow to process data locally or in a folder not visible to AI tools with code generated manually. If coding assistance were needed, the creation of dummy variables and simulated data would be used to generate code, followed by porting it to a location without AI access to run scripts to generate data layers.
Rule 6. Manage long GenAI conversations	Team members noticed that during long exploratory sessions in which the target goal shifted, the model began suggesting code and fixes that ignored earlier outputs and produced complex workarounds. When directly probed, the model identified context rot as the issue – it had forgotten previous attempted solutions. Starting a fresh session with a concise summary of prior decisions restored output quality.
Rule 7. Verify and validate GenAI output	Validation was a continuous, multi-layered process throughout project development. The team generated extensive visualizations at every stage, for example, distributions of domain scores, maps of intermediate outputs, and cross-variable comparisons to catch anomalies before they propagated downstream. Handling missing data was a recurring challenge: NA values in data layers required explicit decisions about imputation, masking, and edge cases, all of which were tested deliberately rather than assumed. The team also used multiple GenAI models for cross-validation, drafting code in one model and having another critique the logic and assumptions. Internal peer review among team members added a human layer of domain-grounded scrutiny. Throughout, GenAI accelerated the validation process itself by generating summary tables and interactive maps, flagging distributional outliers, and annotating functions, making rigorous checking faster and more tractable than it would have been otherwise.

(Continued)

Table 2. (Continued)

Rule	Application
Rule 8. Document your process	The WRI team documented their process at multiple levels. README files in the project repository recorded data source conventions, pipeline structure, and coding standards. GitHub issues served as a living log of analytical decisions and captured what was decided and why, including justifications for domain weighting choices and how edge cases were resolved. A project manager maintained notes on key decisions made during team meetings, creating an auditable record that did not rely on an individual's memory or on GenAI chat histories. Finally, the team created an extensive supplementary document alongside the index describing all data choices and methodological rationale, a resource that serves both transparency and reproducibility for the broader research community.
Rule 9. Leverage GenAI to follow best practices	GenAI was transformative for code quality in the project in ways the team did not anticipate. Rather than simply generating new code, GenAI tools standardized syntax and style across a multi-contributor R and Python codebase, untangled spaghetti dependencies, and ensured that modular components worked together consistently. The team wrote down the best practices they wanted to follow, and AI implemented them uniformly across the entire pipeline quickly. The same process would have taken months to do manually across a medium-sized team. Annotation improved dramatically: functions that would previously have had minimal or no documentation were returned with clear, consistent comments. GenAI code review also caught insidious issues like hard-coded values and flagged them for correction before they could affect results. The team's use of GenAI created a cleaner, more accessible, and more reproducible codebase than they would have produced without GenAI assistance.
Rule 10. Prepare for a minimal coding future	The WRI team invested significant time in figuring out which tools to use, how to configure them effectively, how to prompt well, and how to build and maintain workspace infrastructure. This investment allowed the team to glimpse what environmental science looks like when coding is no longer the bottleneck. The team recovered hours from debugging, dependency management, and code standardization they could redirect toward doing science: asking better questions, interrogating the data more deeply, and accelerating the path from analysis to real-world impact.

<https://doi.org/10.1371/journal.pcbi.1014627.t002>

programmers [23] to substantial methodological errors that influence critical decisions [24]. For example, subtle analytical choices made in ecological modeling (e.g., species distribution modeling), data analysis and bioinformatics pipelines, and data cleaning can alter conclusions and lead to incorrect inferences [25–29]. Left undetected, those mistakes could carry serious consequences for end users. Notably, GenAI can help motivated users identify and fill in knowledge gaps to prevent such consequences (e.g., breaking down complex topics, creating study guides [30]).

Rule 2. Use a GenAI tool tailored to your task

Selecting the right GenAI coding tool requires matching its capabilities to the security needs, scale, and complexity of your work. If tasks involve sensitive data, proprietary code, or other legal considerations, private or self-hosted tools that do not transmit inputs to external servers may be needed. Beyond that constraint, different GenAI tools have strengths and limitations that may suit different project stages. For instance, general-purpose chatbot interfaces, such as ChatGPT [18] and Google Gemini [19], may serve early ideation (e.g., brainstorming analysis strategies) and provide simple coding assistance (e.g., explaining unfamiliar error messages, drafting pseudocode). With more code-intensive tasks, using platforms with built-in GenAI agents (e.g., Posit AI within Positron [31]) provides a familiar coding interface while adding access to more advanced GenAI assistance. As project complexity grows, AI-native IDEs, such as Cursor [20], Devin Desktop (formerly Windsurf) [32], or GitHub Copilot [33], become increasingly valuable. Thanks to their ability to draw from an entire codebase, they provide context-aware code completion, multi-file edits, and inline chat grounded in your actual project files, which helps with tasks such as automated refactoring and bug fixing [34–36]. In all cases, the specific model used (e.g., Gemini 3.5, GPT-5) also influences performance, as models vary in the size of their context window, speed of analysis, and resource usage [37]. Because the GenAI landscape is evolving rapidly, the tools named here reflect the state of the field at the time of writing; the enduring principle is to match the tool to the task, not to adopt any particular product.

