

Vibe Coding als Assessment fachdidaktischer und KI-bezogener Kompetenzen

Erfahrungen aus dem Lehrprojekt KIVIMA

Valentin Katter und Nicolás Alejandro Alarcón Relmucao

Der Einsatz von KI in der Hochschullehre verändert derzeit grundlegend die Diskussion über Lernen, Lehren und Prüfen. Während einige vor negativen Auswirkungen auf den Kompetenzerwerb warnen, sehen andere darin die Chance, (Hochschul-)lehre neu zu denken und stärker auf Prozesse des Verstehens und Reflektierens auszurichten. Auf der einen Seite stehen die Bedenken. Da ist zunächst das Phänomen, welches mit dem Begriff *Skill Skipping* beschrieben wird (Nuxoll 2024). Wenn Lernende KI nutzen, um den Bearbeitungsprozess zu überspringen, gehen potentielle Lerngelegenheiten zum Aufbau mathematischer Kompetenzen verloren. Eng damit verwandt ist das Phänomen des *Cognitive Offloading*, das Fan et al. (2025) empirisch nachweisen konnten. Lernende, die mit generativer KI arbeiteten, erzielten kurzfristig bessere Ergebnisse, lagerten jedoch systematisch die metakognitiven Schritte des Planens, Überwachens und Evaluierens, die für nachhaltiges Lernen entscheidend sind, an die KI aus. Bastani et al. (2025) konnten in diesem Zusammenhang zeigen, dass der unreflektierte KI-Einsatz beim Lernen den Kompetenzerwerb langfristig beeinträchtigen kann und Laun et al. (2026) kommen für den Mathematikunterricht zu dem Ergebnis, dass menschliches Tutoring bei Modellierungsaufgaben größere Leistungszuwächse ermöglicht als KI-gestützte Unterstützung. Dazu kommt das Problem der Fehlinformation: LLMs halluzinieren, sie erfinden Quellen, formulieren fachlich klingende Sätze ohne inhaltliche Substanz oder verwenden mathematische Fachbegriffe fehlerhaft (Katter & Huget, 2026).

Auf der anderen Seite stehen Potenziale. Mertens (2025) thematisiert in ihrer Arbeit die Potenziale eines chancengerechten Einsatzes generativer KI und weist darauf hin, dass LLMs bestehende Bildungungleichheiten abmildern könnten, indem sie Lernenden unabhängig von ihrer Bildungsbiografie einen niedrigschwelligen Zugang zu individueller Unterstützung ermöglichen. Weißels und Maibaum (2026) argumentieren außerdem, dass generative KI bei reflektiertem Einsatz nicht zwangsläufig zum Verlust von Kompetenzen führt, sondern sehen im Konzept des *Newskilling* den möglichen Erwerb neuer Fähigkeiten, die ohne KI gar nicht entstehen würden. Walkington et al. (2025)

zeigen zudem, dass generative KI im Mathematikunterricht kreative Prozesse freisetzen kann, die im traditionellen Unterricht kaum entstehen würden. In ihrer Studie entwickelten Mittelschüler*innen mit ChatGPT eigene mathematische Aufgaben und passende Illustrationen, die ihre persönlichen Interessen aufgriffen. Auch aus didaktischer Perspektive eröffnet KI neue Möglichkeiten. Puentedura (2014) beschreibt mit seinem SAMR-Modell verschiedene Stufen des Technologieeinsatzes, von der einfachen Substitution bis hin zur Redefinition, also der Schaffung von Lernformaten, die ohne digitale Technologie nicht realisierbar wären. Genau auf dieser höchsten Stufe kann kompetent eingesetzte KI wirksam werden, indem Aufgabenformate entwickelt werden, die zuvor schwer umzusetzen waren.

