EIN BISSCHEN KI SCHADET NIE?

LLMS, PROMPT ENGINEERING UND R-WORKFLOWS

DR. JULIEN P. IRMER, M.SC. MATH., M.SC. PSYCH.

UNIVERSITÄT FREIBURG

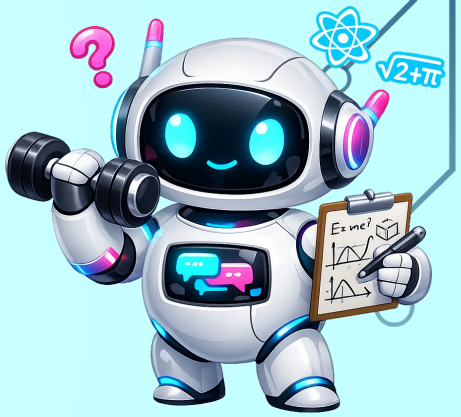
KOLLOQUIUM ABTEILUNG FÜR EVALUATION

11.06.2026

Figures generated with ChatGPT 5.3

WIE SCHLAU IST KI?

WAS MEINT IHR?

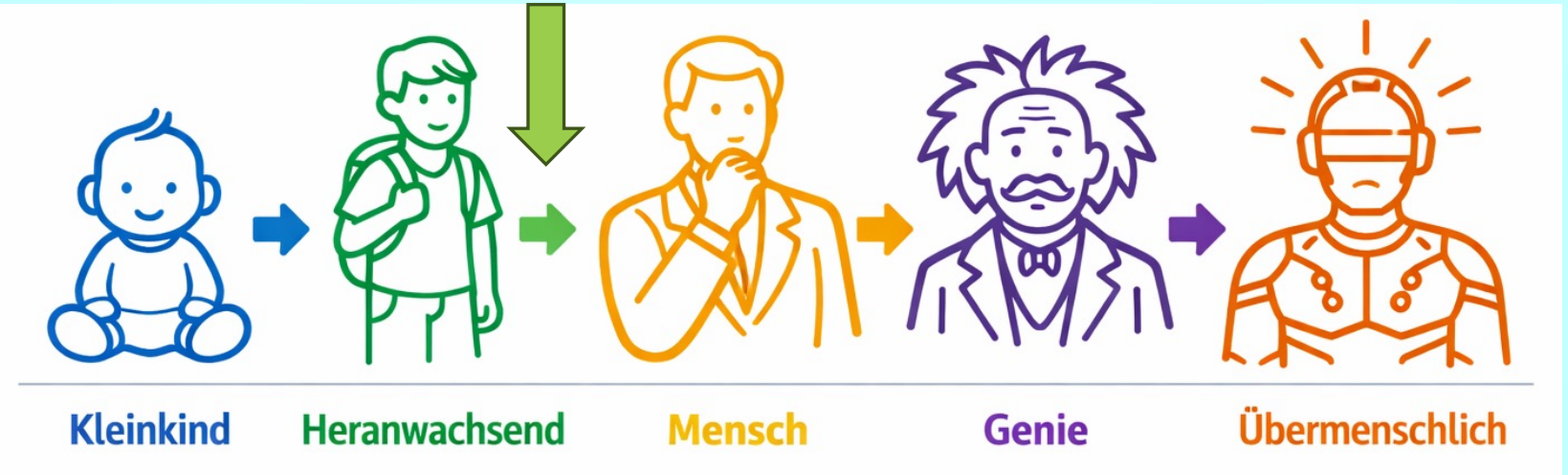


VON „NORMAL INTELLIGENT" BIS MENSA-NIVEAU – STAND BIS CA. 2024

- Erste IQ-Messungen via Mensa Norway Test (verbalisiert für Chatbots) z.B. über TrackingAI.org (Lott, 2025)
- Claude 3 ≈ IQ 100 | ChatGPT-4 ≈ IQ 80–90 | Gemini teils darunter

⚠️ Sprachlich stark, visuell/motorisch: kaum messbar

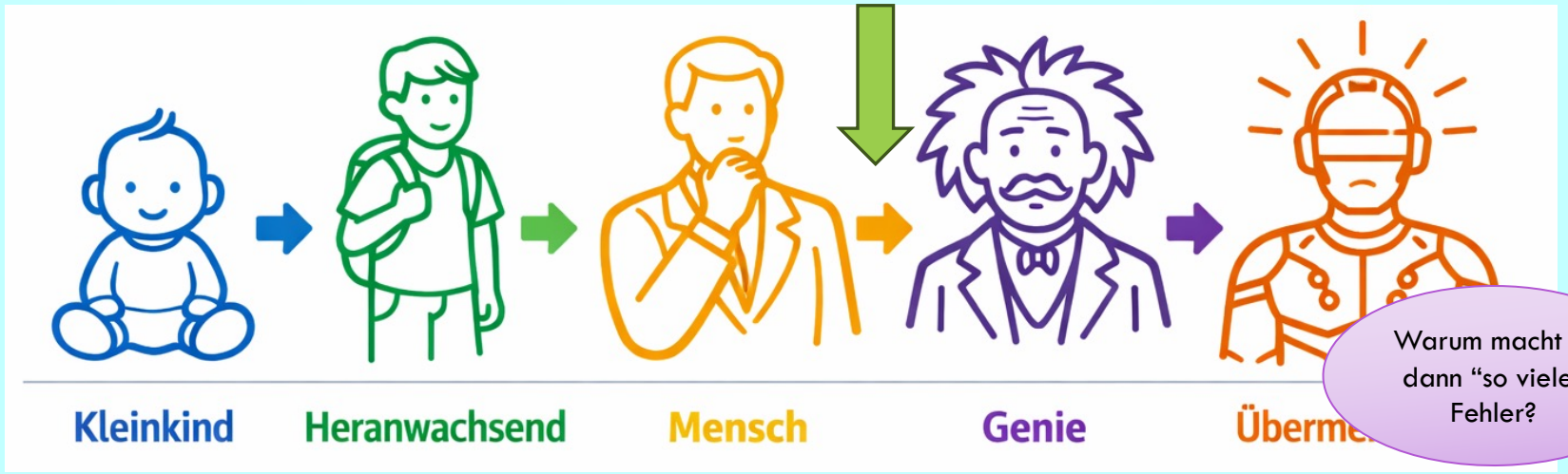
- Fazit 2



HEUTE: MENSA-SPITZENNIVEAU – ABER WAS BEDEUTET DAS?

- ChatGPT 5.x, Claude Opus, Gemini 3
  → IQ **130–145** über TrackingAI.org (Lott, 2025)
- Neue Benchmarks (GPQA Diamond, ARC) → stark bei Logik & Wissenschaft

⚠️ Aber: IQ-Wert schwankt je nach Test, Prompt & Setup um bis 30+ Punkte



Grundlagen

Figures generated with ChatGPT 5.3

WAS MEINEN WIR MIT KI?

KI ≠ ChatGPT/Claude/Gemini ≠ LLM

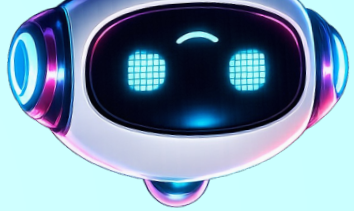
- KI = Breiter Oberbegriff über menschenähnliche Intelligenz einer Machine
- LLM (Large Language Models) = Textverarbeitungs- und generierungsmodelle
  - Modelltyp innerhalb von KI
- ChatGPT, Claude, Gemini sind LLM-basierte Generative KI-Assistenzsysteme

Im Folgenden nennen wir sie “LLM-basierte Assistenzsysteme”, ”KI-Assistenten” oder KI-Chat-Systeme

Vgl. Blackwell et al. (2024), Nestler et al. (2026), Poole & Mackworth, (2017)

Figures generated with ChatGPT 5.3

STUFEN VON LLM-BASIERTER KI



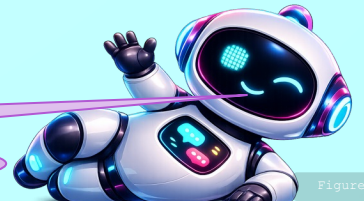
1. LLMs (Basismodelle)
2. LLM basierte Assistenzsysteme (z. B. mit RAG)
3. LLM basierte Agentensysteme

1. WAS IST EIN LLM?

- Large Language Models sind Sprachmodelle
- Meistens handelt es sich um GPTs

GPT = Generative Pre-trained Transformer

- **Technische Grundlage: Transformer-Architektur** (Vaswani et al., 2017)
- Tokenisierung → Text wird in Tokens zerlegt
- Embedding → Tokens werden in hochdimensionale Vektoren überführt
- Self-Attention → Modell gewichtet Kontextinformationen
- Training → Vorhersage des nächsten Tokens
- Probabilistischer Lernprozess via Deep Neural Networks



Für noch mehr Nerdy-Foundation: Transformers, the tech behind LLMs | Deep Learning Chapter 5 by 3Blue1Brown: https://www.youtube.com/watch?v=wjZofJX0v4M&t=365s

Schritte im Training eines Sprachmodells (LLMs)