Rule 3. Set up your GenAI platform for your task

Deliberate setup of GenAI tools for a specific task improves their performance and efficiency [38] and streamlines communication between you and the AI tool [39]. Describing the goals and providing background information (e.g., the problem

setting, data, constraints, and desired output) improves the consistency and reliability of generated output (see Broderick [40] for an example setup). Depending on the complexity of the task and capabilities of the GenAI tool, the necessary setup can vary substantially. For beginner users, setup may be as simple as creating a reusable project description in a chatbot interface, while more advanced users of GenAI tools may benefit from hierarchical directories of context files for agentic workflows. Users should carefully consider what to include in context to ensure security of sensitive information (e.g., through .ignore files). Proper customization can also heighten reproducibility and transparency of analysis pipelines and complicated code bases when setup files encapsulate the appropriate background information.

During

Rule 4. Give GenAI tools only as much access as needed

The use of AI agents can introduce significant risks because these agents often inherit the same permissions and system access as the user and have been observed to enact adversarial and unexpected behaviors [41,42]. Some recommended practices to minimize risks include: running agents within isolated environments (e.g., virtual machines, containers, or dedicated systems), enforcing restricted permissions (such as through separate non-administrative user accounts), and ensuring secure, authenticated interactions with external tools and resources [43]. Importantly, these practices have restrictions enforced by the system surrounding the AI, rather than relying on the model itself, since language models are unreliable security gatekeepers [44]. Underlying these recommendations is the security principle of least privilege: an AI system should be granted the minimum access necessary. When in doubt, start with the most restrictive access that still allows the task to be completed, and expand only as needed.

Rule 5. Adhere to principles of data privacy and security

GenAI usage complicates data confidentiality and carries leakage risk that users may underestimate [45]. Under standard consumer accounts, user prompts and outputs may be retained and incorporated into AI training pipelines, creating potential exposure [46,47]. Even where contractual agreements contain data in a controlled institutional ecosystem, prompts and outputs can be subject to legal discovery [48]. Several measures can reduce exposure, including the use of anonymized data, deployment of local GenAI models (e.g., LLaMA), and restricting access to sensitive data and folders (e.g., password protection). Users of GenAI should pay particular attention to these measures when working with sensitive or private data, such as those from Indigenous communities [49]. Ultimately, users bear the responsibility to uphold rigorous privacy and security standards.

Rule 6. Manage long GenAI conversations

Long GenAI conversations that use more of a model's available context window degrade model accuracy, a phenomenon known as "context rot" [50,51]. This happens because the self-attention mechanism that governs the model becomes less reliable as the number of tokens increases, raising the risk of "hallucinations". Mitigation techniques to maintain accuracy include keeping chats short and focused and initiating new chats or sessions following changes in tasks and scope. For long-horizon tasks, compaction (i.e., condensing information size while maintaining key details from context), structured note-taking, and multi-agent architectures can help manage context effectively [52].

After

Rule 7. Verify and validate GenAI output

GenAI tools can both introduce and catch errors in code, which makes deliberate validation practices essential. Code produced by GenAI tools may execute cleanly but incorrectly [53], so users should verify outputs at multiple stages to catch issues early. Appropriate validation may include checking that data is read in accurately, testing edge cases informed by

domain expertise, and confirming that intermediate results align with expectations. GenAI can accelerate this process: cross-model review (e.g., writing in one tool, reviewing in another) provides an independent check, while requesting on-demand visualizations or summary tables can surface problems that are hard to catch by reading code alone. GenAI also lowers the barrier to follow good coding practices that support validation [54,55], such as modular code with discrete, testable steps and well-annotated functions [56].

Rule 8. Document your process

GenAI usage can obscure the decision-making process behind an analysis when key assumptions, decision pathways, and iterative refinements live only in chat histories that are difficult to trace and may be lost entirely. To counter this risk, users should document critical decisions directly in their code base through comments, README files, or version-controlled decision logs, akin to lab notebooks in experimental fields, ensuring that the reasoning behind analytical choices is preserved independently of any GenAI tool. In fact, GenAI can assist in this process by logging key decisions and justifications and saving these, such as in a markdown (.md) file. Configuration files that shape the behavior of GenAI, including system prompts, custom instructions, or project knowledge files, can also be version controlled and shared in repositories as part of the analytical record and to help others build on the user's approach. When chat histories contain critical information not captured elsewhere, saving them to local, backed-up storage provides an additional safeguard.

