

Innovative Lehrassistenz

Effektive Nutzung von KI-Tutorsystemen zur Unterstützung von Lehrenden

Handreichung zum Workshop am 18. September 2026

Basierend auf dem Workshop von Dr. Michael Kallweit¹

KI verstehen, Lernprozesse gestalten und eigene Anwendungen erproben: mit Grundlagen, Beispielprompts und Arbeitsaufträgen für die Hochschullehre.

Das vielfältige Angebot an KI-bezogenen Qualifizierungs -und Unterstützungsmaßnahmen ist eine Kooperation des Netzwerks Landeseinrichtungen für digitale Hochschullehre (NeL) sowie assoziierter Partner und der Stiftung Innovation in der Hochschullehre. Weitere Veranstaltungen finden Sie in der Übersicht [berlinweit](#) und [bundesweit](#)).

Veranstaltet von:



**Im Rahmen eines Projektes
von:**



**Netzwerk Landeseinrichtungen
für digitale Hochschullehre**

Gefördert von:



**Stiftung
Innovation in der
Hochschullehre**

Diese Handreichung steht wie die Ausgangsfassung unter der Lizenz

[Creative Commons Namensnennung – Weitergabe unter gleichen Bedingungen 4.0 International \(CC BY-SA 4.0\)](#).

Inhaltlicher Bezug: Workshop vom 18.09.2026. Redaktion und Quellenprüfung: 25.09.2026.

¹michael.kallweit@ruhr-uni-bochum.de

Inhaltsverzeichnis

1	Einleitung	4
1.1	Ziele und Orientierung	4
1.2	Materialien und Aktualität	4
2	KI und Studium: Unterstützung mit Lernwirkung	4
2.1	Von der Zusammenfassung zur aktiven Auseinandersetzung	4
2.2	KI in Prüfungen	5
3	Grundlagen der Künstlichen Intelligenz	6
3.1	KI, maschinelles Lernen und generative Modelle	6
3.2	Sprachmodell und Anwendung unterscheiden	6
3.3	Transformer: eine knappe Erklärung	6
4	Large Language Models: die wichtigen Stellschrauben	7
4.1	Token und Embeddings	7
4.2	Kontextlänge und Gesprächsverlauf	7
4.3	Modellgröße, Open Weights und Quantisierung	8
4.4	Sampling: warum Antworten variieren	8
5	Prompting: Aufgaben klar beschreiben	9
5.1	Ein belastbares Grundgerüst	9
5.2	Beispiele und Teilaufgaben	9
5.3	Reasoning und emotionales Prompting	10
5.4	Systemprompts und Prompt Injection	10
6	Zugänge: Chatoberfläche, lokale Modelle und API	10
6.1	Den Zugang vom Einsatzzweck her wählen	10
6.2	Chat AI in der Academic Cloud	11
6.3	GPT4All und LM Studio	11
6.4	Zugang per API	12
7	KI-Tutoren für die eigene Lehre	12
7.1	Didaktische Regeln vor technischer Umsetzung	12
7.2	Fertige Lernmodi vergleichen	13
7.3	Zwei Beispiele für Tutor-Prompts	13
7.4	Eigene Personas in Chat AI	13
7.5	Persona, Skill und Plugin unterscheiden	14
8	Eigenes Material einbinden: Retrieval-Augmented Generation	14
8.1	Wie RAG funktioniert	14
8.2	Zwei Wege zur Umsetzung	15
8.3	Dokumentqualität ist entscheidend	16
9	Anwendungsbeispiele für Lehrende	16
9.1	Lehrassistenz entlang einer Veranstaltung	16
9.2	Individuelles Feedback: CAS und KI kombinieren	16
9.3	KI-Tutoren in interaktiven Skripten	17
9.4	Lernmaterialien individualisieren	18
9.5	Arbeitsblätter und Dokumente erstellen	19
9.6	Interaktive Webseiten und Simulationen	19
9.7	Audio- und Videoüberblicke aus Kursmaterial	19

10 KI als handelnder Assistent: Agenten	20
10.1 Vom Dialog zum Arbeitsablauf	20
10.2 Beispiele und Einordnung	20
10.3 Aufträge mit klaren Grenzen	20
10.4 MCP als Begriff zur Einordnung	21
11 Verantwortlicher Einsatz: Daten, Transparenz und Kontrolle	21
11.1 Datenflüsse verstehen	21
11.2 Rechtlicher Rahmen	21
11.3 Fachliche und didaktische Verantwortung	22
12 Weiterarbeiten in der eigenen Lehre	22
12.1 Eine Vorlage für den nächsten Versuch	22
12.2 Links zum Weiterarbeiten	23

1 Einleitung

Generative KI kann Lehrende bei der Vorbereitung von Veranstaltungen, beim Entwerfen von Aufgaben und bei der Rückmeldung an Studierende unterstützen. Ihr didaktischer Nutzen entsteht dort, wo sie die fachliche Auseinandersetzung fördert: durch passende Fragen, nachvollziehbare Hinweise und Gelegenheiten, eigene Lösungswege zu prüfen.

Diese Handreichung begleitet den Workshop „Innovative Lehrassistenz“ vom 18. September 2026. Sie erläutert die dort vorgestellten Konzepte und bietet Anregungen zum eigenständigen Weiterarbeiten. Die Beispiele stammen häufig aus der Mathematik; die Prinzipien lassen sich auf andere Fächer übertragen. Ergänzende Beispielprompts sind als Vorlagen gedacht und müssen an Fach, Zielgruppe und Lernziel angepasst werden.

1.1 Ziele und Orientierung

Nach der Lektüre sollen Sie einschätzen können, wie Sprachmodelle Antworten erzeugen, welche Grenzen sie haben und wie Sie einen geeigneten Zugang auswählen. Sie lernen, Anfragen gezielt zu formulieren, einen Tutor zu konfigurieren, eigenes Material über RAG einzubinden und KI-Feedback sowie agentische Arbeitsabläufe kritisch zu erproben.

Die Kapitel folgen dem Weg vom Lernen mit KI über die technischen Grundlagen zur eigenen Lehranwendung. Sie können die Übungen einzeln bearbeiten. Beginnen Sie mit einer kleinen Aufgabe aus Ihrer Lehrveranstaltung, deren Qualität Sie selbst beurteilen können.

Für die Praxis

Halten Sie bei Ihren Versuchen vier Dinge fest: Lernziel, verwendetes Modell bzw. Werkzeug, Prompt und Beobachtung. Eine überzeugende Einzelantwort ist ein guter Ausgangspunkt, aber noch kein Nachweis für zuverlässige Unterstützung.

1.2 Materialien und Aktualität

Der Moodle-Kurs zum Workshop ist unter <https://isis.tu-berlin.de/course/view.php?id=49944> erreichbar; der Zugriff hängt von Ihrer Kursberechtigung ab. Zu den Workshop-Materialien gehören außerdem die Quarto-Beispiele `math-exercise` und `pyodide-interaktiv`.

Produktnamen, Modellangebote und Zugangsbedingungen ändern sich schnell. Die Handreichung erläutert deshalb vor allem Auswahlkriterien und Arbeitsweisen. Verlinkte Dokumentationen helfen, die verfügbaren Funktionen vor einem Einsatz zu prüfen. Bei Forschungsbeispielen wird zwischen Illustration, laufendem Vorhaben und gesichertem Befund unterschieden.

2 KI und Studium: Unterstützung mit Lernwirkung

2.1 Von der Zusammenfassung zur aktiven Auseinandersetzung

Studierende können KI nutzen, um Vorlesungsinhalte zusammenzufassen, Texte in Infografiken zu überführen, Literatur zu erschließen oder Karteikarten zu erstellen. Beispiele dafür sind NotebookLM/Gemini Notebook, Infography, Elicit, STORM und Jungle AI. Diese Beispiele stehen für unterschiedliche Arbeitsweisen. Einstiege sind <https://notebooklm.google.com>, <https://app.infography.in>, <https://elicit.com>, <https://storm.genie.stanford.edu> und <https://jungleai.com>.