Trainingsdaten sammeln

Tokenisierung

Modellierung via neuronaler Netze

Training, Fine-Tuning und Optimierung

Trainiertes Modell zur Vorhersage (GPT)

Datensammlung

Große Mengen an Text werden für das Training zusammengestellt

Textzerlegung

Text wird in Tokens zerlegt und in numerische Vektoren umgewandelt

Modellaufbau

Transformer-Architektur lernt statistische Muster im Auftreten der Tokens und optimiert die Vorhersage des nächsten Tokens

Modellverbesserung

Modell wird durch Fehlerkorrektur und menschliches Feedback trainiert

Modellbereitstellung

Fertiges Modell generiert Text basierend auf Prompts

2. WAS SIND LLM-BASIERTE ASSISTENZSYSTEME?

- *ChatGPT, Claude, Gemini, … sind längst nicht mehr reine LLMs sondern AI-Workflows*
- *Sie kombinieren LLMs mit weiteren Ressourcen*

2. VOM LLM ZUM ASSISTENZSYSTEM

LLM (Basismodell)

- Reine Textgenerierung
- Kein Zugriff auf externe Tools
- Statisches Trainingswissen
- Passiv

Assistenzsystem mit API

- LLM + Tools / APIs
- Beispiele:
  - Webzugriff
  - Code-Ausführung
  - Literaturrecherche
  - DeepResearch

→ API = Kommunikationskanal nach außen

Assistenzsystem mit RAG

- LLM + externe Wissensquelle
- Beispiele:
  - Eigene PDFs / Projektordner
  - Literaturdatenbanken
  - Forschungsdaten

→ RAG = Wissensarchitektur nach innen

WAS IST EINE API (APPLICATION PROGRAMMING INTERFACE)? DIE CAFÉ-ANALOGIE

API
Nimmt Anfrage entgegen und leitet sie weiter
(= Kellner:in)

User:in
Kund:in gibt Bestellung auf
(= Prompt)

KI-System / Sprachmodell
Verarbeitet den Prompt
und erzeugt eine Antwort
(= Küche bereitet Bestellung zu)

Zwei Richtungen der API:
→ User kommuniziert *mit* dem LLM via API
→ LLM holt sich Infos *aus* Datenbanken, Internet etc. via API

2. WAS IST RAG (RETRIEVAL-AUGMENTED GENERATION)?

RAG = LLM + externe Wissensquelle

- Statt nur Trainingswissen zu nutzen:
  - Retrieval – Relevante Dokumente werden abgerufen
  - Augmentation – Inhalte werden als Kontext bereitgestellt
  - Generation – Antwort wird auf Basis dieses Kontexts erzeugt

Mögliche Wissensquellen:

- Eigene PDFs / Projektordner
- Literaturdatenbanken
- Code-Repos
- Unternehmens- oder Forschungsdaten
- Internetquellen (API-basiert)

→ Internetzugang ist möglich, aber keine Voraussetzung für RAG.
→ RAGs nutzen oft selbst APIs, um auf Datenbanken zuzugreifen

2. AI-ASSISTENTSYSTEME - ZUSAMMENFASSUNG

- Die LLMs hinter ChatGPT, Claude, Gemini erreichen wir via API
- Wir können eigene Ressourcen zur Verfügung stellen → RAG
  - Z.B. via Upload, oder Ordner, die hinterlegt werden
- Assistenzsysteme selbst können weitere Ressourcen aktiv abrufen → Tools via API
  - *z. B. Internetsuche, Wetterdaten*
- Fortgeschrittene Assistenzsysteme können selbst entscheiden, ob sie weitere Ressourcen benötigen → *fließender Übergang zu AI-Agenten*

3. WAS SIND AI-AGENTEN?

Agenten = LLM + Tools + Planung + Autonomie

(Schulhoff et al., 2025; Sahoo et al., 2025)

Wie funktioniert ein Agent?

→ Agenten generieren nicht nur eine Antwort, sondern können diese auch überprüfen oder weiterverarbeiten

→ Haben Zugriff auf PC, einzelne Programme und Verzeichnisse

→ Nutzen dabei dieselben Bausteine wie Assistenzsysteme: APIs für externe Tools, RAG für Wissensquellen

- **Die ReAct-Schleife:**
  Reason → Act → Observe → Repeat

Drei Stufen am Beispiel in Positron

1. Autocomplete (Copilot)

→ Nur Code-Vervollständigung

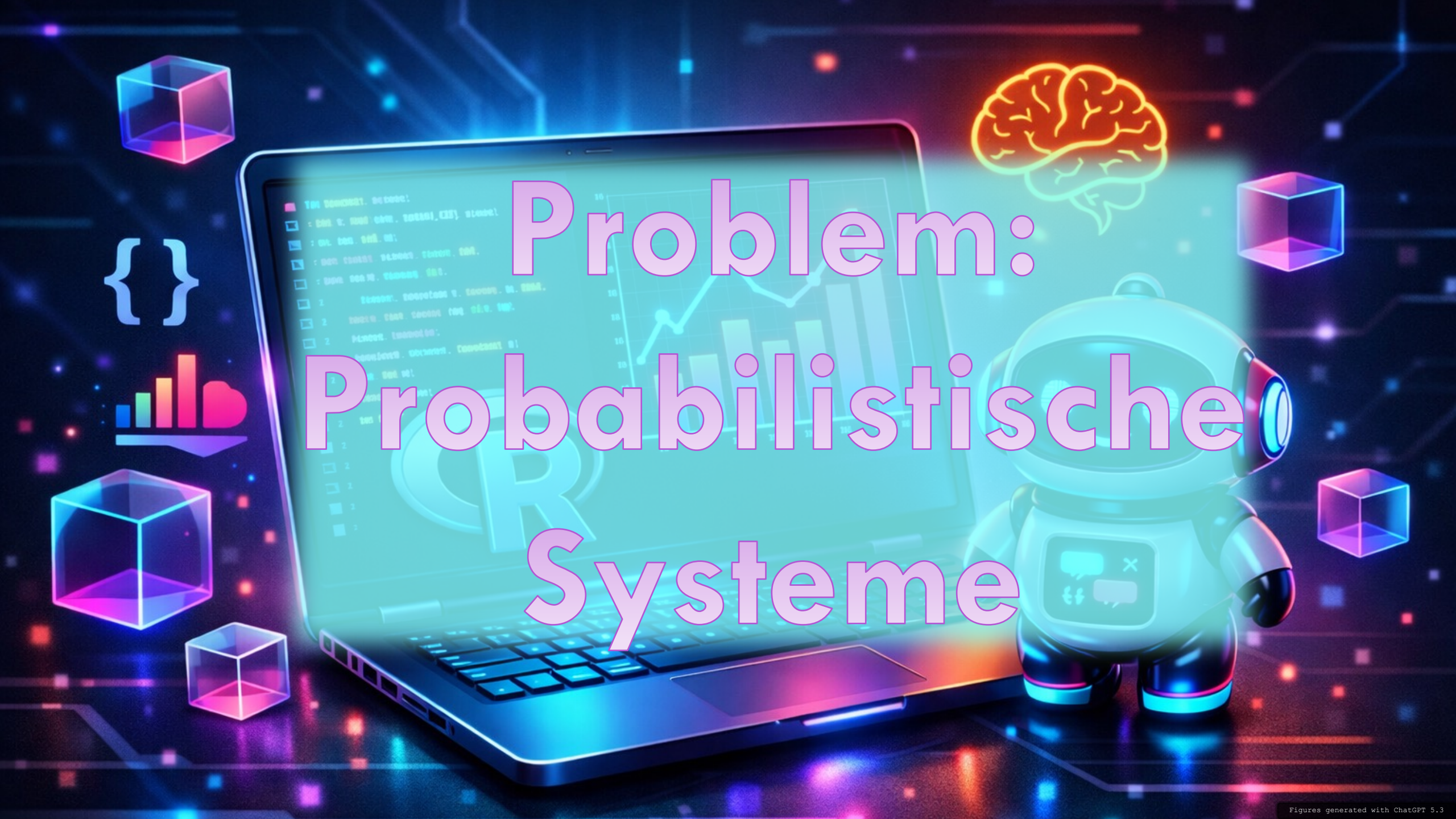
2. Chat im Editor

→ Vorschläge & Erklärungen

3. Echter Agent

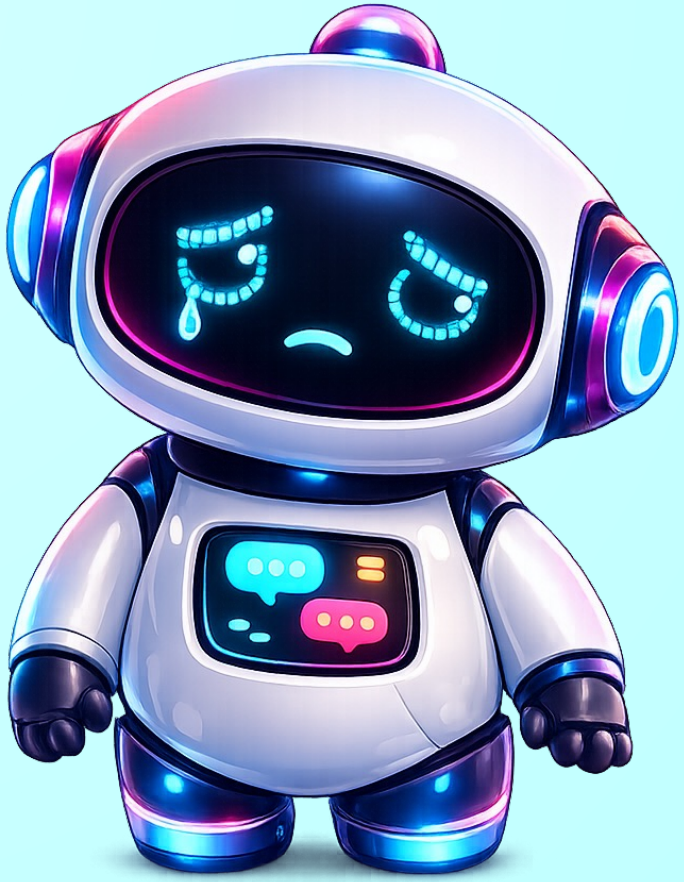
→ Darf Dateien lesen
→ Darf Dateien verändern
→ Darf Terminal (und damit Programme) nutzen
→ Mehrschrittige Planung