Rule 9. Leverage GenAI to follow best practices

GenAI tools can significantly improve adherence to best practices that people often struggle to maintain consistently, such as thorough documentation, style standardization across multi-contributor code bases, and well-organized data repositories. Tasks that are tedious or time-consuming to do manually, such as annotating functions, flagging hard-coded values, enforcing consistent coding standards, and writing comprehensive metadata, are well-suited to GenAI and can be applied uniformly across an entire project. Per rule 7, all AI-generated content should be reviewed and validated to ensure that it documents the project accurately. The result is cleaner, more auditable, and more reproducible work—not because the standards have changed, but because GenAI dramatically lowers the cost of meeting them.

Rule 10. Prepare for a minimal coding future

Limited coding proficiency has long been a barrier to research in environmental science [57]. GenAI is shifting the bottleneck from programming toward scientific thinking, research design, and methodological rigor. Going forward, the ability to clearly communicate goals, frame problems, and effectively direct GenAI systems may well be more valuable than programming fluency for research productivity [58]. To prepare for this shift, users of GenAI will benefit from a continued focus on foundational skills, including a solid understanding of quantitative methods, data provenance, and research design, along with the ability to articulate scientific intent with precision. These skills are not new demands but fundamental needs to produce robust science efficiently with any tool, including GenAI.

What could go wrong

In addition to discussing how individuals and communities can improve their GenAI coding practices, our workshop discussions surfaced a range of concerns stemming from the rapid adoption of GenAI in combination with a sparse regulatory environment. At their core, many of our concerns pointed to the same underlying issue: the risk that the transition to GenAI may narrow, rather than broaden, participation in environmental data science. Additional considerations related to the environmental impacts of expanding data centers to fuel the expansion of GenAI. We explore implications for both types of concerns below.

GenAI has been heralded as an equalizer by some, lowering barriers to entry and broadening participation. Yet, early research suggests that benefits accrue unevenly across experience levels, gender, and social context. One emerging pattern is that experience shapes how effectively users leverage GenAI. Within software development, entry-level developers used GenAI most frequently, but senior-level developers experienced greater benefits of AI adoption [60]. This pattern is paralleled in scientific computing, where students and less experienced programmers adopt GenAI at higher rates and simultaneously face disproportionately high consequences from over-reliance on and insufficient validation of GenAI [61]. This may result in a growing skills gap that accelerates broader changes across sectors that increasingly rely on GenAI.

Gender and wealth are two additional axes on which benefits of GenAI use appear to diverge. Male researchers tend to experience higher productivity gains and use GenAI more than their female counterparts, a pattern that has persisted across regions, sectors, and occupations [4,9]. Persistent underrepresentation of women using GenAI risks self-reinforcement, because systems trained on data that under-sample women may adapt more to the preferences, needs, and norms of men, potentially widening disparities in technology adoption and economic opportunity [9]. Beyond gender identity, patterns of wealth reveal additional barriers to GenAI adoption. A recent United Nations report found that while roughly two-thirds of people in some high-income countries use GenAI tools, usage in many low-income countries hovers near 5%—a staggering gap [8]. These patterns persist even within the wealthiest nation; in the United States, search interest in ChatGPT has been concentrated in coastal cities and depressed in rural regions including the Midwest, Appalachia, and the South, with county-level wealth emerging as a key predictor [62]. Taken together, these patterns suggest that the benefits of GenAI are not accruing equitably, but rather along familiar axes of social and economic inequality.

Not only are access disparities unlikely to resolve on their own, but the structural economics of the GenAI industry may actively deepen them as companies shift to a fee-based model [63]. Most leading GenAI companies (e.g., OpenAI, Anthropic) offer a tiered subscription model that locks the most powerful features behind paywalls. The shift to tiered pricing from free access will likely keep the most capable models out of reach of individuals at underfunded institutions, in low-income countries, or otherwise operating without institutional support. For scientists and coders in these circumstances, a pay-to-play model for GenAI does not democratize participation in science; rather, it perpetuates structural inequities. However, while the aforementioned trajectories of widening disparities are concerning, they are not predetermined outcomes. Realizing the potential of GenAI to help mitigate social inequalities will require both acknowledgement of existing inequities and deliberate policymaking to counteract the disparate harms GenAI could entrench [64].