Die didaktischen Potenziale von KI für die Hochschullehre zeigen sich besonders dort, wo nicht lediglich bestehende Aufgaben schneller bearbeitet werden, sondern neue Formen mathematischer Tätigkeit entstehen. Interessant werden KI-gestützte Lernumgebungen insbesondere dann, wenn sie kreative Prozesse anregen, mathematisches Kommunizieren fördern und prozessbezogene Kompetenzen sichtbar machen.

Das Lehrprojekt KIVIMA – KI-gestützte Visualisierungen im Mathematikunterricht

Das über zwei Semester laufende und vom Qualitätsfond für Lehre der Uni Bielefeld geförderte Lehrprojekt „KIVIMA – KI-gestützte Visualisierungen im Mathematikunterricht“ richtet sich an Lehramtsstudierende der Mathematik für die Sekundarstufen. Es adressiert zwei zentrale Desiderate der Lehramtsausbildung, die von Studierenden regelmäßig artikuliert werden: (1) Ein stärkerer Praxisbezug im Studium und (2) der Wunsch nach konkreten Anwendungen von KI im Mathematikunterricht. Die Idee des Projekts ist es, Studierende im Vibe Coding zu unterrichten und sie am Ende des Semesters mit Schulklassen im Lehr-Lernlabor teuto-lab mathematik ihre entwickelten Lernumgebungen erproben zu lassen. Das bewährte Laborkonzept, in dem Kleingruppen nacheinander drei Stationen durch-

laufen, ermöglicht es jedem Studierendenteam, seine Visualisierung mehrfach im Unterrichtseinsatz zu erproben und zu reflektieren.

Vibe Coding im Mathematikunterricht

Den Begriff „Vibe Coding“ prägte der KI-Forscher Andrej Karpathy im Februar 2025 in einem Beitrag auf dem sozialen Netzwerk X (x.com/karpathy/status/1886192184808149383). Er beschreibt damit ein Vorgehen, bei dem Nutzende vollständig auf den natürlichsprachlichen Dialog mit einem LLM vertrauen, ohne sich um die technische Implementierung zu kümmern (vgl. Dilling 2026). Das LLM übernimmt dabei die gesamte Programmierung, während die Nutzenden lediglich beschreiben, was die Anwendung leisten soll. Für den Mathematikunterricht bringt dieses Prinzip gewisse Vorteile mit sich. Die Aufmerksamkeit kann vollständig auf die mathematische Idee und die inhaltliche Substanz des geplanten Applets gerichtet werden. Auch ohne Programmierkenntnisse lassen sich vielfältige digitale Lernangebote wie spielerische Lernumgebungen, Lernapps, Lernpfade oder Quizzes entwickeln. Wie Dilling (2026) in seinem Überblicksbeitrag ausführt, eignet sich Vibe Coding besonders für die Erstellung mathematischer Applets, da diese klar abgegrenzte Ziele verfolgen und sich mit einer überschaubaren Menge Code umsetzen lassen.

Einsatzmöglichkeiten und Erfahrungen aus dem Seminar

Im Seminar KIVIMA haben Studierende in einem iterativen Prozess interaktive HTML-Lernumgebungen zu zentralen mathematischen Themen der Sekundarstufe I entwickelt. Mit einem Initialprompt wurden erste Applet-Versionen erzeugt, die anschließend kritisch evaluiert und durch weitere Prompts schrittweise verfeinert wurden. Entstanden sind dabei Bruch-Memory-Spiele, Lernpfade zur quadratischen Ergänzung und Simulationen des Galtonbretts. Da der Einsatz von Vibe Coding in der mathematikdidaktischen Hochschullehre einen neuartigen Ansatz darstellt, werden Potenziale und Grenzen dieses Vorgehens kontinuierlich und in enger Rückkopplung mit den Studierenden evaluiert. Der Entwicklungsstand des Seminars sowie ausgewählte Materialien werden auf der Plattform kivima.eu dokumentiert und der Fachöffentlichkeit zugänglich gemacht. Im Verlauf der Umsetzung des Vibe-Coding-Ansatzes mit den Studierenden wurde dabei schnell deutlich, dass ein gelungener Prozess vor allem eine klare didaktische Idee voraussetzt. Wer nicht weiß, welche mathematischen Handlungen ein Applet ermöglichen soll, hat Probleme es sinnvoll zu prompten.