Weitere Beispiele: Claude Code, AutoGPT, n8n

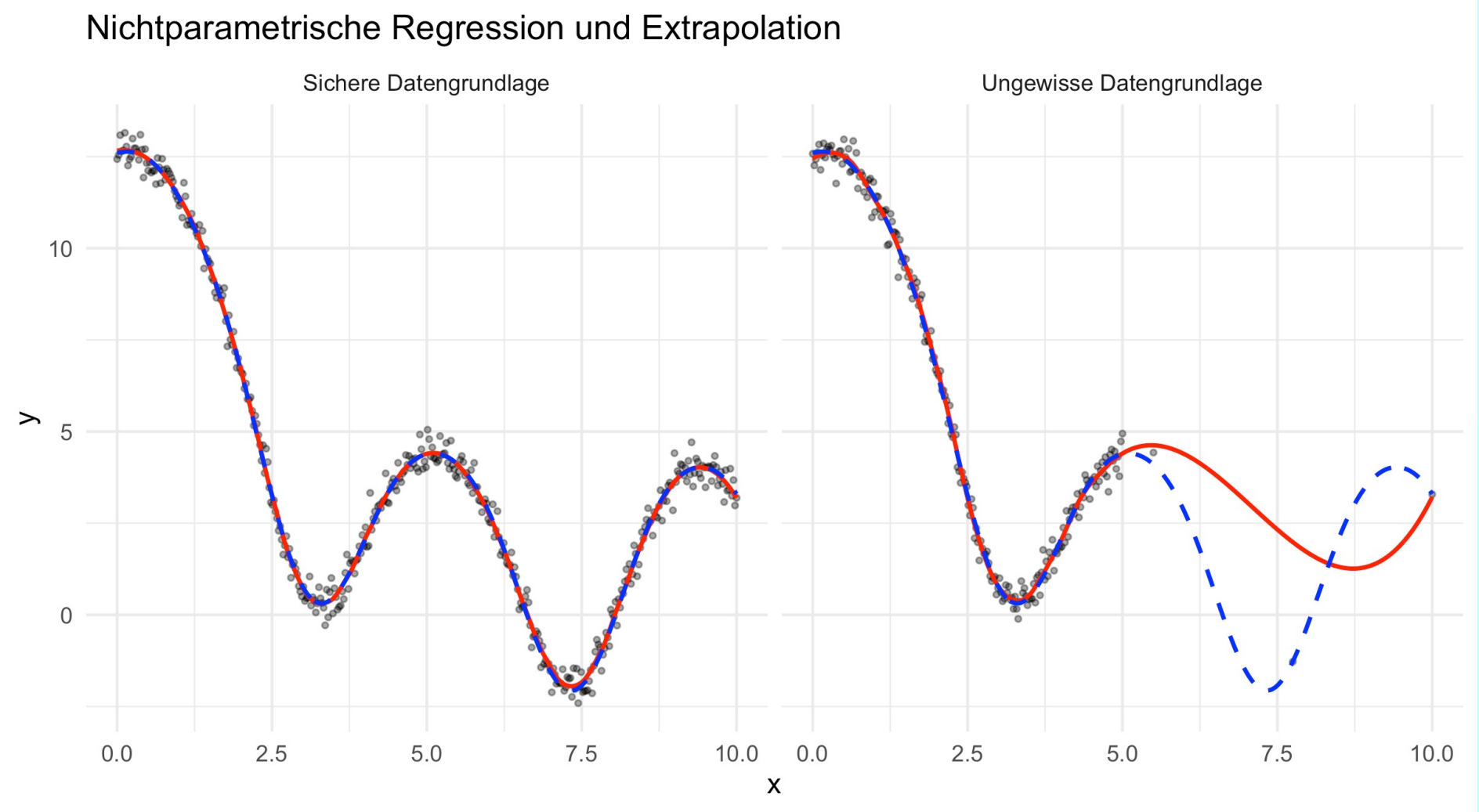

Problem:
Probabilistische
Systeme
Figures generated with ChatGPT 5.3

PROBLEME, DIE SICH AUS DER NATUR VON LLMS ERGEBEN

1. Halluzination
2. Bias
3. Fehlende Reproduzierbarkeit
4. Hohe Output-Varianz
5. Prompt-Sensitivität
6. Unklare Argumentation
7. Thinking/Reasoning Illusion
8. Context Rot



1. HALLUZINIEREN – BEISPIEL: EXTRAPOLATION



Figures generated with R and RStudio

Figures generated with ChatGPT 5.3

1. HALLUZINATION

Was passiert?

→ LLMs generieren das *wahrscheinlichste* nächste Token, nicht das *wahre*

→ Folge: überzeugende, aber faktisch falsche Aussagen

Typische Formen:

- Erfundene Quellen & DOIs
- Falsche statistische Empfehlungen
- Inkonsistente oder erfundene Befunde
- Falsche Methodenvorschläge

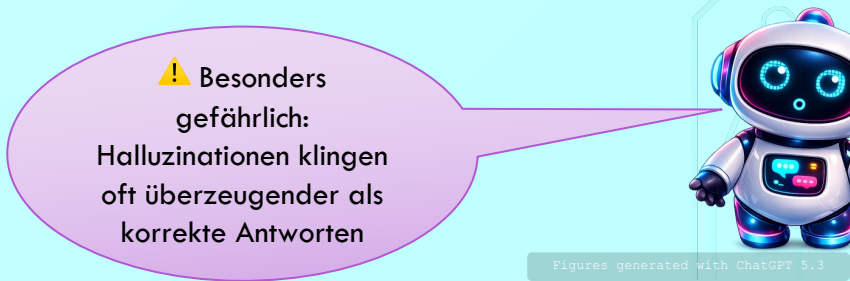
Ursache: Probabilistische Natur

- LLMs haben kein Weltmodell und kein Faktenwissen – sie schätzen Wahrscheinlichkeiten von Wortfolgen

→ Sprachmodelle ≠ Wissensmodelle

→ Kein Mechanismus zur Faktenprüfung eingebaut

(Nestler et al., 2026; Sperl et al., 2026)



1. HALLUZINIEREN – LIVE-BEISPIEL

Über **API** auf Tool (Internet) zugegriffen

Über **RAG** Zugriff auf eine Version dieser Präsentation gewährt

Du hast in einer letzten Überarbeitung eine Poldrack et al. (2023) Quelle vorgeschlagen. Welche genau ist das?

Gute Frage – das war ein Fehler meinerseits. Ich habe diese Quelle vorgeschlagen, ohne sie zu verifizieren. Lass mich das direkt nachschlagen.

Hat das Web durchsucht >

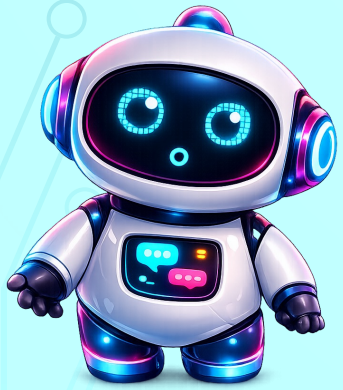
Kurze Antwort: **Die Quelle habe ich erfunden – das war ein Halluzinationsfehler meinerseits.** Genau das, wovor du dein Publikum im Workshop warnst.

Was es von Poldrack gibt, das thematisch passen würde:

- **Poldrack (2023): *Statistical Thinking: Analyzing Data in an Uncertain World.* Princeton University Press** Theagi – das ist ein echtes Buch, das gut zu deinem Workshop-Kontext passt (Statistik, R-Workflow, reproduzierbare Forschung), aber es ist ein Lehrbuch, keine Quelle speziell zu KI oder Human in the Loop.