A key issue among the policy challenges is the potential disruption GenAI could bring to the workforce, which raises deeper questions about institutional responsibility [65]. For decades, the technology industry has successfully advocated for large-scale public investment in computer science education, with the implicit promise that training more software developers would meet a robust and growing demand for their labor. The CHIPS and Science Act of 2022 alone authorized tens of billions for NSF workforce development in key technology areas, including AI and software, representing a substantial public commitment to building a pipeline of computing talent [66]. Yet, GenAI has disrupted that pipeline with striking speed. Unemployment among recent computer science and computer engineering graduates now exceeds 7.8% and 7%, respectively, higher than that of recent art history or English majors [67], while postings for software development roles fell by more than 70% between 2022 and 2025 [68]. As environmental science has become a more data-intensive field, often requiring advanced computational skills [69], the proliferation of GenAI could also alter the balance of jobs within this discipline. Although the data available to make predictions about the future is scarce, the current trend raises an uncomfortable question: if public funds built the pipeline, and private industry deployed the technology that undermined it, who bears responsibility for the workers left behind? Answering that question and designing policy responses equal to the disruption may be among the most consequential challenges GenAI poses at a systems level.

Finally, as environmental scientists, many discussions circled back to concerns about the potential environmental impacts resulting from the proliferation of GenAI. Many of these concerns are not unique to GenAI, but the newness of these tools makes comprehensive research quantifying their impacts sparse. There are, however, an array of social and

environmental concerns across the life cycle of GenAI: from increased mining of rare earth metals for hardware components to noise pollution from cooling fans and the increased demand for electricity and water by data centers [70]. Energy use by data centers in the US is projected to represent 4%–12% of all electricity use by 2030 [71,72]. Cooling data centers requires large quantities of water, and by 2028 data centers are projected to use up to 32 billion gallons of water per year [73]. Limited federal regulations specific to data centers risk unsustainable development that could harm local communities and disproportionately impact marginalized communities if they follow trends similar to other pollutants [74,75]. There are opportunities to minimize the impacts, ranging from co-location of data centers with renewable energy sources to improving energy efficiency to leveraging reclaimed wastewater or other cooling methods to reduce water use [76]. Careful consideration of the environmental impacts of GenAI coupled with sound policies or regulations to encourage responsible development will be needed to avoid propagating harm on communities and ecosystems.

Conclusions

GenAI is a powerful tool, but adopting it effectively for environmental science requires developing new skills and attending to considerations throughout the project that may be unfamiliar to many researchers. As a community, we are navigating a transition into this AI era together: learning, testing, adapting, and improving our skills with these new tools as we go. The 10 rules we present here offer a starting point for that process and complement existing standards that remain crucial to the transparency and rigorous practice of science [57]. When following the rules presented here, coding with GenAI can create rigorous and timely standalone products via transparent, reproducible, and accessible methods that adhere to our community principles. At the same time, organizations and institutions must recognize that without deliberate intervention, the benefits of GenAI may continue to accrue inequitably, maintaining or even widening existing disparities rather than lowering barriers to entry in the field. It is natural to feel skepticism or wariness toward any new tool, and we do not dismiss those concerns. Nevertheless, we have found it highly productive to engage openly with colleagues and explore what works, what does not, and what we do and do not want from these tools. The challenges GenAI poses for our field will not resolve themselves. Meeting them requires the kind of honest, sometimes difficult conversations that move a community forward. Ultimately, we have the power to shape how GenAI integrates into environmental data science, but only if we work proactively and collectively as a community towards that future.

AI use statement

During the development of this manuscript, GenAI tools were used to facilitate writing and synthesis of ideas. Specifically, we used Elicit (<https://elicit.com/>) for portions of the literature review, Claude Opus 4.6 and OpenAI's GPT-5.3 mini for editing and revising text. The authors identified the rules without input from GenAI.

Acknowledgments

The authors would like to thank the administrative and support staff at NCEAS for helping to facilitate the workshop that led to this paper. We would also like to thank others who contributed to our initial discussions, including Shalanda Grier, Casey O'Hara, Melanie Frasier, and Sophia Manos.

Author contributions

Conceptualization: Rachel A. King, Caitlin R. Fong.

Investigation: Rachel A. King.

Methodology: Rachel A. King, Caitlin R. Fong.

Project administration: Caitlin R Fong.

Writing – original draft: Rachel A. King, Laurel Abowd, Carlo W. Broderick, LM Bradley, Max F. Czapanskiy, Mona M. Farnisa, Erica M. Ferrer, Carmen Galaz García, Darian Gill, Nicole M. Greco, Juliette Jacquemont, Justin A. Kadi, Li Kui, Gretchen LeBuhn, Liying Li, Abigail Meyer, Marisa Morse, Evan Patrick, Samantha Shanny-Csik, Nicholas A. C. Tucker, Zhe Wang, Caitlin R. Fong.

Writing – review & editing: Rachel A. King, Laurel Abowd, Carlo W. Broderick, LM Bradley, Max F. Czapanskiy, Mona M. Farnisa, Erica M. Ferrer, Carmen Galaz García, Darian Gill, Nicole M. Greco, Juliette Jacquemont, Justin A. Kadi, Li Kui, Gretchen LeBuhn, Liying Li, Abigail Meyer, Marisa Morse, Evan Patrick, Samantha Shanny-Csik, Nicholas A. C. Tucker, Zhe Wang, Caitlin R. Fong.