Der Erstellungsprozess selbst wird so zur Gelegenheit, fachliches und didaktisches Wissen zu reflektieren.

Mit welchen Tools kann man Vibe Coden?

Grundsätzlich lässt sich Vibe Coding mit jeder LLM-Oberfläche betreiben. Im Seminar wurde zunächst mit BIKI, der hochschuleigenen KI-Plattform der Universität Bielefeld, gearbeitet, über die verschiedene Sprachmodelle zugänglich sind. Dabei zeigte sich jedoch schnell, dass die Qualität der generierten Applets stark vom gewählten Modell abhing. Bei ungeeigneten Modellen entstanden Lernumgebungen, die kaum verwendbar waren und das didaktische Ziel verfehlten. Als besonders geeignet erwiesen sich im Seminar ChatGPT (OpenAI) und Claude (Anthropic), die beide im regulären Chatmodus solide Ergebnisse liefern und im jeweiligen Code-Modus nochmals deutlich bessere. Beide Plattformen ermöglichen zudem eine direkte Vorschau des erzeugten HTML-Codes, was den iterativen Überarbeitungsprozess erheblich erleichtert. ChatGPT bietet darüber hinaus mit der Codex-Umgebung und der direkten GitHub-Integration die Möglichkeit, erstellte Applets unmittelbar zu hosten und an der überarbeiteten Datei weiterzuarbeiten, was den Workflow der schrittweisen Verbesserung spürbar vereinfacht. Ergänzend wurde im Seminar auch Canva genutzt, das über eine KI-gestützte Funktion zur Erstellung interaktiver Inhalte verfügt. Hervorzuheben ist dabei das visuelle Design der erzeugten Applets, das im ersten Vergleich zu ChatGPT und Claude deutlich ansprechender ausfiel. Interessantes Potenzial zeigte sich hier insbesondere für stochastische Inhalte. So ließen sich etwa Applets zum Ziegenproblem, zur Monte-Carlo-Methode und zur Binomialverteilung mit vergleichsweise einfachen Prompts erstellen, deren Umsetzung in GeoGebra einen deutlich höheren Zeitaufwand erfordert hätte und die sich beispielsweise für Demonstrationszwecke im Unterricht eignen.

- Ziegenproblem: ziegenproblem.my.canva.site/c0qqey65xeqn83ta
- Monte Carlo Simulation: ziegenproblem.my.canva.site/monte-carlo-methode
- Binomialverteilung: ziegenproblem.my.canva.site/binomialverteilung

Grenzen und Herausforderungen

Ein ehrlicher Bericht über das Seminar muss auch die Grenzen des Vibe Codings thematisieren, denn auch wir haben im Verlauf des Semesters den klassischen Hype-Zyklus durchlaufen. Die anfängliche Begeisterung wich mit zunehmender Praxis einem nüchterne-

ren Blick auf das, was tatsächlich gut funktioniert und was nicht.