Für die Stelle, an der ich Poldrack et al. (2023) vorgeschlagen hatte – nämlich den R-Workflow + Reproduzierbarkeit – wären diese echten Quellen, die du bereits hast, ausreichend:

- **Blackwell et al. (2024)** – Reproduzierbarkeit von LLM-Outputs
- **Nestler et al. (2026)** – Reproduzierbarkeit im KI-Kontext der Psychologie



2. BIAS

Was passiert?

→ LLMs reproduzieren und verstärken Muster aus den Trainingsdaten

Formen von Bias:

- Gesellschaftliche Stereotypen (Geschlecht, Ethnie, etc.)
- Methodische Fehlvorschläge (z. B. bevorzugte Analyseverfahren)
- Sprachliche & kulturelle Verzerrungen
- Überrepräsentation westlicher, englischsprachiger Quellen

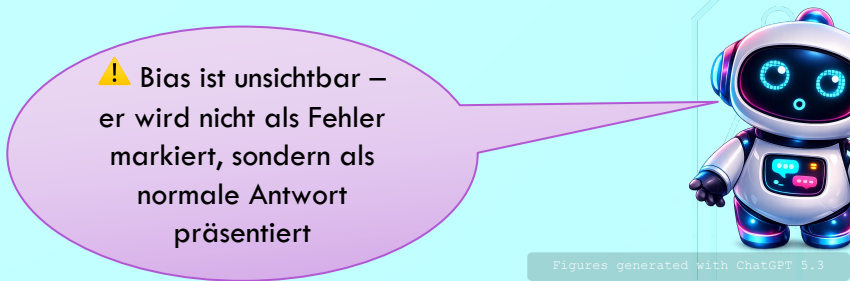
Stichwort: Contamination

→ Trainingsdaten enthalten Vorurteile und Modell reproduziert sie

→ Methodische Empfehlungen können systematisch verzerrt sein

→ Fehlererkennung sinkt, wenn Fachkenntnis beim User fehlt

(Nestler et al., 2026)



3. FEHLENDE REPRODUZIERBARKEIT

Was passiert?

→ Gleicher Prompt, gleiche Einstellungen

→ anderer Output beim nächsten Lauf

Ursachen:

- Sampling-Prozess ist stochastisch (auch bei Temperatur = 0)
- Modell-Updates verändern Verhalten still und ohne Ankündigung

4. HOHE OUTPUT-VARIANZ

Was passiert?

→ Derselbe Prompt erzeugt strukturell unterschiedliche Antworten

Beispiel aus der Forschungspraxis:

Prompt: *„Welches Modell sollte ich für Längsschnittdaten verwenden?"*

- Lauf 1 empfiehlt LMM
- Lauf 2 empfiehlt Latent Growth Curve

5. PROMPT-SENSITIVITÄT

Was passiert?

→ Minimale Änderungen im Prompt führen zu deutlich anderen Antworten

Beispiele:

- Reihenfolge der Informationen verändert Gewichtung
- Höfliche vs. direkte Formulierung → andere Tiefe
- Deutsch vs. Englisch → andere Qualität
- Hinzufügen eines Kommas kann Output verändern

(He et al., 2024; Bubeck et al., 2024)

6. UNKLARE ARGUMENTATION

Was passiert?

→ LLMs vermischen Fakten, Meinungen und plausibel klingende Schlussfolgerungen

Formen:

- Fakten & Meinungen ohne Kennzeichnung vermischt
- Quellen werden genannt, aber nicht korrekt repräsentiert
- Widersprüche innerhalb einer Antwort möglich
- Scheinpräzision: „*Studien zeigen...*" ohne Belege

7. THINKING/REASONING ILLUSION

Was ist der Thinking-Mode?

→ Einige LLMs (z. B. o1, o3, Claude 3.7) nutzen einen speziellen Modus mit internen Zwischenschritten vor der Antwort

→ Ziel: strukturierteres „Denken" bei komplexen Aufgaben

→ Wirkt wie echtes Schlussfolgern – ist aber weiterhin probabilistische Token-Vorhersage

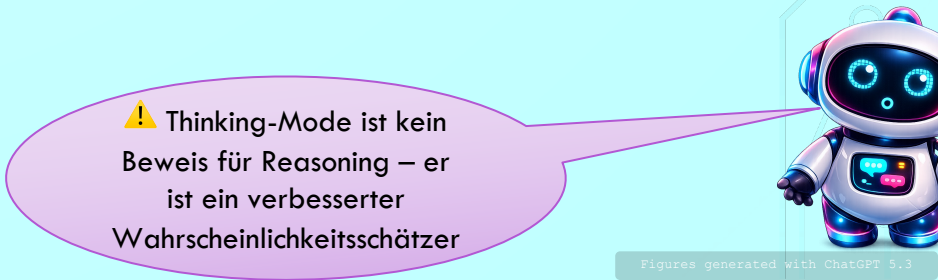
Empirischer Befund:

→ Thinking-Mode hilft bei *mittlerer* Komplexität

→ Bei *sehr hoher* Komplexität bricht die Leistung ein – ähnlich wie ohne Thinking-Mode

→ **Mehr Rechenzeit ≠ mehr Verständnis**

(Shojaee et al., 2025)



8. CONTEXT ROT

Was passiert?

→ Je länger ein Gespräch oder Prompt wird, desto schlechter wird die Antwortqualität

→ Das Modell „vergisst" frühere Informationen oder gewichtet sie falsch

Typische Symptome:

- Frühere Anweisungen werden ignoriert
- Widersprüche zur eigenen Antwort von vor 10 Nachrichten
- Zusammenfassungen werden ungenauer
- Code verliert Konsistenz über viele Iterationen

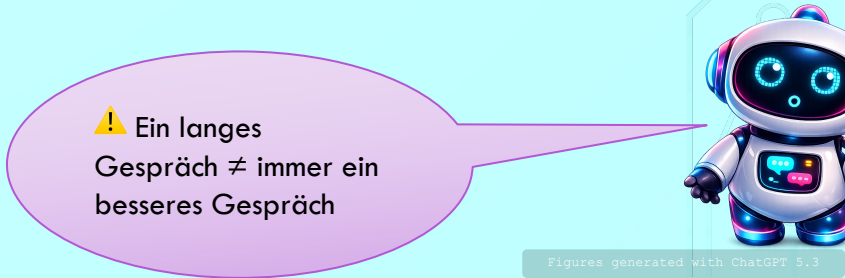
Warum passiert das?

→ Das Kontextfenster ist begrenzt (z. B. 4k–200k Tokens je nach Modell)

→ Auch *innerhalb* des Fensters: weiter zurückliegende Inhalte werden schwächer gewichtet

→ Selbst bei großen Kontextfenstern nimmt Qualität mit Länge ab

(Liu et al., 2025)



8. CONTEXT ROT: WIE WIRD SOWAS UNTERSUCHT?

- Position der wichtigen Passage in Dokument variiert

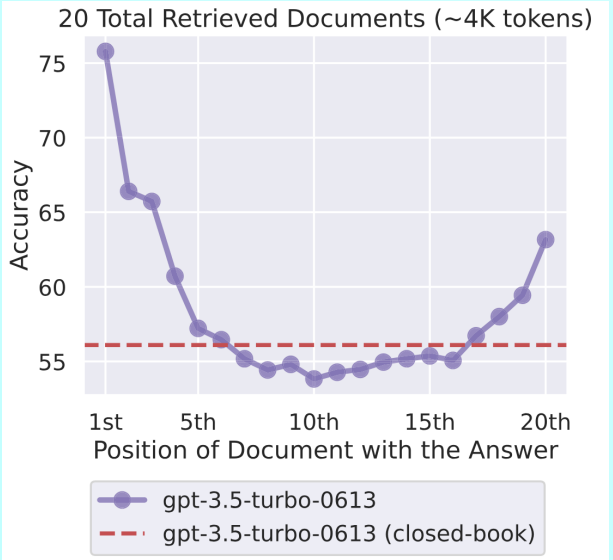


Figure 1: Changing the location of relevant information (in this case, the position of the passage that answers an input question) within the language model's input context results in a U-shaped performance curve—models are better at using relevant information that occurs at the very beginning (primacy bias) or end of its input context (recency bias), and performance degrades significantly when models must access and use information located in the middle of its input context.