References

1. Siler K. The diffusion of large language models in published academic articles. *Proc Natl Acad Sci U S A*. 2026;123(22):e2605754123. <https://doi.org/10.1073/pnas.2605754123> PMID: 42213755
2. Liang W, Zhang Y, Wu Z, Lepp H, Ji W, Zhao X, et al. Quantifying large language model usage in scientific papers. *Nat Hum Behav*. 2025;9(12):2599–609. <https://doi.org/10.1038/s41562-025-02273-8> PMID: 40760036
3. Pramanick A, Hou Y, Mohammad SM, Gurevych I. Transforming scholarly landscapes: the influence of large language models on academic fields beyond computer science. *PLoS One*. 2026;21(1):e0337127. <https://doi.org/10.1371/journal.pone.0337127> PMID: 41533700
4. Mohammadi E, Thelwall M, Cai Y, Collier T, Tahamtan I, Eftekhari A. Is generative AI reshaping academic practices worldwide? A survey of adoption, benefits, and concerns. *Inf Process Manag*. 2026;63(1). <https://doi.org/10.1016/j.ipm.2025.104350>
5. Kusumegi K, Yang X, Ginsparg P, de Vaan M, Stuart T, Yin Y. Scientific production in the era of large language models. *Science*. 2025;390(6779):1240–3. <https://doi.org/10.1126/science.adw3000> PMID: 41411417
6. Zenni RD, Andrew NR. Artificial Intelligence text generators for overcoming language barriers in ecological research communication. *Austral Ecol*. 2023;48(7):1225–9. <https://doi.org/10.1111/aec.13417>
7. Ji X, Wu X, Deng R, Yang Y, Wang A, Zhu Y. Utilizing large language models for identifying future research opportunities in environmental science. *J Environ Manage*. 2025;373:123667. <https://doi.org/10.1016/j.jenvman.2024.123667> PMID: 39673851
8. United Nations Development Programme (UNDP). The next great divergence [Internet]. 2025 [cited 2026]. Available from: https://www.undp.org/sites/g/files/zskgke326/files/2025-12/undp-rbap-the-next-great-divergence_1.pdf
9. Otis NG, Delecourt S, Cranney K, Koning R. Global evidence on gender gaps and generative AI. Harvard Business School Working Paper No. 25-023, 2024 [cited 2026]. Available from: <https://www.hks.harvard.edu/centers/cid/publications/global-evidence-gender-gaps-and-generative-ai>
10. Wills S, Poon STS, Salili-James A, Scott B. The use of generative AI for coding in academia. *Methods Ecol Evol*. 2024;15(12):2189–91. <https://doi.org/10.1111/2041-210x.14454>
11. Blair GS, Henrys P, Leeson A, Watkins J, Eastoe E, Jarvis S. Data science of the natural environment: a research roadmap. *Front Environ Sci*. 2019;7:121. <https://doi.org/10.3389/fenvs.2019.00121>
12. Pollock LJ, Kitzes J, Beery S. Harnessing artificial intelligence to fill global shortfalls in biodiversity knowledge. *Nat Rev Biodivers*. 2025;1:166–82. <https://doi.org/10.1038/s44358-025-00022-3>
13. Helmy M, Jin L, Alhossary A, Mansour T, Pellagrini D, Selvarajoo K. Ten simple rules for optimal and careful use of generative AI in science. *PLoS Comput Biol*. 2025;21(10):e1013588. <https://doi.org/10.1371/journal.pcbi.1013588> PMID: 41150680
14. Bridgeford EW, Campbell I, Chen Z, Lin Z, Ritz H, Vandekerckhove J, et al. Ten simple rules for ai-assisted coding in science. *arXiv:2510.22254v2* [Preprint]. 2025. [cited 2026 May 18]. Available from: <https://arxiv.org/abs/2510.22254>
15. Huang Y, Arora C, Huong WC, Kanij T, Madugalla A, Grundy J. Ethical concerns of generative AI and mitigation strategies: a systematic mapping study. *Appl Soft Comput*. 2026;193:114789. <https://doi.org/10.1016/j.asoc.2026.114789>
16. Stryker C, Lee F, Bergman D, Scapicchio M. The 2026 Guide to Machine Learning [Internet]. IBM Think. [cited 2026]. Available from: <https://www.ibm.com/think/machine-learning#605511093>
17. Vassilev A, Oprea A, Fordyce A, Anderson H, Davies X, Hamin M. Adversarial machine learning: a taxonomy and terminology of attacks and mitigations. Gaithersburg, MD: National Institute of Standards and Technology. 2025. <https://doi.org/10.6028/NIST.AI.100-2e2025>
18. OpenAI. ChatGPT [large language model]. Version GPT-5.3 mini. San Francisco (CA): OpenAI; 2026 [cited 2026 Jul 16]. Available from: <https://chatgpt.com>
19. Google. Gemini [large language model]. Version 3.5 Flash. Mountain View (CA): Google; 2026 [cited 2026 Jul 16]. Available from: <https://gemini.google.com>
20. Anysphere. Cursor [software]. Version 3.11.25. San Francisco (CA): Anysphere; 2026 [cited 2026 Jul 16]. Available from: <https://cursor.com>
21. Pahuja S, Kukreja S, Singh A. Comprehensive review of generative artificial intelligence: mechanisms, models and applications. *Procedia Comput Sci*. 2025;258:3731–40. <https://doi.org/10.1016/j.procs.2025.04.628>