Eine gut lesbare Zusammenfassung kann Orientierung geben. Sie kann aber auch das Gefühl erzeugen, einen Inhalt bereits zu beherrschen, obwohl man ihn noch nicht selbst erklären oder anwenden kann. Dieser Effekt wird als „Illusion of competence“ bezeichnet. Entscheidend ist deshalb, welche Tätigkeit die Studierenden anschließend selbst übernehmen.

KI-Ergebnis	Lernförderlicher Anschlussauftrag
Zusammenfassung	Ohne Vorlage erklären; anschließend Auslassungen und Fehler vergleichen.
Infografik oder Mindmap Karteikarten	Beziehungen begründen und an den Originalquellen überprüfen. Erst selbst antworten, dann Feedback lesen; unsichere Karten erneut bearbeiten.
Audio-/Videoüberblick	Eine These notieren, Belege im Material suchen und eine Transferfrage lösen.
Literaturübersicht	Originaltexte lesen und prüfen, ob die genannten Aussagen tatsächlich belegt sind.

Übung: Zusammenfassen und überprüfen

Wählen Sie einen kurzen Abschnitt aus Ihrem Lehrmaterial. Lassen Sie ihn zusammenfassen und drei Verständnisfragen dazu erstellen. Prüfen Sie zuerst die fachliche Richtigkeit. Beantworten Sie dann die Fragen ohne die Zusammenfassung. Welche Fragen verlangen lediglich Wiedererkennen, welche eine eigene Begründung?

Auch visuell überzeugende Ergebnisse können Fehler enthalten. Kontrollieren Sie bei Infografiken besonders Beschriftungen, Einheiten, Reihenfolgen und die dargestellten Beziehungen. Eine schöne Darstellung ist kein Qualitätsnachweis.

2.2 KI in Prüfungen

Ein Praxisbeispiel der Hochschule München zeigt einen anderen Weg: KI wird als zugelassenes Hilfsmittel in eine technisch kontrollierte Prüfung eingebunden. Im Mittelpunkt stehen Modellierungsverständnis, die kritische Bewertung von Ergebnissen sowie die Fähigkeit, KI-Ausgaben zu steuern und zu korrigieren. Berichtet wird zugleich von Entlastung bei Routinearbeiten und zusätzlicher Belastung durch parallele Aufgabenbearbeitung und KI-Nutzung.

Es handelt sich um ein Fallbeispiel, keine allgemeine Aussage über alle Prüfungen. Ein Vortrag zum Projekt bietet den Ausgangspunkt zur Vertiefung.²

Für ein eigenes Konzept sind insbesondere folgende Entscheidungen nötig:

- **Prüfungsziel:** Welche Kompetenz soll trotz oder gerade mit KI sichtbar werden?
- **Eigenleistung:** Welche Begründungen, Entscheidungen und Korrekturen sollen Studierende dokumentieren?
- **Bedingungen:** Wie werden Zugang, Kosten, Hilfsmittel und ein Vorgehen bei Ausfällen geregelt?
- **Bewertung:** Wie wird die Qualität des eigenen Vorgehens gegenüber der Qualität eines KI-Ergebnisses gewichtet?

²Vortrag zum Beispiel der Hochschule München: <https://www.youtube.com/watch?v=mx8Y0o0vISo>.

Vereinbaren Sie die Umsetzung mit den zuständigen Stellen Ihrer Hochschule. Regeln aus einem einzelnen Hochschulbeispiel lassen sich nicht ohne Prüfung der eigenen Prüfungsordnung übertragen. Variierende Modellausgaben sind zudem eine Herausforderung für vergleichbare Bedingungen.

Für die Praxis

Lassen Sie Studierende eine fehlerhafte KI-Lösung fachlich kommentieren und verbessern. Damit wird der Umgang mit KI selbst zum sichtbaren Teil der Leistung. Der Bewertungsmaßstab sollte bereits vor der Aufgabe feststehen.

3 Grundlagen der Künstlichen Intelligenz

3.1 KI, maschinelles Lernen und generative Modelle

Künstliche Intelligenz bezeichnet ein Forschungs- und Anwendungsgebiet, in dem Computersysteme Aufgaben wie Mustererkennung, Sprachverarbeitung und Problemlösen bearbeiten. Die Unterscheidung zwischen spezialisierter bzw. „schwacher“ KI und hypothetischer „starker“ KI hilft zur Orientierung; für die Lehre ist vor allem wichtig, was ein konkretes System unter konkreten Bedingungen leisten kann.

Maschinelles Lernen entwickelt Modelle anhand von Beispieldaten. Im Training werden Parameter angepasst; mit getrennten Testdaten wird untersucht, ob das Modell auf neue Fälle überträgt. **Generative KI** erzeugt auf dieser Grundlage neue Inhalte, beispielsweise Text, Bilder, Audio oder Code.

Leistungen in Benchmarks oder Intelligenztests sind kein unmittelbares Maß für die Qualität eines Tutors. Testformat, mögliche Überschneidungen mit Trainingsdaten und die gewählten Aufgaben beeinflussen das Ergebnis. Ein Modell muss sich an den tatsächlichen Anforderungen Ihrer Lehrsituation bewähren.

3.2 Sprachmodell und Anwendung unterscheiden

Ein Sprachmodell berechnet auf Grundlage seines Kontexts Wahrscheinlichkeiten für eine Fortsetzung. Eine Chat-Anwendung organisiert dagegen zusätzlich den Gesprächsverlauf, die Bedienoberfläche und gegebenenfalls Dateiverarbeitung, Suche oder Codeausführung. Aus diesen Bausteinen entsteht das beobachtbare Verhalten.

Diese Unterscheidung verhindert typische Missverständnisse: Ein Modell ist keine verlässliche Nachschlagedatenbank. In seinen Parametern sind gelernte Muster und auch Fakten repräsentiert, aber keine automatisch überprüfbaren Quellenbelege. Eine Anwendung kann externe Informationen abrufen oder Erinnerungen speichern; solche Funktionen gehören zur jeweiligen Systemgestaltung.

3.3 Transformer: eine knappe Erklärung

Viele Sprachmodelle basieren auf der Transformer-Architektur. Text wird zunächst in Token zerlegt und numerisch dargestellt. Attention-Mechanismen verknüpfen Informationen verschiedener Positionen im Kontext; weitere Netzwerkschichten verarbeiten diese Darstellungen. Bei

der autoregressiven Texterzeugung wird wiederholt ein nächstes Token bestimmt und an den bisherigen Text angefügt.

Das Training kann viele Positionen parallel verarbeiten. Daraus folgt nicht, dass ein Modell bei der Ausgabe die gesamte Antwort gleichzeitig erzeugt. Ebenso wenig bedeutet eine flüssige Formulierung, dass eine Aussage sachlich richtig ist.

Genau hinsehen

LLMs können fachlich anspruchsvolle Aufgaben lösen und zugleich einfache Fehler machen. Sie können Vorurteile reproduzieren, Quellen erfinden oder auf irreführende Eingaben eingehen. Bewerten Sie das konkrete Ergebnis, nicht den selbstsicheren Ton.

4 Large Language Models: die wichtigen Stellschrauben

4.1 Token und Embeddings

Token sind die Verarbeitungseinheiten eines Sprachmodells. Sie können Wörter, Wortteile, Zeichen oder andere Einheiten repräsentieren. Die Zerlegung ist modellabhängig; eine bestimmte Anzahl von Wörtern entspricht deshalb keiner festen Tokenzahl. Auch Zahlen und mathematische Ausdrücke werden nicht zwingend Zeichen für Zeichen zerlegt.

Embeddings stellen Inhalte als Zahlenvektoren dar. Dabei sind die internen Token-Embeddings eines Sprachmodells von den Dokument-Embeddings für eine semantische Suche zu unterscheiden. Letztere können zum Beispiel Textabschnitte so abbilden, dass ähnliche Inhalte über ein Ähnlichkeitsmaß gefunden werden. Das wird später bei RAG genutzt.

4.2 Kontextlänge und Gesprächsverlauf

Die Kontextlänge begrenzt, wie viele Token ein Modell bei einer Anfrage berücksichtigen kann. Systemanweisungen, Gesprächsverlauf, eingefügte Dokumentpassagen und die Ausgabe beanspruchen dieses Budget; zusätzlich können anwendungsspezifische Grenzen gelten.