Ein häufig auftretendes Problem lässt sich am besten mit dem Ripple-Effect beschreiben. Ändert man über einen neuen Prompt eine bestimmte Funktion des Applets, verändert das LLM gleichzeitig auch andere, nicht adressierte Bereiche des Codes. Dadurch können in zuvor stabil laufenden Applets plötzlich neue Fehler auftauchen oder mathematische Darstellungen voneinander abweichen. Ein zweites Problem betraf die technische Umsetzung. Die von LLMs erzeugten HTML-Dateien ließen sich auf Windows-Laptops problemlos lokal öffnen, auf iPads hingegen nicht ohne Weiteres. Im Seminar wurde als Lösung der webbasierte Dienst OneCompiler (onecompiler.com) eingesetzt, über den HTML-Code direkt im Browser ausgeführt werden kann, ohne lokale Speicherung. Schließlich zeigte sich das Problem, dass selbst kleinste Änderungen am Code, wie etwa die Korrektur einer fehlerhaften mathematischen Notation wie „x2“ statt „x²“, häufig von den Studierenden nicht direkt vorgenommen werden konnten, sondern erneut den Einsatz der KI erforderten. Wer die grundlegende Syntax von HTML kennt, könnte solche Korrekturen schnell händisch vornehmen. Wer es nicht tut, muss einen neuen Prompt formulieren, hoffen, dass das LLM die richtige Stelle ändert, und anschließend prüfen, ob dabei nichts anderes kaputtgegangen ist. Fawzy et al. (2025) beschreiben dieses Muster als strukturelles Problem des Vibe Codings. Die Qualitätssicherung bleibt häufig aus oder wird unkritisch an die KI delegiert, weil die nötige Kompetenz fehlt, um Ergebnisse eigenständig zu beurteilen. Diese Abhängigkeit von einem System, das man nicht durchschauen kann, macht deutlich, dass ein gewisses Grundverständnis für Code auch beim Vibe Coding hilfreich bleibt.

Vibe Coding als Assessment – eine neue Perspektive

Die größte Überraschung des Seminars war keine technische, sondern eine didaktische: Im Laufe der Sitzungen kristallisierte sich heraus, dass der Prozess des Vibe Codings selbst ein äußerst aufschlussreiches Diagnoseinstrument darstellt. Wer ein mathematisches Konzept visualisieren soll, muss es zunächst selbst gut durchdacht haben. Welche Größen sind variabel? Welche Darstellungsform ist mathematisch korrekt? Welche Interaktionen sollen möglich sein? Diese Fragen können nicht durch generische Prompts beantwortet werden, sondern verlangen fachliches Verständnis.

Darüber hinaus wird im Vibe Coding auch die KI-Kompetenz der Studierenden sichtbar. Wer Prompts

formuliert, die den Kontext der Aufgabe mitliefern, also z. B. die Zielgruppe, das mathematische Thema, gewünschte Darstellungsweisen oder didaktische Schritte, erhält qualitativ hochwertigere Ergebnisse als jemand, der nur einen Satz eingibt. Der Vibe-Coding-Prozess wird damit zu einer Form des project-based assessment, das sowohl fachliches als auch KI-bezogenes Kompetenzerleben ermöglicht.

Diese Erkenntnis hat die ursprüngliche Projektidee verschoben. Die Ausgangsidee war es, dass Studierende Materialien erstellen und diese dann mit Schulklassen erproben. Die neue Perspektive versteht die Materialerstellung selbst als Aufgabe. Vibe Coding wird zum neuen Lerngegenstand sowohl für Studierende als auch, in einem nächsten Schritt, für Schülerinnen und Schüler. Die Studierenden, die im Laufe des Semesters Expertise im Vibe Coding aufgebaut haben, können diese Expertise unmittelbar an Schülerinnen und Schüler weitergeben. Die SuS werden selbst kreativ: Sie visualisieren ein mathematisches Konzept, das sie gerade lernen, mithilfe von KI und müssen dafür sowohl das Konzept verstehen als auch darüber kommunizieren. Das eröffnet eine Lernform, die weit über das Bearbeiten mathematischer Aufgaben hinausgeht. Es geht um projektbasiertes, kreatives Arbeiten an mathematischen Ideen, verbunden mit dem kompetenten Umgang mit KI. Diese Verschiebung von der Produkt- hin zur Prozessorientierung deckt sich mit den Handlungsempfehlungen der KMK (2024), die im Sinne eines formativen Assessments empfehlen, Leistungsbewertung „weg von der Produkt- hin zur Prozessorientierung zu lenken, wobei der Reflexion durch die Lernenden eine entscheidende Rolle zugesprochen wird“ (KMK, 2024, S. 7).