22. Freeman L, Robert J, Wojton H. The impact of generative AI on test & evaluation: challenges and opportunities. In: Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering, 2025. 1376–80. <https://doi.org/10.1145/3696630.3728723>
23. Shen JH, Tamkin A. How AI impacts skill formation. arXiv:2601.20245 [Preprint]. 2026. [cited 2026 May 18]. Available from: <https://arxiv.org/abs/2601.20245>
24. Dobler D, Binder H, Boulesteix AL, Igelmann JB, Köhler D, Mansmann U, et al. ChatGPT as a tool for biostatisticians: a tutorial on applications, opportunities, and limitations. *Stat Med*. 2025;44(23–24):e70263. <https://doi.org/10.1002/sim.70263>
25. Christin S, Hervet É, Lecomte N. Applications for deep learning in ecology. *Methods Ecol Evol*. 2019;10(10):1632–44. <https://doi.org/10.1111/2041-210x.13256>
26. Norouzzadeh MS, Nguyen A, Kosmala M, Swanson A, Palmer MS, Packer C, et al. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proc Natl Acad Sci U S A*. 2018;115(25):E5716–25. <https://doi.org/10.1073/pnas.1719367115> PMID: 29871948
27. Hallgren W, Santana F, Low-Choy S, Zhao Y, Mackey B. Species distribution models can be highly sensitive to algorithm configuration. *Ecol Modell*. 2019;408:108719. <https://doi.org/10.1016/j.ecolmodel.2019.108719>
28. Rick JA, Brock CD, Lewanski AL, Golcher-Benavides J, Wagner CE. Reference genome choice and filtering thresholds jointly influence phylogenomic analyses. *Syst Biol*. 2024;73(1):76–101. <https://doi.org/10.1093/sysbio/syad065> PMID: 37881861
29. Tiu ESK, Huang YF, Ng JL, AIDahoul N, Ahmed AN, Elshafie A. An evaluation of various data pre-processing techniques with machine learning models for water level prediction. *Nat Hazards*. 2021;110(1):121–53. <https://doi.org/10.1007/s11069-021-04939-8>
30. Pallant JL, Blijlevens J, Campbell A, Jopp R. Mastering knowledge: the impact of generative AI on student learning outcomes. *Stud High Educ*. 2026;51(4):714–35. <https://doi.org/10.1080/03075079.2025.2487570>
31. Posit PBC. Positron [software]. Version 2026.07.1. Boston (MA): Posit PBC; 2026 [cited 2026 Jul 16]. Available from: <https://positron.posit.co>
32. Cognition. Devin Desktop [software]. Version 2026.07.15. San Francisco (CA): Cognition; 2026 [cited 2026 Jul 16]. Available from: <https://devin.ai/>
33. GitHub. GitHub Copilot [software]. Version 1.0.24. San Francisco (CA): GitHub; 2026 [cited 2026 Jul 16]. Available from: <https://github.com/features/copilot>
34. Chintakunta SS, Nascimento N, Everton G. Large language models in the data science lifecycle: a systematic mapping study. arXiv. 2025. <https://doi.org/10.48550/arXiv.2508.11698>
35. Ross SI, Martinez F, Houde S, Muller M, Weisz JD. The programmer's assistant: conversational interaction with a large language model for software development. In: Proceedings of the 28th International Conference on Intelligent User Interfaces, 2023. 491–514. <https://doi.org/10.1145/3581641.3584037>
36. Meyer JG, Urbanowicz RJ, Martin PCN, O'Connor K, Li R, Peng P-C, et al. ChatGPT and large language models in academia: opportunities and challenges. *BioData Min*. 2023;16(1):20. <https://doi.org/10.1186/s13040-023-00339-9> PMID: 37443040
37. Altman S, Couch S. Which AI model writes the best R code? 2023 Jan 13 [cited 2026 May 18]. In: Posit Blog [Internet]. Posit Software. Available from: <https://posit.co/blog/r-llm-evaluation-03>
38. Paul S, Kumar R, Das L. Smarter AI through prompt engineering: insights and case studies from data science application. arXiv. 2026. <https://doi.org/10.48550/arXiv.2602.00337>
39. Geroimenko V. Beyond and after prompt engineering: the future of AI communication. Springer Nature Switzerland AG; 2026. <https://doi.org/10.1007/978-3-032-04569-0>
40. Broderick C. Coding-agent rules for data science. Zenodo. 2026. <https://doi.org/10.5281/zenodo.21364495>
41. He Y, Wang E, Rong Y, Cheng Z, Chen H. Security of AI agents. In: IEEE/ACM International Workshop on Responsible AI Engineering (RAIE), 2025. 45–52. <https://doi.org/10.1109/RAIE66699.2025.00013>
42. Shapira N, Wendler C, Yen A, Sarti G, Pal K, Floody O, Belfki A, et al. Agents of chaos. arXiv:2602.20021 [Preprint]. 2026. [cited 2026 May 18]. Available from: <https://doi.org/10.48550/arXiv.2602.20021>
43. Sengupta A. Securing the autonomous future a comprehensive analysis of security challenges and mitigation strategies for AI agents. *IJSREM*. 2024;08(12):1–2. <https://doi.org/10.55041/ijrem40091>
44. Wu Y, Roesner F, Kohno T, Zhang N, Iqbal U. IsolateGPT: an execution isolation architecture for LLM-based agentic systems. arXiv. 2025. <https://doi.org/10.48550/arXiv.2403.04960>
45. Gershon B. Securing AI coding tools: permission controls and credential protection for engineering teams. 2025 Dec 2 [cited 19 May 2026]. Available from: <https://www.briangershon.com/blog/securing-ai-coding-tools/>
46. Huang J, Shao H, Chang KCC. Are large pre-trained language models leaking your personal information?. In: Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, 2022. 2038–47. <https://doi.org/10.18653/v1/2022.findings-emnlp.148>
47. Itoi NG. Be careful what you tell your AI chatbot [Internet]. Stanford, CA: Stanford Institute for Human-Centered AI; 2025 Oct 15 [cited 2026 May 19]. Available from: <https://hai.stanford.edu/news/be-careful-what-you-tell-your-ai-chatbot>