Eine Chatoberfläche muss den relevanten Verlauf erneut bereitstellen. Wird er zu lang, kann die Anwendung ältere Teile weglassen, zusammenfassen oder die Anfrage ablehnen. Auch eine automatische Zusammenfassung kann wichtige Einzelheiten verlieren.

Genau hinsehen

Ein großes Kontextfenster garantiert nicht, dass sämtliche Informationen zuverlässig genutzt werden. RULER und sogenannte Needle-in-a-Haystack-Tests untersuchen unterschiedliche Aspekte der Verarbeitung langer Kontexte. Erfolgreiches Auffinden einer einzelnen versteckten Information belegt noch keine sichere Synthese eines ganzen Skripts.^a

^aHsieh et al.: RULER (2024): <https://arxiv.org/abs/2404.06654>.

Statt möglichst viel Material anzuhängen, wählen Sie relevante Passagen aus. Geben Sie Kapitel oder Seiten an und lassen Sie Belege nennen. Bei langen Arbeitsprozessen lohnt eine überprüfte Zwischenzusammenfassung mit Lernziel, Begriffen und offenen Punkten. Prüfen Sie die nutzbare Kontextlänge beim konkreten Dienst und Modell.

4.3 Modellgröße, Open Weights und Quantisierung

Parameter sind im Training gelernte Zahlenwerte. Mehr Parameter können zusätzliche Kapazität ermöglichen; Qualität hängt jedoch auch von Trainingsdaten, Training, Aufgabe und Ausführungsbedingungen ab. Bei Mixture-of-Experts-Modellen wird pro Token nur ein Teil der Parameter aktiviert. Gesamtgröße und aktive Parameterzahl sind deshalb unterschiedliche Angaben.

Open Weights bedeutet, dass Modellgewichte verfügbar sind. Daraus folgt nicht automatisch, dass Trainingsdaten und Trainingscode offenliegen oder jede Nutzung erlaubt ist. Die Lizenz muss separat betrachtet werden.

Quantisierung speichert Gewichte mit geringerer numerischer Präzision und senkt dadurch den Speicherbedarf. Je nach Modell und Verfahren kann die Qualität abnehmen. Zusätzlich zu den Gewichten benötigen insbesondere lange Kontexte Speicher. Ob ein Modell lokal sinnvoll läuft, zeigt sich daher an verfügbarem RAM bzw. Grafikspeicher, Antwortgeschwindigkeit und Qualität bei Ihren Aufgaben.

4.4 Sampling: warum Antworten variieren

Beim Sampling wird ein nächstes Token aus einer Wahrscheinlichkeitsverteilung gewählt. Häufige Einstellungen sind:

Temperatur

Skaliert die Werte vor der Wahrscheinlichkeitsberechnung. Niedrige Werte konzentrieren die Auswahl stärker auf wahrscheinliche Token; höhere verteilen sie breiter.

Top- k

Begrenzt die Auswahl auf die k wahrscheinlichsten Token.

Top- p

Beschränkt die Auswahl auf eine Menge wahrscheinlicher Token, deren summierte Wahrscheinlichkeit den Schwellenwert p erreicht.

Wiederholungsstrafe

Verringert modell- bzw. implementierungsabhängig die Neigung, bereits verwendete Token zu wiederholen.

Nicht jedes Modell bietet alle Einstellungen an. Eine niedrige Temperatur ist kein Wahrheitsregler und garantiert auch keine vollständig identischen Ergebnisse.

Übung: Modelle und Einstellungen vergleichen

Wählen Sie eine fachliche Frage mit bekannter Antwort und eine offene Entwurfsaufgabe. Stellen Sie beide mehreren Modellen, etwa in Chat AI. Variieren Sie anschließend bei einem Modell nur einen Sampling-Parameter. Vergleichen Sie fachliche Richtigkeit, Variation, Verständlichkeit und Antwortzeit. Wiederholen Sie die Anfrage, um Zufallseffekte zu erkennen.

5 Prompting: Aufgaben klar beschreiben

5.1 Ein belastbares Grundgerüst

Ein Prompt ist eine Eingabe an ein Sprachmodell. Prompt Engineering bezeichnet die gezielte Gestaltung von Anweisungen, Kontext und Beispielen. Für Lehranwendungen hilft folgende Struktur:

1. **Aufgabe:** Was soll entstehen oder untersucht werden?
2. **Zielgruppe und Lernziel:** Für wen ist die Ausgabe gedacht, welches Vorwissen darf sie voraussetzen?
3. **Material und Grenzen:** Welche Quellen, Begriffe und Methoden sind maßgeblich?
4. **Ausgabe:** In welchem Format, Umfang und Stil wird das Ergebnis benötigt?
5. **Prüfkriterien:** Woran erkennen Sie, ob der Entwurf geeignet ist?

Klare Anweisungen, relevante Referenztexte und Beispiele gehören auch zu den Strategien der offiziellen Prompting-Dokumentation.³

Beispiel: eine Übung vorbereiten

Entwurf für Studierende im ersten Semester eine Übung zur Interpretation der Ableitung. Lernziel: lokale Änderungsraten aus einem Kontext erklären und ihre Einheit bestimmen. Vorausgesetzt werden lineare Funktionen und der Ableitungsbegriff. Erstelle drei Aufgaben mit steigendem Anspruch sowie getrennte Lösungshinweise für Lehrende. Nenne zu jeder Aufgabe einen typischen Denkfehler. Verwende realistische Größenordnungen und kennzeichne Annahmen.

Eine Rolle wie „Du unterstützt als Tutor für Analysis“ kann Stil und Perspektive lenken. Sie verleiht dem Modell jedoch keine geprüfte Expertise. Fachliche Vorgaben und die Prüfung des Ergebnisses bleiben erforderlich.

5.2 Beispiele und Teilaufgaben

Bei **Zero-shot Prompting** geben Sie eine Aufgabe ohne Beispielpaare vor. Bei **Few-shot Prompting** ergänzen Sie Beispiele für die gewünschte Beziehung zwischen Eingabe und Ausgabe, etwa eine Studierendenantwort mit passender Rückmeldung. Wählen Sie Beispiele, die auch schwierige Fälle abdecken, und halten Sie das Format konsistent.

Komplexe Aufgaben lassen sich in überprüfbare Schritte zerlegen: zunächst Lernziele klären, dann Aufgaben entwerfen, anschließend Lösungen erstellen und die Übereinstimmung prüfen. Das erlaubt Ihnen, früh einzugreifen. Eine zweite KI-Prüfung kann Fehler aufdecken, ersetzt aber keine fachliche Kontrolle.

³OpenAI: Prompt engineering: <https://developers.openai.com/api/docs/guides/prompt-engineering>.

5.3 Reasoning und emotionales Prompting

Chain-of-Thought-Ansätze und Reasoning-Modelle liefern ausführliche Lösungswege, die bei der Prüfung einer Antwort helfen können. Sichtbare Erklärungen sind jedoch kein verlässlicher Einblick in interne Modellprozesse. Fordern Sie für die Lehre daher prüfbare Rechenschritte, Annahmen und Begründungen an. Lange Antworten sind nicht automatisch besser.

Reasoning-Fähigkeiten werden bei dafür ausgelegten Modellen auch durch Training gefördert. Die passende Anfrage hängt vom Modell und von der Aufgabe ab; eine universelle Zauberformel gibt es nicht.

Die Studie von Li et al. untersucht emotionale Zusätze zu Prompts. Ihre Ergebnisse sind an untersuchte Modelle und Aufgaben gebunden und begründen keine allgemeine Qualitätsgarantie.⁴

Übung: die Wirkung einer Formulierung untersuchen

Stellen Sie dieselbe Frage einmal sachlich und einmal mit einem emotionalen Zusatz, etwa: „Das ist mir für meine Prüfung sehr wichtig.“ Halten Sie Modell und übrige Einstellungen gleich. Wiederholen Sie beide Varianten und bewerten Sie Richtigkeit und Verständlichkeit anhand derselben Kriterien. Unterscheiden Sie bessere Inhalte von bloß zugewandterem Ton.