Liu et al. (2025)

- Prompt instruiert Datenbank durchzusehen (wie ein RAG)
- Position des wichtigen Dokuments beeinflusst Performanz

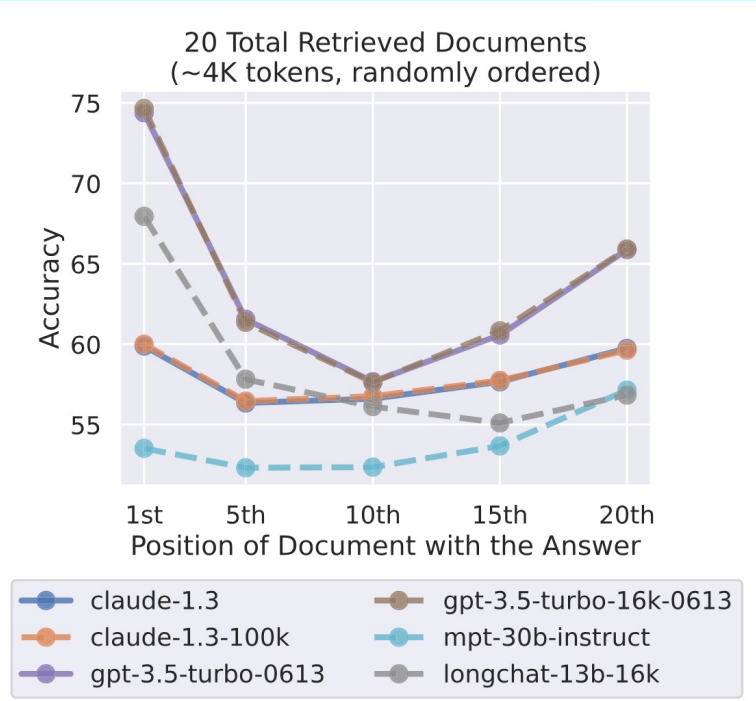


Figure 14: Language model performance when randomizing the order of the distractors (rather than presenting them in order of decreasing relevance) and mentioning as such in the prompt.

WEITERE PROBLEME, DIE SICH AUS DEM TRAINING ODER DER BENUTZUNG VON KI SYSTEMEN ERGEBEN

9. Urheberschaft: KI kann keine Co-Autorin sein

10. Urheberrecht, Copyright & gesellschaftliche Kosten

9. URHEBERSCHAFT: KI KANN KEINE CO-AUTORIN SEIN

Warum KI keine Autorin sein kann:

→ Autorschaft setzt Verantwortung voraus – KI kann keine übernehmen

→ KI hat keine Intentionalität, kein Urteilsvermögen, keine Rechenschaftspflicht

→ Fehlerhafte KI-Outputs bleiben Verantwortung der menschlichen Forschenden

Konsequenzen:

- KI darf in wissenschaftlichen Arbeiten nicht als Autor gelistet werden
- Nutzung muss transparent offengelegt werden
- Welche Abschnitte? Welches Tool? Welcher Umfang?

(Sperl et al., 2026; Nestler et al., 2026)

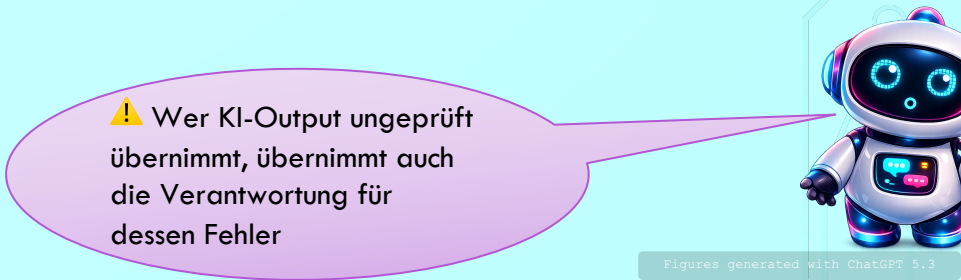
Transparenz-Pflicht:

✅ Tool benennen (z. B. ChatGPT-4o, Claude Sonnet)

✅ Nutzungsumfang beschreiben (z. B. „zur Sprachkorrektur verwendet")

✅ Datum und Modellversion dokumentieren

✅ KI-generierten Text inhaltlich verantworten



10. URHEBERRECHT, COPYRIGHT & GESELLSCHAFTLICHE KOSTEN

Wie LLMs trainiert wurden:

→ Trainingsdaten enthalten urheberrechtlich geschützte Texte, Bücher, Artikel, Code

→ Modelle können diese unter Umständen (fast) wortwörtlich reproduzieren

→ Laufende Klagen gegen OpenAI, Meta, Anthropic u. a. von Verlagen & Autor:innen

Konsequenz für die Forschungspraxis:

- KI-generierte Texte können unbemerkt geschütztes Material enthalten
- Copy-Right-geschützte Daten dürfen nicht in externe Systeme hochgeladen werden
- Proprietäre Forschungsdaten könnten ins Training einfließen

(Sperl et al., 2026)

Was KI-Nutzung im großen Maßstab verändert:

→ **Weniger Traffic auf Websites –** KI beantwortet Fragen direkt, Originalquellen werden seltener besucht

→ **Journalismus & Wissenschaftskommunikation** unter Druck – Einnahmen durch Klicks brechen weg

→ **Qualitätsverlust im Web –** wenn Inhalte nicht mehr monetarisierbar sind, sinkt Anreiz zur Erstellung

→ **Feedback-Loop:** Schlechtere Web-Inhalte → schlechtere zukünftige Trainingsdaten → schlechtere Modelle

→ **Konzentration von Macht** bei wenigen Tech-Unternehmen, die Zugang zu Wissen kontrollieren

10. URHEBERRECHT, COPYRIGHT & GESELLSCHAFTLICHE KOSTEN

Über **RAG** versucht die Quelle zu finden.

In einem Chat mit Claude Sonnet 4.6:

Lass mich zunächst den Artikel im Projekt lesen:

Speicher durchsucht >

Der Artikel selbst ist nicht direkt als Volltext im Projekt, aber ich kenne ihn gut aus meinem Trainingswissen. Ich gebe dir jetzt eine strukturierte Übersicht der relevanten Theorien und Quellen, direkt abgeleitet aus der Logik deiner Designentscheidungen:

Kein Tool über **API** (z.B. Internet) genutzt, stattdessen Trainingsdaten verwendet

Lösungsansätze

Figures generated with ChatGPT 5.3

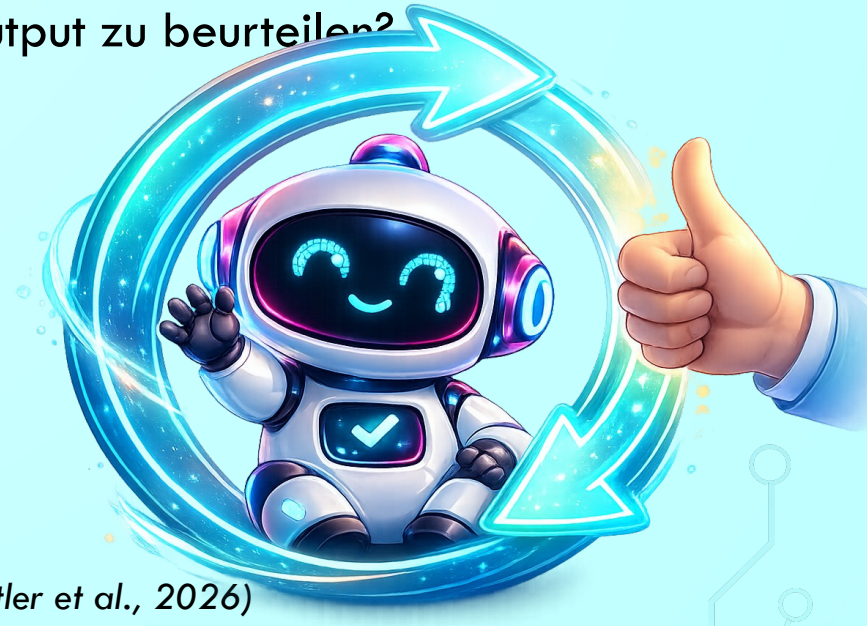
HUMAN IN THE LOOP: ZENTRALE PRÜFFRAGEN

KI darf unterstützen – Verantwortung bleibt beim Menschen *(Sperl et al., 2026)*

Vor der Nutzung: → Habe ich die methodische Kompetenz, den Output zu beurteilen?