48. Perez A. Navigating discovery risks when using AI tools in legal practice [Internet]. Des Moines, IA: Iowa State Bar Association; 2026 Mar 1 [cited 2026 May 19]. Available from: <https://www.iowabar.org/?pg=IowaBarBlog&blAction=showEntry&blogEntry=138783>
49. Carroll SR, Garba I, Figueroa-Rodriguez OL, Holbrook J, Lovett R, Materechera S. The CARE principles for indigenous data governance. *Data Sci J*. 2020;19:43. <https://doi.org/10.5334/dsj-2020-043>
50. Du Y, Tian M, Ronanki S, Rongali S, Bodapati SB, Galstyan A. Context length alone hurts LLM performance despite perfect retrieval. In: Findings of the Association for Computational Linguistics: EMNLP 2025, 2025. <https://doi.org/10.18653/v1/2025.findings-emnlp.1264>
51. Lee TB. Context Rot: The emerging challenge [Internet]. Understanding AI. 2025 [cited 2026]. Available from: <https://www.understandingai.org/p/context-rot-the-emerging-challenge>
52. Anthropic. Effective context engineering for AI agents [Internet]. 2025 [cited 2026]. Available from: <https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents>
53. Chatterjee A. State of Code Developer Survey report: the current reality of AI coding [Internet]. Geneva: SonarSource; 2026 Jan 8 [cited 2026 May 19]. Available from: <https://www.sonarsource.com/blog/state-of-code-developer-survey-report-the-current-reality-of-ai-coding>
54. Ivimey-Cook ER, Pick JL, Bairos-Novak KR, Culina A, Gould E, Grainger M, et al. Implementing code review in the scientific workflow: insights from ecology and evolutionary biology. *J Evol Biol*. 2023;36(10):1347–56. <https://doi.org/10.1111/jeb.14230> PMID: 37812156
55. Swannack TM, Cushway KC, Carrillo CC, Calvo C, Determan KR, Mierzejewski CM. Cracking the code: linking good modeling and coding practices for new ecological modelers. *Ecol Modell*. 2025;499:110926. <https://doi.org/10.1016/j.ecolmodel.2024.110926>
56. Hartung T, Kleinstreuer N. Challenges and opportunities for validation of AI-based new approach methods. *ALTEX*. 2025;42(1):3–21. <https://doi.org/10.14573/altex.2412291> PMID: 39815689
57. Lowndes JSS, Best BD, Scarborough C, Afflerbach JC, Frazier MR, O'Hara CC, et al. Our path to better science in less time using open data science tools. *Nat Ecol Evol*. 2017;1(6):160. <https://doi.org/10.1038/s41559-017-0160> PMID: 28812630
58. New York Times. AI coders [Internet]. The Daily. 2026 Apr 14 [cited 2026]. Available from: <https://www.nytimes.com/2026/04/14/podcasts/the-daily/ai-coders.html>
59. OpenClaw Contributors. OpenClaw - Personal AI Assistant. Version 2025.5.19-BETA.1 [software]. [cited 2026 May 18]. Available from: <https://github.com/openclaw/openclaw>
60. Daniotti S, Wachs J, Feng X, Neffke F. Who is using AI to code? Global diffusion and impact of generative AI. *Science*. 2026;391(6787):831–5. <https://doi.org/10.1126/science.adz9311> PMID: 41570112
61. O'Brien G, Parker A, Eisty N, Carver J. A survey of generative AI adoption and perceived productivity among scientists who program. arXiv:2512.19644 [Preprint]. 2025 [cited 2026 May 18]. Available from: <https://arxiv.org/abs/2512.19644>
62. Daepf MIG, Counts S. The emerging generative artificial intelligence divide in the United States. *ICWSM*. 2025;19:443–56. <https://doi.org/10.1609/icwsml.v19i1.35825>