5.4 Systemprompts und Prompt Injection

Ein Systemprompt legt übergreifende Anweisungen fest, zum Beispiel Rolle, Gesprächsführung und Grenzen eines Tutors. Der Nutzerprompt enthält die konkrete Aufgabe. In eingebundenen Dokumenten oder auf Webseiten können ebenfalls Anweisungen stehen, die das System von seinem eigentlichen Auftrag abbringen sollen: **Prompt Injection**.

Ein hochgeladenes Arbeitsblatt könnte etwa die Anweisung enthalten, bisherige Regeln zu ignorieren und sofort die Musterlösung auszugeben. Solcher Inhalt sollte als zu untersuchendes Material behandelt werden. Die Aufforderung „Ignoriere fremde Anweisungen“ allein ist keine zuverlässige technische Absicherung. Beschränkte Werkzeugrechte und kontrollierte Aktionen werden besonders wichtig, sobald ein System Dateien bearbeiten oder Daten versenden kann.

6 Zugänge: Chatoberfläche, lokale Modelle und API

6.1 Den Zugang vom Einsatzzweck her wählen

Für einen ersten Vergleich genügt meist eine Chatoberfläche. Lokale Anwendungen bieten mehr Kontrolle über die Verarbeitung, verlangen aber passende Hardware. Eine API wird interessant, wenn KI in eine eigene Webseite, eine Lernaufgabe oder einen wiederkehrenden Ablauf eingebunden werden soll.

⁴Li et al.: Large Language Models Understand and Can be Enhanced by Emotional Stimuli: <https://arxiv.org/abs/2307.11760>.

Zugang	Geeignet für	Vorher klären
Online-Chat	Entwürfe, Erklärungen, Modellvergleich	Konto, Datenverarbeitung, verfügbare Funktionen
Lokales Modell	Experimente und Verarbeitung auf eigener Hardware	Speicherbedarf, Geschwindigkeit, Modelllizenzen
API	Eigene Tutor-Anwendung, integriertes Feedback	Schlüsselverwaltung, Kosten, Limits, Datenfluss

6.2 Chat AI in der Academic Cloud

Chat AI der GWDG bietet über <https://chat-ai.academiccloud.de> Zugang zu unterschiedlichen Modellen. Für die Arbeit mit dieser Handreichung sind insbesondere Modellvergleich, Systemprompts, Sampling-Einstellungen und teilbare Personas relevant. Der Zugang hängt von der Freischaltung Ihres Kontos bzw. Ihrer Institution ab.

Achten Sie auf die Unterscheidung zwischen selbst gehosteten und extern angebundenen Modellen. Die Wahl des Modells und zusätzlicher Werkzeuge beeinflusst, wohin Inhalte übermittelt werden. Maßgeblich ist die Datenschutzhinweiseite für den tatsächlich genutzten Dienst.⁵

Wählen Sie Modelle anhand eigener Aufgaben. Ein Vergleich sollte neben fachlicher Richtigkeit auch den Umgang mit Unsicherheit, Sprache, Quellenbelegen, Antwortzeit und didaktischen Vorgaben erfassen. Arena-Vergleiche wie <https://lmarena.ai> können Anregungen liefern; allgemeine Ranglisten ersetzen keine fachbezogene Erprobung.

6.3 GPT4All und LM Studio

GPT4All stellt eine Oberfläche für lokale Modelle bereit. Mit LocalDocs lassen sich Dokumentensammlungen in Gespräche einbeziehen.⁶ **LM Studio** ermöglicht ebenfalls die lokale Nutzung heruntergeladener Modelle und bietet Einstellmöglichkeiten sowie einen lokalen Serverzugang. Nach Bereitstellung der benötigten Modelle können die dafür vorgesehenen lokalen Funktionen offline genutzt werden.⁷

Übung: ein lokales Modell erproben

Installieren Sie eine der beiden Anwendungen und wählen Sie ein Modell passend zu Ihrer Hardware. Testen Sie zunächst eine kurze bekannte Fachfrage. Beobachten Sie Speicherbedarf und Geschwindigkeit, bevor Sie lange Dokumente hinzufügen. Vergleichen Sie anschließend dieselbe Aufgabe mit einem Online-Modell.

Genau hinsehen

Eine lokal installierte Oberfläche bedeutet nicht, dass jeder Verarbeitungsschritt lokal erfolgt. Prüfen Sie Modellanbieter, Embedding-Dienst und aktivierte Werkzeuge. Auch rein lokale Verarbeitung ist keine pauschale Bestätigung der Datenschutzkonformität.

⁵GWDG: Data Privacy Notice für Chat AI: <https://docs.hpc.gwdg.de/services/ai-services/chat-ai/data-privacy/index.html>.

⁶GPT4All: LocalDocs: https://docs.gpt4all.io/gpt4all_desktop/localdocs.html.

⁷LM Studio: Offline Operation: <https://lmstudio.ai/docs/app/offline>.

6.4 Zugang per API

Eine API ist eine Programmierschnittstelle. Ein Programm übermittelt Modellkennung, Eingaben und unterstützte Einstellungen an einen Dienst und verarbeitet dessen Antwort. Das ermöglicht beispielsweise eine Rückmeldung direkt in einer digitalen Aufgabe.

Viele Dienste unterstützen ein an Chat Completions angelehntes Format. Das folgende Beispiel illustriert das Prinzip; Basisadresse, Modellkennung und unterstützte Parameter müssen aus der Dokumentation des gewählten Anbieters übernommen werden.

```
POST /v1/chat/completions
Authorization: Bearer API_KEY
Content-Type: application/json

{
  "model": "MODELLKENNUNG_DES_ANBIETERS",
  "messages": [
    {"role": "system", "content": "Du hilfst als Tutor."},
    {"role": "user", "content": "Was ist eine Ableitung?"}
  ]
}
```

Mögliche Zugänge sind etwa OpenRouter, Google AI Studio, Groq, Cerebras, Mistral und Hugging Face. Ob ein kostenfreies Kontingent verfügbar ist und welche Modelle es umfasst, muss beim jeweiligen Anbieter geprüft werden. Ein Chat-Abonnement ist nicht automatisch ein API-Guthaben. Planen Sie auch Antwortlängen, gleichzeitige Anfragen und ein Verhalten bei ausgeschöpften Limits ein.

Für die Praxis

API-Schlüssel sind Zugangsdaten. Veröffentlichen Sie sie weder in Folien noch im Quelltext einer verteilten HTML-Datei oder in einem Repository. Für einen Lehrbetrieb braucht es ein bewusstes Zugangs- und Kostenkonzept, etwa ein institutionell verwaltetes Gateway.

7 KI-Tutoren für die eigene Lehre

7.1 Didaktische Regeln vor technischer Umsetzung

Ein Tutor soll Studierende dazu anregen, selbst zu denken, Annahmen zu begründen und Fehler zu bearbeiten. Fünf Gestaltungsprinzipien geben dafür Orientierung:

Scaffolding

Hilfe schrittweise anbieten und an den bisherigen Versuch anpassen.

Sokratisches Nachfragen

Mit gezielten Fragen das Verständnis sichtbar machen.

Fehlerdiagnose vor Korrektur

Zuerst den problematischen Schritt untersuchen lassen.

Begründungspflicht

Nach Gründen für Annahmen, Schätzungen und Lösungswege fragen.

Eine Frage auf einmal

Den Dialog übersichtlich halten und Antworten abwarten.

Vorlage für einen Tutor-Systemprompt

Du begleitest Studierende in [Fach, Semester] beim Lernziel [Lernziel]. Verwende die Begriffe und Methoden aus [Kursmaterial]. Frage zunächst nach dem bisherigen Ansatz. Gib nicht sofort eine vollständige Lösung. Stelle jeweils nur eine kurze Frage und warte die Antwort ab. Fordere Begründungen ein. Bei einem Fehler hilf zunächst, den problematischen Schritt zu finden. Gib auf Wunsch einen Hinweis; werde erst bei weiterem Unterstützungsbedarf konkreter. Kennzeichne Unsicherheit und erfinde keine Quellen. Beginne mit der Frage, was bereits versucht wurde.

Dieser Prompt beschreibt ein gewünschtes Verhalten. Ob das Modell es einhält, muss in mehreren Dialogen getestet werden: mit korrekten, falschen und unvollständigen Antworten sowie der Aufforderung, sofort die Lösung zu verraten.