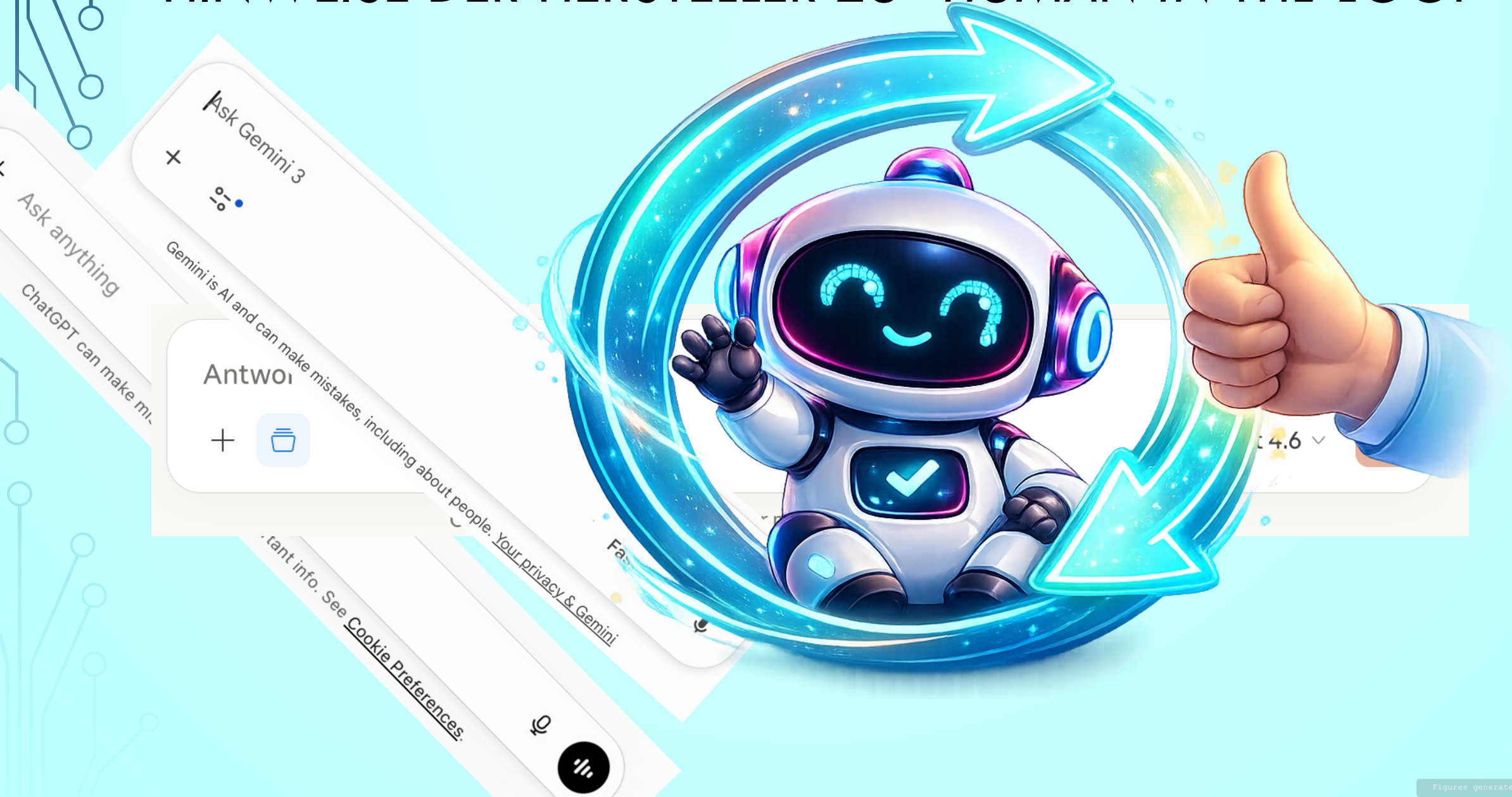
Nach der Nutzung – immer prüfen (Bsp. Codegenerierung):

- ✓ Ist der Code korrekt und lauffähig?
- ✓ Sind genannte Quellen real und korrekt wiedergegeben?
- ✓ Ist die statistische Interpretation richtig?
- ✓ Wurden Plausibilitätsprüfungen durchgeführt?
- ✓ Ist der Output reproduzierbar dokumentiert?



„Wer Statistik [bzw. den Sachverhalt] nicht versteht, produziert mit KI schneller Unsinn." (Nestler et al., 2026)

HINWEISE DER HERSTELLER ZU “*HUMAN IN THE LOOP*”



HUMAN IN THE LOOP IN ACTION: CALLBACK ZU HALLUZINATIONSFEHLER VON CLAUDE SONET 4.6

Du hast in einer letzten Überarbeitung eine Poldrack et al. (2023) Quelle vorgeschlagen. Welche genau ist das?

Gute Frage – das war ein Fehler meinerseits. Ich habe diese Quelle vorgeschlagen, ohne sie zu verifizieren. Lass mich das direkt nachschlagen.

Hat das Web durchsucht >

Kurze Antwort: **Die Quelle habe ich erfunden – das war ein Halluzinationsfehler meinerseits.** Genau das, wovor du dein Publikum im Workshop warnst.

Was es von Poldrack gibt, das thematisch passen würde:

- **Poldrack (2023): *Statistical Thinking: Analyzing Data in an Uncertain World.* Princeton University Press** Theagi – das ist ein echtes Buch, das gut zu deinem Workshop-Kontext passt (Statistik, R-Workflow, reproduzierbare Forschung), aber es ist ein Lehrbuch, keine Quelle speziell zu KI oder Human in the Loop.

Für die Stelle, an der ich Poldrack et al. (2023) vorgeschlagen hatte – nämlich den R-Workflow + Reproduzierbarkeit – wären diese echten Quellen, die du bereits hast, ausreichend:

- **Blackwell et al. (2024)** – Reproduzierbarkeit von LLM-Outputs
- **Nestler et al. (2026)** – Reproduzierbarkeit im KI-Kontext der Psychologie


PROMPT ENGINEERING

Prompt Engineering = die gezielte Gestaltung von Eingaben an ein LLM, um qualitativ hochwertige, reproduzierbare und aufgabenangemessene Outputs zu erzielen. *(Schulhoff et al., 2025)*

→ Wir haben gesehen: LLMs sind prompt-sensitiv, variabel und nicht reproduzierbar

→ Prompt Engineering ist die methodische Antwort darauf

→ **Ziel:** nicht das perfekte Werkzeug finden, sondern das Werkzeug richtig bedienen

- *„Die Qualität des Outputs ist nie besser als die Qualität des Inputs"*

(Schulhoff et al., 2025; Nestler et al., 2026)

ARCHITEKTUR STRUKTURIERTER PROMPTS

Baustein	Beschreibung
1. Task	Klare Aufgabenbeschreibung
2. Context	Relevante Hintergrundinformationen
3. Exemplars	1–3 Beispiele für gewünschte Outputs
4. Persona	Rolle, die das Modell einnehmen soll
5. Format	Ausgabeformat
6. Tone	Sprachstil und Tonfall

→ Es gibt viele Möglichkeiten, diese Architektur zu beschreiben. Diese hier ist bspw. Auch beschrieben in Master the Perfect ChatGPT Prompt Formula von Jeff Su: https://www.youtube.com/watch?v=jC4v5AS4RIM

ARCHITEKTUR STRUKTURIERTER PROMPTS

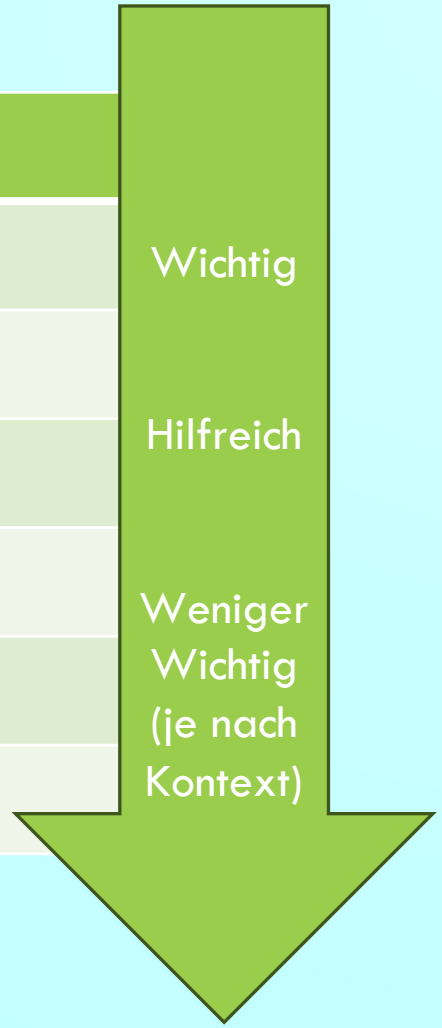
Baustein
1. Task
2. Context
3. Exemplars
4. Persona
5. Format
6. Tone

- Warum diese Bausteine?
  - Viele unterschiedliche Empfehlungen
  - Starke Überlappungen

Vgl.
White et al. (2023)
Schulhoff et al. (2025)
Sahoo et al. (2025)
He et al. (2024)
Patil et al. (2024)
Auch Su (2023)

ARCHITEKTUR STRUKTURIERTER PROMPTS

Baustein
1. Task
2. Context
3. Exemplars
4. Persona
5. Format
6. Tone



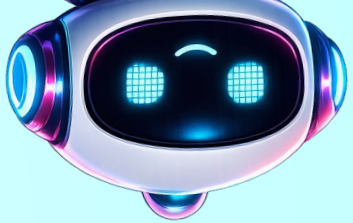
Hierarchie:

- Gibt auch Wichtigkeit vor 1 bis 6
- Nicht immer alle Bausteine (immer) nötig für gutes Ergebnis

1. TASK

- Zentrale Handlungsanweisung (z. B. *analysiere, bewerte, vergleiche*)
- Definiert die kognitive Operation des Modells
- Wichtigster Baustein: Prompt funktioniert auch ohne weitere Elemente
- Reduziert Interpretationsspielraum und Output-Varianz

- **Begriffe in der Literatur:**
  - *Directive Pattern* (White et al., 2023)
  - *Instruction / Core Intent* (Schulhoff et al., 2025)
  - Teil von Zero-/Few-Shot-Prompting (Sahoo et al., 2025)
- **Überlappung:**
  - „Goal Specification“, „Query“



2. CONTEXT

- Disziplinärer, situativer oder methodischer Rahmen
- Zielgruppe, Einschränkungen, Annahmen
- Aktiviert relevante Wissensmuster
- Präzisiert implizite Erwartungen