63. Zuora. The future of GenAI pricing metrics and models [Internet]. 2024 [cited 2026]. Available from: <https://www.zuora.com/resource/the-future-of-genai-pricing-metrics-and-models/>
64. Capraro V, Lentsch A, Acemoglu D, Akgun S, Akhmedova A, Bilancini E, et al. The impact of generative artificial intelligence on socioeconomic inequalities and policy making. *PNAS Nexus*. 2024;3(6):pgae191. <https://doi.org/10.1093/pnasnexus/pgae191> PMID: 38864006
65. New York Times. Big tech told kids to code. The jobs didn't follow [Internet]. The Daily. 2025 Sep 29 [cited 2026]. Available from: <https://www.nytimes.com/2025/09/29/podcasts/the-daily/big-tech-told-kids-to-code-the-jobs-didnt-follow.html>
66. HR 4346. CHIPS and Science Act. 117th Congress of the United States. 2022. Available from: <https://www.congress.gov/bills/117th-congress/house-bill/4346>
67. Federal Reserve Bank of New York. College labor market: outcomes by major [Internet]. [cited 2026 Mar 3]. Available from: <https://www.newyork-fed.org/research/college-labor-market>
68. Federal Reserve Bank of St. Louis. FRED: software developer job postings index (IHLIDXUSTPSOFTDEVE) [Internet]. [cited 2026 Mar 3]. Available from: <https://fred.stlouisfed.org/series/IHLIDXUSTPSOFTDEVE>
69. Michener W, Jones M. Ecoinformatics: supporting ecology as a data-intensive science. *Trends Ecol Evol*. 2012;27:85–93. <https://doi.org/10.1016/j.tree.2011.11.016>
70. Hosseini M, Gao P, Vivas-Valencia C. A social-environmental impact perspective of generative artificial intelligence. *Environ Sci Ecotechnol*. 2024;23:100520. <https://doi.org/10.1016/j.esel.2024.100520> PMID: 39811486
71. Environmental and Energy Study Institute (EESI). Data center energy needs are upending power grids and threatening the climate [Internet]. 2025 [cited 2026]. Available from: <https://www.eesi.org/articles/view/data-center-energy-needs-are-upending-power-grids-and-threatening-the-climate>
72. Electric Power Research Institute (EPRI). Powering intelligence: analyzing artificial intelligence and data center energy consumption [Internet]. 2024 [cited 2026]. Available from: https://www.wpr.org/wp-content/uploads/2024/06/3002028905_Powering-Intelligence_-_Analyzing-Artificial-Intelligence-and-Data-Center-Energy-Consumption.pdf
73. World Resources Institute (WRI). US Data Center growth impacts [Internet]. 2026 [cited 2026]. Available from: <https://www.wri.org/insights/us-data-center-growth-impacts>

74. Schaider LA, Swetschinski L, Campbell C, Rudel RA. Environmental justice and drinking water quality: are there socioeconomic disparities in nitrate levels in U.S. drinking water?. *Environ Health*. 2019;18(1):3. <https://doi.org/10.1186/s12940-018-0442-6>
75. Jbaily A, Zhou X, Liu J, Lee T-H, Kamareddine L, Verguet S, et al. Air pollution exposure disparities across US population and income groups. *Nature*. 2022;601(7892):228–33. <https://doi.org/10.1038/s41586-021-04190-y> PMID: [35022594](https://pubmed.ncbi.nlm.nih.gov/35022594/)
76. Walker CD, Goldsmith I. From energy use to air quality, the many ways data centers affect US communities. World Resources Institute [Internet]. 2026 Feb 17 [cited 2026 Jul 16]. Available from: <https://www.wri.org/insights/us-data-center-growth-impacts>