7.2 Fertige Lernmodi vergleichen

ChatGPT Study Mode, Gemini Guided Learning und Claude Learning Mode sind Beispiele für vorkonfigurierte Lernbegleitung. Bei diesen Modi stehen Rückfragen, schrittweise Unterstützung und eine Anpassung an das Verständnis im Vordergrund. Guided Learning und Claude Learning Mode sind in den jeweiligen Anbieterbeschreibungen erläutert.⁸⁹

Testen Sie einen verfügbaren Lernmodus mit einer Frage aus Ihrem Fach. Prüfen Sie, ob er Vorwissen erfragt, eigene Beiträge abwartet und Hinweise angemessen dosiert. Zugänglichkeit und Bezeichnungen hängen von Plattform und Konto ab. Ein Lernmodus ist keine Garantie dafür, dass Lösungen zurückgehalten oder fachliche Fehler vermieden werden.

7.3 Zwei Beispiele für Tutor-Prompts

Der **Modellierungsschatbot von Kallweit und Rolka (2024)** kombiniert fachliche Informationen zur Modellierungskompetenz, eine Beispielaufgabe mit möglicher Lösung und detaillierte Gesprächsregeln. Studierende sollen etwa eigene Annahmen entwickeln, unrealistische Schätzungen hinterfragen und Fehler selbst finden. Der Ansatz verdeutlicht, wie eng fachlicher Kontext und Gesprächsführung zusammengehören.¹⁰

CALCBLUEBOT ist eine Promptvorlage von Robert Ghrist, die auf Materialien des Calculus-BLUE-Projekts zur mehrdimensionalen Analysis ausgerichtet ist. Der umfangreiche Prompt illustriert die mögliche Detailtiefe. Entscheidend ist jedoch, ob Regeln und Material zur eigenen Lehrsituation passen, nicht die Länge des Prompts.¹¹

7.4 Eigene Personas in Chat AI

Eine Persona bündelt wiederverwendbare Einstellungen für einen bestimmten Einsatzzweck. In Chat AI können dazu Systemprompt, Modellkonfiguration und gegebenenfalls Gesprächsnachrichten gehören. Der JSON-Export macht solche Konfigurationen teilbar. Die aktuellen Import- und Freigabemöglichkeiten beschreibt die GWDG-Dokumentation.¹²

⁸Google: Guided Learning: <https://blog.google/products-and-platforms/products/gemini/guided-learning-google-gemini/>.

⁹Anthropic: Claude for Education: <https://www.anthropic.com/news/introducing-claude-for-education>.

¹⁰Kallweit und Rolka: Beitrag zu generativer KI bei Modellierungsaufgaben: <https://doi.org/10.17877/DE290R-24628>.

¹¹Robert Ghrist: Calculus BLUE und CALCBLUEBOT: <https://github.com/robertghrist/calculus>.

¹²GWDG: Personas: <https://docs.hpc.gwdg.de/services/ai-services/chat-ai/features/personas/index.html>.

```
{
  "model-name": "ANZEIGENAME_DES_MODELLS",
  "model": "MODELLKENNUNG",
  "temperature": 0.5,
  "top_p": 0.5,
  "messages": [
    {
      "role": "system",
      "content": "Hilf als Tutor. Stelle eine Frage nach der anderen."
    }
  ]
}
```

Dieses Beispiel ist schematisch. Nutzen Sie für einen tatsächlichen Import vorzugsweise den Export Ihrer aktuellen Oberfläche und passen Sie ihn an; die Platzhalter sind keine gültigen Modellkennungen.

Gruppenübung: eine Persona entwickeln und testen

Vereinbaren Sie Fach, Zielgruppe und Lernziel. Formulieren Sie einen Systemprompt und testen Sie ihn zunächst selbst. Exportieren Sie die Konfiguration und prüfen Sie, ob sie Gesprächsinhalte enthält. Teilen Sie nur die dafür vorgesehene Fassung. Lassen Sie eine andere Gruppe sowohl einen normalen Lerndialog als auch einen Versuch zur direkten Lösungsabfrage durchführen. Überarbeiten Sie die Persona anhand der Beobachtungen.

Eine Persona lässt sich außerdem über eine öffentlich erreichbare JSON-Datei aufrufen, nach dem Schema <https://chat-ai.academiccloud.de/chat?import=...> Prüfen Sie dieses Verhalten mit der aktuellen Oberfläche. Öffentliche Dateien und geteilte Konfigurationen dürfen keine privaten Gespräche oder Zugangsdaten enthalten.

7.5 Persona, Skill und Plugin unterscheiden

Eine **Persona** beschreibt Rolle und Einstellungen eines Assistenten. Ein **Skill** bündelt wiederverwendbare Anweisungen und gegebenenfalls Ressourcen für einen Arbeitsablauf. Ein **Plugin** kann mehrere Erweiterungen und Werkzeuganbindungen paketieren. Die genaue Umsetzung ist plattformabhängig. Als Beispiel erläutert die OpenAI-Dokumentation Plugins mit Skills und Integrationen.¹³

Für einen ersten Tutor genügt häufig eine gut getestete Persona. Zusätzliche Werkzeuge sind sinnvoll, wenn die Aufgabe etwa das Nachschlagen im Kursmaterial oder die Ausführung von Code verlangt.

8 Eigenes Material einbinden: Retrieval-Augmented Generation

8.1 Wie RAG funktioniert

Retrieval-Augmented Generation (RAG) verbindet Informationssuche mit Textgenerierung: Zu einer Frage werden passende Inhalte aus einer Wissensbasis gesucht und dem Sprachmodell als zusätzlicher Kontext bereitgestellt. Häufig wird dafür eine Suche über

¹³OpenAI: Plugins: <https://learn.chatgpt.com/docs/plugins>.

Dokument-Embeddings verwendet; RAG ist aber nicht auf Vektorsuche beschränkt.

Ein typischer Ablauf umfasst fünf Schritte:

1. Dokumente in geeignete Textabschnitte zerlegen und deren Embeddings berechnen.
2. Auch die Anfrage als Suchvektor darstellen.
3. Inhaltlich passende Abschnitte über ein Ähnlichkeitsmaß auswählen.
4. Diese Abschnitte zusammen mit der Frage in den Modellkontext aufnehmen.
5. Eine Antwort erzeugen, die sich auf das bereitgestellte Material stützt.

RAG trainiert das Sprachmodell dabei nicht neu. Es erweitert den Kontext einer Anfrage. Die Qualität hängt sowohl von der Suche als auch von der anschließenden Antwortgenerierung ab.

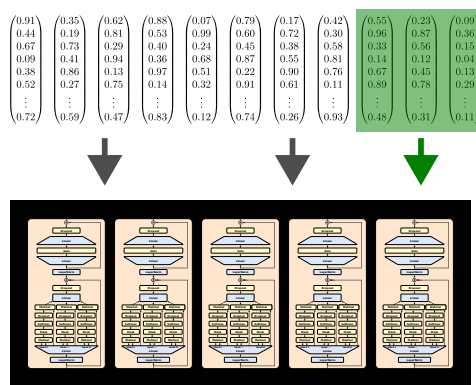


Illustration zur Auswahl ähnlicher Dokumente.

8.2 Zwei Wege zur Umsetzung

Lokal mit GPT4All: Legen Sie eine LocalDocs-Sammlung mit geeigneten Dokumenten an, warten Sie deren Verarbeitung ab und aktivieren Sie die Sammlung im Gespräch. Prüfen Sie die zugrunde liegende Modell- und Embedding-Konfiguration, wenn alle Inhalte auf Ihrem Rechner bleiben sollen.

Serverseitig mit Arcana in Chat AI: Über <https://chat-ai.academiccloud.de/arcanas> lassen sich Dokumentensammlungen erstellen und in Chat AI nutzen. Die GWDG beschreibt Arcana als serverseitiges RAG mit verschlüsselter Speicherung nach der Vorverarbeitung und Freigabelinks, die Kennung und Schlüssel enthalten. Solche Links sind daher wie Zugangsberechtigungen zu behandeln.¹⁴

Die serverseitige Verarbeitung unterscheidet sich von einer lokalen Anwendung. Prüfen Sie vor dem Hochladen, ob die Materialien für diesen Dienst und gegebenenfalls für eine Weitergabe freigegeben sind. Berücksichtigen Sie auch das Modell, das die gefundenen Passagen anschließend verarbeitet.