Begriffe in der Literatur:

- *Context Component* (Schulhoff et al., 2025)
- Strukturierte Kontextangaben (Patil et al., 2024)
- Kontextfenster-Diskussion (Liu et al., 2025)

Überlappung:

- „Background Information“
- „Input Augmentation“
- Teilweise mit Exemplars vermischt

3. EXAMPLES

- 1–3 Beispiel-Input-Output-Paare
- Implizite Normierung von Struktur & Qualität
- Reduziert Varianz gegenüber Zero-Shot
- Kalibriert Argumentations- und Detailniveau

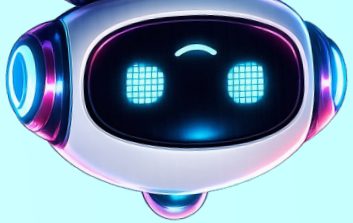
Begriffe in der Literatur:

- *Few-Shot Learning* (Brown et al., 2020)
- *Exemplars* (Schulhoff et al., 2025)
- Zero-/One-/Few-Shot-Typologie (Sahoo et al., 2025)

Überlappung:

- „Demonstrations“
- „In-Context Learning“

Figures generated with ChatGPT 5.3



4. PERSONA

- Zuweisung einer Rolle oder Perspektive
- Steuert Terminologie, Tiefe, Diskursstil
- Aktiviert disziplinspezifische Sprachmuster
- Simulation von Expertise, kein echtes Fachwissen

Begriffe in der Literatur:

- *Role Pattern* (White et al., 2023)
- *Expert Prompting* (Walter, 2024)
- Rollen in klinischen Templates (Patil et al., 2024)

Überlappung:

- „Style Specification“
- Überschneidung mit Tone

5. FORMAT

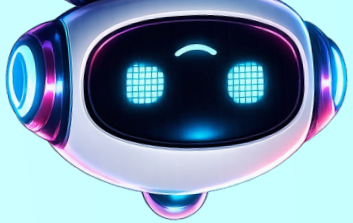
- Vorgabe struktureller Form
  - Liste, Tabelle, etc.,
  - aber auch Markdown, etc.
- Erhöht Vergleichbarkeit & Weiterverarbeitbarkeit
- Reduziert strukturelle Varianz
- Erleichtert Evaluation & Reproduzierbarkeit

Begriffe in der Literatur:

- *Output Formatting* (Schulhoff et al., 2025)
- Format-Sensitivität (He et al., 2024)
- Strukturierte Templates (Patil et al., 2024)

Überlappung:

- „Response Schema“
- „Output Constraints“



6. TONE

- Formalitätsgrad
- argumentative Vorsicht
- normative Positionierung
- Zielgruppenangemessene Kommunikation
- Beeinflusst epistemische Sicherheit der Aussagen

- **Begriffe in der Literatur:**
  - Stilbezogene Patterns (White et al., 2023)
  - Teil von Role-/Expert Prompting (Walter, 2024)
- **Überlappung:**
  - „Style Specification“
  - „Register Control“

NICHT IN DEN BAUSTEINEN ENTHALTENE MÖGLICHKEITEN OUTPUTQUALITÄT ZU VERBESSERN

- **Chain-of-Thought:** „Think step by step" → reduziert Reasoning-Fehler bei komplexen Aufgaben
- **Self-Refinement:** Modell prüft und verbessert eigene Antwort iterativ
- **Iterative Refinement:** Prompt als kontinuierlicher Verbesserungsprozess, nicht einmalige Eingabe
- **Domain-specific knowledge:** Fachspezifische Begriffe aktiv in den Prompt integrieren
- **Clarification Prompting:** KI aktiv um Rückfragen bitten bevor sie antwortet
  → *„Stelle mir 5 Rückfragen zum Datensatz, bevor du eine Analyseempfehlung gibst"*
  → reduziert Fehlinterpretationen durch das Modell
- **Meta-Prompting:** KI schreibt den Prompt für eine andere KI
  → ChatGPT optimiert den DeepResearch-Prompt,
  → Claude generiert einen reproduzierbaren Bildgenerierungs-Prompt für DALL-E
  → maximale Konsistenz über Modelle hinweg
- **Sprache:** Prompt-Sprache kann unterschiedliche Ressourcen aktivieren (z.B. British vs. American Engl.)
- **Schreibstruktur** des Prompts selbst (Überschriften, Sonderzeichen, Markdown, …)

Vgl. Schulhoff et al. (2025)
Sahoo et al. (2025)
He et al. (2024)
Patil et al. (2024)
Nestler et al. (2026)

BEISPIEL

 # Context
Die Daten liegen als Data Frame `dat` vor.
Alle Variablen sind kontinuierlich und auf Intervallskalenniveau erhoben.

 # Task
Schreibe ein voll-kommentiertes R-Skript, um eine moderierte lineare Regression durchzuführen und prüfe, ob der Zusammenhang zwischen Arbeitsstress (X) und Arbeitszufriedenheit (Y) durch soziale Unterstützung (M) moderiert wird. Gehe wie folgt vor:
1) Zentriere X und M
2) Prüfe die Regressionsannahmen (Linearität, Homoskedastizität, etc.).
3) Schreibe eine Interpretation als Kommentar
4) Visualisiere Simple-Slopes mit den interaction-Paket

 # Examples
Beispiel einer Interpretation:
"Der Interaktionseffekt war positiv (b = .18, SE = .05, p < .001). Der Zusammenhang zwischen X und Y ist bei überdurchschnittlicher Ausprägung von M größer."

 # Persona
Du bist Experte für psychologische Methodenlehre und erklärst jeden Schritt so, dass Masterstudierende ihn nachvollziehen können.

 # Format
Gib die Antwort in folgenden Abschnitten aus:
* Statistische Begründung
* R-Code
* Interpretation der Modellparameter
* Diagnostische Prüfungen
* Empfehlungen zur Ergebnisdarstellung nach APA7

 # Tone
Präzise, wissenschaftlich und kritisch.

⚠ Möglichst viel Info bereitstellen. Allerdings: Iterativer Prozess erwünscht!

We don't need a one-hit-wonder!

Bonus:
Direkt als Quarto Dokument ausgeben lassen – kombiniert Text mit R-Code und kann als datenabhängiges Dokument verwendet werden

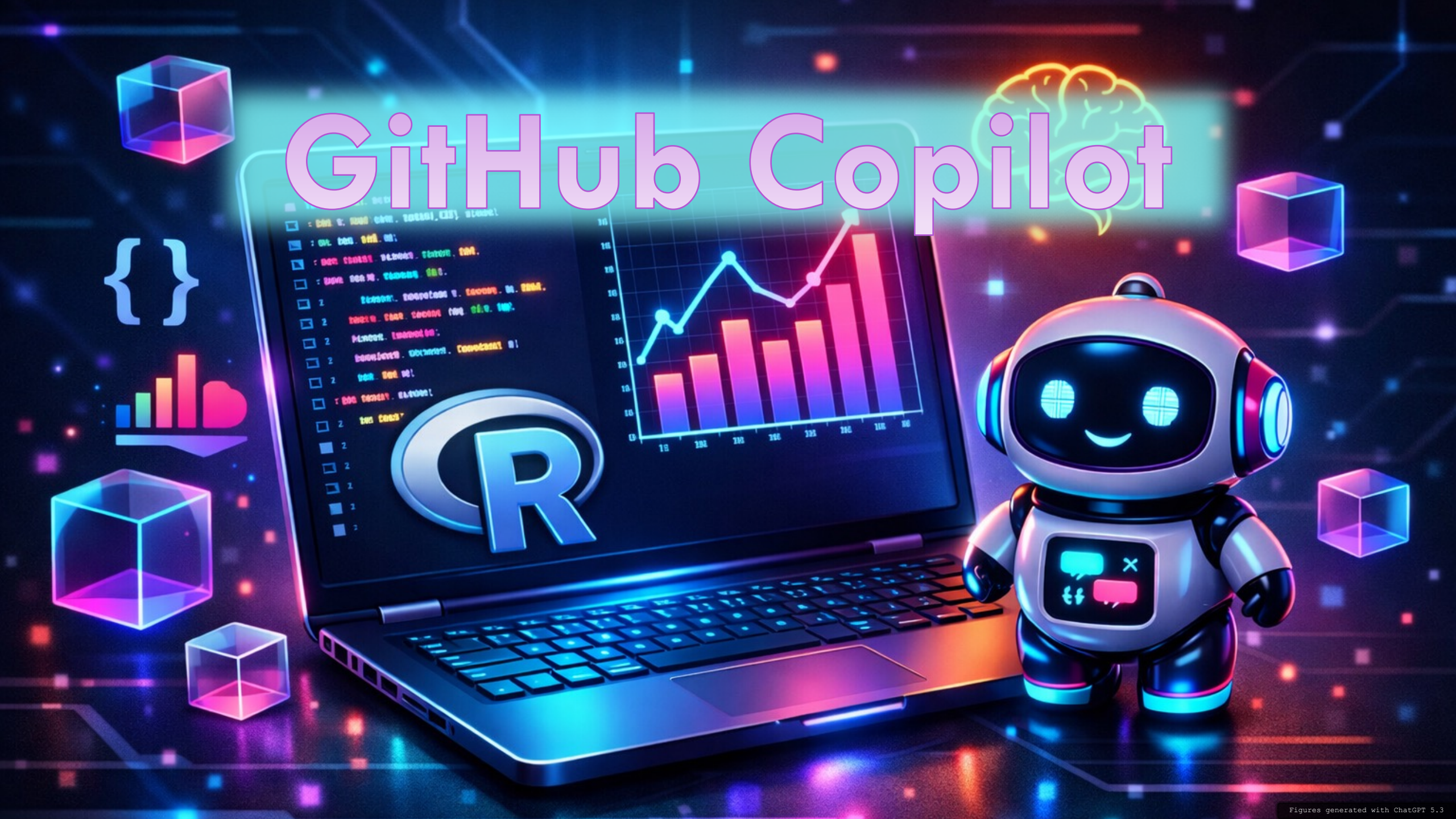
OFFENES PROBLEM BEIM PROMPT ENGINEERING

„Wer mehr zum Thema weiß, kann den Output besser beurteilen" *(Nestler et al., 2026)*