Literatur

- Bastani, H., Bastani, O., Sungu, A., Ge, H., Kabakci, O., & Mariman, R. (2025). Generative AI can harm learning. *Proceedings of the National Academy of Sciences*, 122(26). DOI: [10.1073/pnas.2422633122](https://doi.org/10.1073/pnas.2422633122)
- Dilling, F. (2026). Programmieren mit KI für den Mathematikunterricht. Kann Vibe Coding die Entwicklung digitaler Unterrichtsressourcen revolutionieren? *Mitteilungen der Gesellschaft für Didaktik der Mathematik*, 120, 13–19.
- Fan, Y., Tang, L., Le, H., Shen, K., Tan, S., Zhao, Y., Shen, Y., Li, X. & Gašević, D. (2025). Beware of metacognitive laziness: Effects of generative artificial intelligence on learning motivation, processes, and performance. *British Journal of Educational Technology*, 56, 489–530. DOI: [10.1111/bjet.13544](https://doi.org/10.1111/bjet.13544)
- Fawzy, A., Tahir, A. & Blincoe, K. (2025). Vibe coding in practice: Motivations, challenges, and a future outlook – a grey literature review. [arXiv:2510.00328](https://arxiv.org/abs/2510.00328)

- Katter, V. & Huget, J. (2026) Förderung digitaler Kompetenzen durch LLMs – Darstellungswechsel quadratischer Funktionen mit ChatGPT. *mathematica didactica* [im Druck]
- KMK. (2024). Handlungsempfehlung für die Bildungsverwaltung zum Umgang mit Künstlicher Intelligenz in schulischen Bildungsprozessen. Beschluss der Bildungsministerkonferenz vom 10. 10. 2024. Sekretariat der Kultusministerkonferenz.
- Laun, M., Katter, V., Salle, A., & Wolff, F. (2026). Human tutors outperform ChatGPT in teaching mathematical modeling tasks, yet self-concept matters. [Manuskript eingereicht].
- Mertens, C. (2025). Chancengerechtigkeit beim Einsatz Künstlicher Intelligenz (KI) in sonderpädagogischen und inklusiven Schulsettings. *Zeitschrift für Heilpädagogik*, 76.
- Nuxoll, F. (2024). Skill Skipping. Was Lehrpersonen tun können, damit KI nicht Lernprozesse verhindert. *Lehren & Lernen*, 50(6), 11–12. DOI: [10.25656/01:32580](https://doi.org/10.25656/01:32580)
- PuenteDura, R. (2014). *Building transformation: An introduction to the SAMR model*. Online verfügbar unter: www.hippasus.com/rpweblog/archives/2014/08/22/BuildingTransformation_AnIntroductionToSAMR.pdf (Zugriff: 26. 5. 2026).
- Weßels, D. & Maibaum, M. (2026). Vom Deskillung zum Newskilling mit KI. *Forschung & Lehre*, 2. 2. 2026. Online verfügbar unter: www.forschung-und-lehre.de/lehre/vom-deskillung-zum-newskilling-mit-ki-7512 (Zugriff: 26. 5. 2026).
- Walkington, C., Pando, M., Lipsmeyer, L. L., Beauchamp, T., Sager, M. & Milton, S. (2025). Middle school girls using generative AI to engage in mathematical problem-solving. *Mathematical Thinking and Learning*, 1–22. DOI: [10.1080/10986065.2025.2542724](https://doi.org/10.1080/10986065.2025.2542724)
- Valentin Katter, Universität Bielefeld
vkatter@uni-bielefeld.de
- Nicolás Alejandro Alarcón Relmucao, Universität Bielefeld
nicolas.alarcon@uni-bielefeld.de