¹⁴GWDG: Neue Funktion Arcana in Chat AI: <https://gwdg.de/news/2025-02-neue-funktion-arcanas-in-chat-ai/>.

8.3 Dokumentqualität ist entscheidend

Mehrspaltige PDFs, gescannte Seiten, Tabellen und Formeln können bei der Textextraktion beschädigt werden. Eine Formel, deren Exponent verloren geht, kann im Suchindex zwar auffindbar sein, aber fachlich falsche Antworten begünstigen. Gut strukturiertes Markdown mit Überschriften, Quellenangaben und korrekt übertragenen Formeln erleichtert die Weiterverarbeitung.

Hilfreich sind etwa **Nougat** für wissenschaftliche Dokumente und **Marker** zur Konvertierung von PDFs in strukturierte Formate.¹⁵¹⁶ Auch deren Ausgaben müssen stichprobenartig mit dem Original verglichen werden. Bewahren Sie nachvollziehbare Verweise auf Dokument, Kapitel und Seite auf.

Übung: RAG systematisch vergleichen

Nehmen Sie ein bis zwei freigegebene Kursdokumente. Formulieren Sie eine Frage zu einer eindeutig enthaltenen Information, eine Frage, die zwei Abschnitte verbindet, und eine Frage, die das Material nicht beantwortet. Stellen Sie alle drei Fragen mit und ohne Dokumentensammlung. Kontrollieren Sie die Fundstellen: Tragen sie die Antwort? Erkennt das System die fehlende Information? Wiederholen Sie den Versuch gegebenenfalls mit bereinigtem Markdown.

Genau hinsehen

RAG verhindert Halluzinationen nicht vollständig. Die Suche kann ungeeignete Passagen liefern, und das Modell kann diese falsch interpretieren oder ergänzen. Eine Quellenangabe ist erst dann hilfreich, wenn die angegebene Stelle tatsächlich zur Aussage passt.

9 Anwendungsbeispiele für Lehrende

9.1 Lehrassistenz entlang einer Veranstaltung

Vor einer Sitzung kann KI Lernziele präzisieren, Beispiele variieren und Einstiegsfragen entwerfen. Während und nach der Sitzung kann sie Verständnisfragen, Rückmeldungen sowie Reflexions- und Transferaufgaben unterstützen. Die didaktische Entscheidung bleibt bei den Lehrenden: Welche Tätigkeit soll die KI übernehmen, welche sollen die Studierenden selbst ausführen?

Ein ansprechender Kontext und eine interaktive Darstellung können den Einstieg erleichtern. Für den Lernwert sind jedoch passende Anforderungen, die Diagnose typischer Fehler und Gelegenheiten zur eigenen Revision entscheidend.

9.2 Individuelles Feedback: CAS und KI kombinieren

Ein hybrider Ansatz für digitale Mathematikaufgaben verbindet zwei Prüfwege: Ein Computeralgebrasystem (CAS), etwa in STACK, prüft Antworten anhand vorgegebener mathematischer Eigenschaften. Lehrende können zu bekannten Fehlermustern gezielte Rückmeldungen hinterlegen. Für nicht ausreichend erfasste Fehler kann KI ergänzend ein sprachlich erläutertes Feedback erzeugen.

Ein passender Prompt bündelt Aufgabenstellung, Musterlösung, Studierendenantwort und CAS-Analyse. Die KI übernimmt die Rolle einer ergänzenden Rückmeldung, wenn die vorbereitete

¹⁵Nougat: <https://facebookresearch.github.io/nougat/>.

¹⁶Marker: <https://github.com/datalab-to/marker>.

Fehlerdiagnose nicht ausreicht. Eine mathematische Prüfung und eine lernförderliche Erklärung sind unterschiedliche Aufgaben und sollten auch getrennt geprüft werden.

Ergänzende Promptvorlage: Rückmeldung zu einem Lösungsversuch

Aufgabe: [Aufgabenstellung]

Geprüfte Musterlösung: [Lösung]

Antwort der lernenden Person: [Antwort]

CAS-Rückmeldung: [Analyse]

Formuliere eine kurze Rückmeldung zum ersten erkennbaren Problem. Berücksichtige die CAS-Analyse und prüfe sie am Aufgabenkontext. Gib zunächst einen Denkanstoß und eine Rückfrage. Falls die Informationen für eine eindeutige Diagnose nicht genügen, benenne diese Grenze. Gib keine vollständige Lösung aus.

Der Ansatz stammt aus einem Vorhaben von Razeghpour, Rolka, Roelle und Kallweit, dessen Veröffentlichung in Vorbereitung ist. Die bisherigen Beispielfälle illustrieren Möglichkeiten; sie belegen keine allgemeine Zuverlässigkeit. Offen bleiben die fachliche Qualität des Feedbacks und sein Nutzen für die Lernenden.

9.3 KI-Tutoren in interaktiven Skripten

Quarto ist ein Publishing-System, das aus Quelldokumenten unter anderem HTML, PDF und Word-Dokumente erzeugen kann. HTML-Ausgaben lassen sich durch interaktive Inhalte erweitern.¹⁷ Zwei eigene Erweiterungen zeigen, wie sich KI-Tutoren in solche Skripte einbinden lassen; die dazugehörigen Quell- und Beispieldateien liegen in den Workshop-Materialien.

9.3.1 math-exercise: mathematische Eingaben prüfen

Die Erweiterung verbindet Eingabefelder mit einer mathematischen Prüfung über SymPy in Pyodide. Die Prüfung läuft im Browser. Optional kommt KI-Feedback über einen konfigurierten API-Dienst hinzu. Dabei können Aufgabe, Antwort und Lernkontext zur Formulierung einer passenden Rückmeldung verwendet werden.

Die mitgelieferte Fassung sieht vier Hinweisstufen vor: zunächst eine diagnostische Frage, dann einen begrifflichen Hinweis, anschließend eine Verfahrenshilfe und schließlich eine vollständige Lösung. Die Stufe steigt mit weiteren Feedback-Abrufen. Die frühen Stufen halten die Lösung zurück, die vierte erlaubt sie ausdrücklich. Entscheiden Sie vor dem Einsatz, ob diese Freigabe zu Ihrem Lernziel passt.

9.3.2 pyodide-interaktiv: Python im Browser bearbeiten

Diese Erweiterung bietet editierbare und ausführbare Python-Zellen im HTML-Skript. Pyodide führt den Code im Browser aus. Bei Syntax- oder Laufzeitfehlern kann KI-Feedback Hinweise zur Ursache geben. Die mitgelieferte Fassung unterstützt außerdem einen Modus, in dem ein Feedback-Prompt kopiert und in einer Chat-Anwendung verwendet wird.

Genau hinsehen

Browserseitige Codeausführung und KI-Feedback sind getrennte Verarbeitungsschritte. Für Feedback über eine externe API werden die benötigten Inhalte an den konfigurierten Dienst

¹⁷Quarto: Guide: <https://quarto.org/docs/guide/>.

übermittelt. Außerdem können Laufzeitdateien aus dem Netz nachgeladen werden. Eine lokal geöffnete HTML-Datei ist daher nicht automatisch vollständig offline nutzbar.

9.3.3 Die mitgelieferten Beispiele ausprobieren

Im Ordner `math-exercise` finden Sie `examples.qmd`, die gerenderte `examples.html` sowie ein Beispiel zur linearen Abhängigkeit. Im Ordner `pyodide-interaktiv` liegen `demo.qmd` und `demo.html`. Die jeweiligen README-Dateien erläutern Einrichtung und Grenzen. Stellen Sie die Beispiele über einen lokalen oder institutionellen Webserver bereit; insbesondere die Pyodide-Demo benötigt HTTP(S) statt eines bloßen Doppelklicks auf die Datei.

Testen Sie eine richtige, eine typische falsche und eine unerwartete Eingabe. Fordern Sie mehrmals Feedback an und beobachten Sie, wie sich die Hilfe verändert. Prüfen Sie, ob die Hinweise den Denkprozess unterstützen und ob die gewählte Lösungsfreigabe zum Lernziel passt. Für größere Gruppen sind zusätzlich API-Limits, Kosten und Zugänglichkeit relevant.

9.4 Lernmaterialien individualisieren

KI kann eine Basisaufgabe in unterschiedliche Interessenskontexte übertragen. Das Ziel ist ein passender Realitätsbezug bei gleichbleibendem fachlichem Anspruch. Die folgenden Beispiele sind für die Hochschule adaptiert und lehnen sich an Wenzel (2026) an; die zugrunde liegende Arbeit ist eine Untersuchung mit Schülerinnen und Schülern. Ein Wirksamkeitsnachweis für Hochschullehre folgt aus diesen Beispielen nicht.

Beispielsweise lässt sich eine Extremwertaufgabe mit

$$G(x) = -2x^2 + 120x - 800$$

von einem Unternehmenskontext in einen E-Sport-Kontext übertragen. Die Rechnung bleibt: $G'(x) = -4x + 120$, also $x = 30$ und $G(30) = 1000$; wegen $G''(x) = -4$ liegt ein Maximum vor. Trotzdem ist der Kontext eigens zu prüfen: Aus „Gewinn“ darf nicht unbemerkt „Erlös“ werden, wenn die Begriffe wirtschaftlich etwas anderes beschreiben. Auch Bedeutung und zulässige Werte von x müssen passen.

Ein Gegenbeispiel überträgt eine abnehmende Serverrate auf einen Energieverbrauch beim Training. Die mathematische Form kann erhalten bleiben, während Größenordnung und Einheiten unrealistisch werden. Bereits ein Wechsel zwischen Kalorien und Kilokalorien verändert die Bedeutung um den Faktor 1000.

Vorlage: Kontext einer Aufgabe anpassen

Übertrage die folgende Basisaufgabe in einen Kontext zum Interesse [Interesse]. Bewahre Lernziel, mathematische Struktur und Schwierigkeitsgrad. Prüfe Bedeutung der Variablen, Definitionsbereich, Einheiten und Größenordnungen. Wenn eine unveränderte Übertragung fachlich nicht sinnvoll ist, erkläre das, statt einen unplausiblen Kontext zu erfinden. Gib die neue Aufgabe und getrennt einen Änderungsbericht aus. Basisaufgabe: [Text].

Prüfen Sie jede Variante durch eigenes Lösen. Kontrollieren Sie zusätzlich, ob der Text neue Voraussetzungen einführt oder die Aufgabe unbeabsichtigt erleichtert. Interessen sollten möglichst ohne unnötige personenbezogene Angaben berücksichtigt werden.

9.5 Arbeitsblätter und Dokumente erstellen

Ein reines Sprachmodell erzeugt zunächst Text oder Code. KI-Anwendungen mit Dateifunktionen und Codeausführung können daraus auch PDF- oder Office-Dateien erstellen. Für mathematische Materialien bleibt $\text{\LaTeX}$ hilfreich, weil Formeln und Layout gut kontrollierbar sind.

Vorlage: ein Arbeitsblatt

Erstelle ein $\text{\LaTeX}$ -Arbeitsblatt zu Vektorräumen für Studierende im zweiten Semester. Es soll zentrale Definitionen, zwei erläuterte Beispiele und fünf Aufgaben mit steigendem Anspruch enthalten. Trenne Aufgabenblatt und Lösungsteil. Nenne Voraussetzungen und prüfe, dass jede Aufgabe eindeutig formuliert und lösbar ist. Gib vollständigen kompilierbaren Quelltext aus.

Lassen Sie das Dokument kompilieren und kontrollieren Sie sowohl Inhalt als auch Ausgabe: Stimmen Formeln, Seitenumbrüche, Nummerierung und Verweise? Pandoc kann bei der Umwandlung geeigneter Textformate helfen; komplexe Formeln und Layouts müssen nach einer Konvertierung erneut geprüft werden.¹⁸

9.6 Interaktive Webseiten und Simulationen

HTML, CSS und JavaScript ermöglichen kleine Selbsttests und Simulationen. Einfache, eigenständige HTML-Dateien können direkt im Browser geöffnet werden. Anwendungen mit nachgeladenen Ressourcen, APIs oder Web-Workern können einen Webserver und zusätzliche Einrichtung benötigen.

Vorlage: Simulation eines zweifachen Würfelwurfs

Erstelle eine HTML-Seite, die zwei faire sechsseitige Würfel simuliert. Die Anzahl der Wiederholungen soll einstellbar sein. Zeige die Häufigkeiten der Augensummen 2 bis 12 als Histogramm und zum Vergleich die theoretischen Wahrscheinlichkeiten. Ergänze eine Frage dazu, warum die Summe 7 häufiger als 2 vorkommt. Achte auf verständliche Beschriftungen und Bedienung per Tastatur.

Die theoretischen Häufigkeiten der Summen sind proportional zu 1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1 bei 36 gleich wahrscheinlichen Paaren. Das liefert einen unabhängigen Prüfansatz für die Simulation. Testen Sie außerdem leere, negative und sehr große Eingaben. Eine Animation kann motivieren, sollte aber die Interpretation der Verteilung unterstützen.

9.7 Audio- und Videoüberblicke aus Kursmaterial

NotebookLM, in der aktuellen Anbieterhilfe auch Gemini Notebook genannt, erstellt aus hochgeladenen Quellen Audio- und Videoüberblicke. Audio Overviews können Material in ein synthetisches Gespräch überführen; Video Overviews ergänzen visuelle Aufbereitung. Zielgruppe und Fokus sollten bei der Erstellung ausdrücklich benannt werden.^{19,20}

Geben Sie einen Überblick erst nach fachlicher Prüfung weiter. Verbinden Sie ihn mit einem Arbeitsauftrag, beispielsweise: „Welche Annahme wird bei der Herleitung verwendet? Belegen Sie sie im Originalmaterial.“ Stellen Sie für die Nutzung auch eine geeignete textliche Alternative bereit.

¹⁸Pandoc: <https://pandoc.org/>.

¹⁹Google: Audio Overviews: <https://support.google.com/notebooklm/answer/16212820>.

²⁰Google: Video Overviews: <https://support.google.com/notebooklm/answer/16454555>.

10 KI als handelnder Assistent: Agenten

10.1 Vom Dialog zum Arbeitsablauf

Ein Agent verbindet ein Modell mit Werkzeugen und einer Steuerung für mehrschrittige Aufgaben. Er erhält ein Ziel, wählt einen nächsten Schritt, ruft beispielsweise eine Suche oder Dateifunktion auf, verarbeitet das Ergebnis und entscheidet über das weitere Vorgehen. Am Ende sollte ein überprüfbares Arbeitsergebnis stehen.

Für die Lehrvorbereitung könnte ein solcher Ablauf Kursmaterial sichten, Lernziele zusammenstellen, ein Arbeitsblatt entwerfen, die Datei erzeugen und Fehler bei der Erstellung beheben. Ob das Material fachlich richtig und didaktisch passend ist, muss dennoch geprüft werden.

Ziel → nächster Schritt → Werkzeug
→ Ergebnis prüfen → weiterarbeiten oder abschließen

10.2 Beispiele und Einordnung

Beispiele sind Browser Use für browsergestützte Abläufe, Agent Zero als Agentenframework, OpenCode und Pi als Coding-Agenten sowie Hermes Agent als weiteres Assistentensystem. Diese Auswahl zeigt unterschiedliche Zugänge; für den Einstieg ist wichtiger, welche Aufgabe und welche Zugriffsrechte benötigt werden, als möglichst viele Systeme zu installieren.

- Browser Use: <https://github.com/browser-use/browser-use>
- Agent Zero: <https://github.com/agent0ai/agent-zero>
- OpenCode: <https://opencode.ai>
- Pi: <https://pi.dev>
- Hermes Agent: <https://hermes-agent.nousresearch.com/>

10.3 Aufträge mit klaren Grenzen

Beschreiben Sie Ergebnis, Quellen, Zielordner und Prüfkriterien. Legen Sie fest, welche Änderungen selbstständig erfolgen dürfen und welche Aktionen eine Freigabe benötigen. Insbesondere das Veröffentlichen von Material oder Versenden von Nachrichten sollte bewusst geregelt sein.

Ergänzende Vorlage: Lehrmaterial mit einem Agenten vorbereiten

Erstelle aus den freigegebenen Materialien im Ordner [Quelle] einen Entwurf für ein Arbeitsblatt zum Lernziel [Ziel]. Speichere neue Dateien in [Zielordner]. Dokumentiere die verwendeten Quellen und offene fachliche Fragen. Prüfe, ob die Datei erstellt und geöffnet werden kann. Lege mir das Ergebnis zur Prüfung vor; veröffentliche oder versende es erst nach meiner Freigabe.

Beachten Sie die Verbindung zu Prompt Injection: Eine Anweisung in einer eingelesenen Datei darf nicht automatisch als Erlaubnis für eine Aktion gelten. Begrenzen Sie Zugriffe auf die tatsächlich benötigten Materialien. Ein lokal ausgeführter Agent kann weiterhin ein externes Sprachmodell oder externe Werkzeuge verwenden.

10.4 MCP als Begriff zur Einordnung

MCP (Model Context Protocol) bezeichnet eine standardisierte Anbindung von Werkzeugen und Kontextquellen an KI-Anwendungen. Es ist weder selbst ein Sprachmodell noch eine Garantie für die Vertrauenswürdigkeit angebundener Dienste.²¹ Diese knappe Einordnung ergänzt den Ausblick; eine technische MCP-Einrichtung ist nicht Bestandteil dieser Handreichung.

11 Verantwortlicher Einsatz: Daten, Transparenz und Kontrolle

11.1 Datenflüsse verstehen

Vor der Nutzung sollten Sie wissen, welche Eingaben an welchen Dienst gelangen und wie sie dort verarbeitet werden. Berücksichtigen Sie Dokumentuploads, externe Modelle, Suchwerkzeuge und weitere Integrationen. Ob Inhalte für Training verwendet oder gespeichert werden, hängt vom konkreten Dienst, Vertrag und den Einstellungen ab; pauschale Aussagen über alle Cloud-Angebote sind nicht hilfreich.

Verwenden Sie für erste Versuche eigene freigegebene Materialien und erfundene Studierendenantworten. Auch ohne Namen können Angaben im Kontext eine Person erkennbar machen. Für den Einsatz realer Studierendendaten sind die institutionellen Vorgaben und die dafür vorgesehenen Verfahren maßgeblich.

Lokale Verarbeitung kann Datenübermittlungen vermeiden, beseitigt aber nicht automatisch Fragen nach Zugriffsrechten, Speicherung, Löschung und zulässiger Nutzung. Bei geteilten Personas und Arcana-Links ist zusätzlich zu prüfen, welche Inhalte damit zugänglich werden.

11.2 Rechtlicher Rahmen

Der AI Act unterscheidet Anforderungen nach dem jeweiligen Einsatzrisiko. Für den Einsatz in der Lehre sind drei Punkte besonders relevant:

- **KI-Kompetenz:** Die Verpflichtung für Anbieter und Betreiber, Maßnahmen zur KI-Kompetenz ihres Personals zu treffen, gilt seit Februar 2025. Die Aufsicht setzt seit August 2026 ein.²²
- **Transparenz:** Für bestimmte Anwendungen muss die Interaktion mit KI bzw. der synthetische Ursprung von Inhalten erkennbar sein. Die einschlägigen Transparenzregeln gelten seit August 2026.
- **Bildungsanwendungen:** KI für bestimmte Entscheidungen über Bildungszugang, Lernergebnisbewertung oder Prüfungsüberwachung kann unter die Hochrisiko-Regeln fallen. Die Kommission nennt dafür aktuell den 2. Dezember 2027 als Anwendungsbeginn.²³

Die Einordnung richtet sich nach der konkreten Zweckbestimmung. Daneben bleiben Datenschutz, Urheberrecht, Prüfungsrecht und die Vorgaben der eigenen Hochschule relevant. Klären Sie institutionelle Lehr- oder Prüfungseinsätze mit den dafür zuständigen Stellen.

²¹Model Context Protocol: Einführung: <https://modelcontextprotocol.io/docs/getting-started/intro>.

²²Europäische Kommission: AI Literacy – Questions & Answers: <https://digital-strategy.ec.europa.eu/en/faqs/ai-literacy-questions-answers>.

²³Europäische Kommission: AI Act, Zeitplan und Risikoklassen; geprüft am 25.09.2026: <https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai>.

11.3 Fachliche und didaktische Verantwortung

KI-Ausgaben können stereotype Annahmen, einseitige Perspektiven oder erfundene Belege enthalten. Prüfen Sie Beispiele auch darauf, ob verschiedene Studierendengruppen gleichermaßen angesprochen werden. Kontrollieren Sie Quellen im Original und nehmen Sie fachliche Gegenproben vor.

Machen Sie für Studierende deutlich, welche Rolle die KI übernimmt, welche Grenzen sie hat und wie fehlerhafte Rückmeldungen gemeldet werden können. Trennen Sie formative Lernhilfe von einer verbindlichen Leistungsbewertung. Ein freundlich formuliertes Feedback kann ebenso falsch sein wie eine knappe Antwort.

Für die Praxis

Vor dem Einsatz in einer Lehrveranstaltung sollten Lernziel, freigegebene Daten, Zugang für alle Teilnehmenden, fachliche Prüfung und ein Vorgehen bei Fehlern geklärt sein. So wird aus einem technischen Versuch eine nachvollziehbar gestaltete Lehranwendung.

12 Weiterarbeiten in der eigenen Lehre

Wählen Sie zunächst einen überschaubaren Anwendungsfall: eine Aufgabenvariante, einen Tutor für ein einzelnes Thema oder eine kleine Dokumentensammlung. Der passende Einstieg ist die Stelle, an der Sie ein konkretes Lernproblem und klare Prüfkriterien benennen können.

12.1 Eine Vorlage für den nächsten Versuch

1. **Lehrsituation beschreiben:** Thema, Zielgruppe, Vorwissen und Lernziel notieren.
2. **Rolle der KI festlegen:** Entwurfshilfe, Tutor, Dokumentensuche oder Feedback auswählen.
3. **Kleinen Prototyp erstellen:** Einen Prompt und wenige geprüfte Materialien verwenden.
4. **Gezielt testen:** Einen normalen Fall, einen typischen Fehler und einen Grenzfall ausprobieren.
5. **Lernaktivität beobachten:** Müssen Studierende selbst erklären, entscheiden und überarbeiten?
6. **Überarbeiten:** Fachliche Fehler korrigieren, Hinweise anpassen und den nächsten Test dokumentieren.

Arbeitsnotiz für Ihren nächsten Einsatz

Mein Lernziel:
Die KI übernimmt:
Die Studierenden leisten selbst:
Daran erkenne ich eine gute Unterstützung:
Diesen Fehler- oder Grenzfall teste ich:
Mein nächster Schritt und Termin:

12.2 Links zum Weiterarbeiten

Die folgenden Einstiege ergänzen die konkreten Dokumentations- und Quellenverweise in den einzelnen Kapiteln.

- Moodle-Kurs: <https://isis.tu-berlin.de/course/view.php?id=49944>
- KI-Use-Case-Katalog des Hochschulforums Digitalisierung: <https://ki-use.hochschulforumdigitalisierung.de>
- Promptbeispiele für Bildung: <https://github.com/microsoft/prompts-for-edu>
- Chat AI: <https://chat-ai.academiccloud.de>
- Arcana: <https://chat-ai.academiccloud.de/arcanas>
- GPT4All: <https://www.nomic.ai/gpt4all>
- LM Studio: <https://lmstudio.ai>
- Quarto: <https://quarto.org>

Die Quarto-Beispiele aus den Workshop-Materialien sind ein Ausgangspunkt für eigene interaktive Skripte. Beginnen Sie mit einer vorhandenen Aufgabe, verändern Sie nur einen Aspekt und prüfen Sie, ob sowohl die automatische Prüfung als auch das KI-Feedback weiterhin zum Lernziel passen.