- Inhaltliches Wissen, Methoden/Statistikwissen, Programmierkenntnisse sind nötig, um KI-Output optimal zu prüfen
- **KI verstärkt vorhandene Kompetenz – ersetzt sie nicht**

  → wer `lme4` nicht kennt, weiß nicht ob `(1|id)` im Modell korrekt ist
- **Das ist kein Argument gegen KI** – sondern ein Argument für gute methodische Ausbildung *zusätzlich* zu KI-Nutzung

⚠️ *„Fehlererkennung sinkt bei geringer Fachkenntnis"*
→ das Risiko wächst genau dort, wo man sich am meisten auf KI verlässt *(Nestler et al., 2026)*


GitHub Copilot
R
Figures generated with ChatGPT 5.3

DATENSCHUTZ & VERANTWORTUNGSVOLLER KI-EINSATZ

Was niemals in externe KI-Systeme eingegeben werden darf:

- ❌ Personenbezogene Daten (Name, ID, Gesundheitsdaten)
- ❌ Sensible Studiendaten vor Veröffentlichung
- ❌ Urheberrechtlich geschützte Texte oder Daten Dritter
- ❌ Proprietäre Forschungsdaten oder interne Dokumente

Empfehlungen:

→ Hochschullösungen bevorzugen (z. B. uni-interne ChatGPT-Instanzen)

→ Bei Unsicherheit: lokale Modelle (z. B. Ollama + LLaMA) nutzen

→ Datenschutzhinweise der eigenen Institution prüfen

(Sperl et al., 2026)

Figures generated with ChatGPT 5.3

GITHUB COPILOT IN RSTUDIO – DATENSCHUTZ*

Wie Autocompletion funktioniert*

- Codekontext (aktuelle Datei + Umgebung) wird **verschlüsselt** an GitHub-Server übertragen
- Prompt wird **sofort nach der Vorschlagsgenerierung gelöscht**
- Wird **nicht für das Training** von Sprachmodellen verwendet (explizit in den Terms, März 2026)
- Datensätze im R-Environment werden **nicht übertragen** – nur sichtbarer Code

Personenbezogene Daten – Praktische Regeln

- ✅ Pseudonymisierte Variablennamen → geringes Risiko
- ✅ Algorithmischer Code ohne echte Werte → geringes Risiko
- ⚠️ Echte Namen / IDs / Diagnosen in Kommentaren → vermeiden
- ⚠️ Hardcodierte personenbezogene Werte im Testcode → vermeiden

⚠️ Vorsicht: Agenten (z.B. in Positron)

- Prompts werden bei Agenten-Tools **dauerhaft gespeichert**
- Agenten entscheiden **autonom,** welcher Kontext übermittelt wird (können auf lokale Dateien zugreifen)

Quellen (clickable)
- GitHub Copilot Product Specific Terms (deprecated 5. März 2026)
- GitHub Generative AI Services Terms
- GitHub Data Protection Agreement
- GitHub Acceptable Use Policies

```
load("MyData.rda")
names(data) # id time group age gender procrastination self_efficacy gpa
library(lme4)
model <- lmer(procrastination ~ time
```

Kontext hilft

```
load("MyData.rda")
names(data) # id time group age gender procrastination self_efficacy gpa
library(lme4)
model <- lmer(procrastination ~ time * group
```

Durch “Tab” Vorschlag annehmen

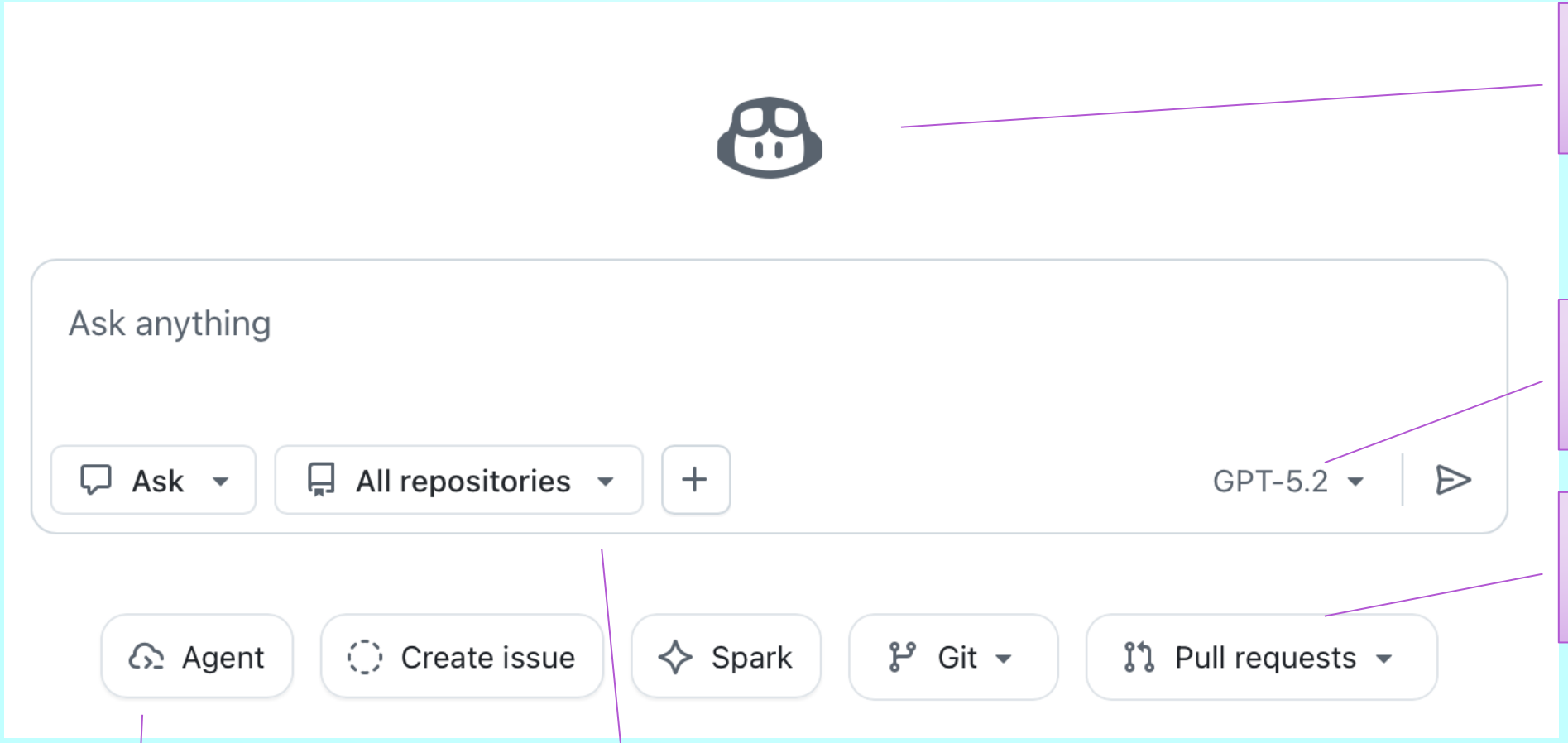
```
load("MyData.rda")
names(data) # id time group age gender procrastination self_efficacy gpa
library(lme4)
model <- lmer(procrastination ~ time * group + age
```

Mehrstufiger Prozess: Human in the Loop

```
load("MyData.rda")
names(data) # id time group age gender procrastination self_efficacy gpa
library(lme4)
model <- lmer(procrastination ~ time * group + age + gender
```

```
load("MyData.rda")
names(data) # id time group age gender procrastination self_efficacy gpa
library(lme4)
model <- lmer(procrastination ~ time * group + age + gender + self_efficacy + gpa + (1 | id), data = data)
```

GITHUB COPILOT: CHAT AUF GITHUB.COM



Friendly animated Copilot

Modellauswahl (ggf. Begrenzt bei free)

Kann Git-Commands ausführen

Agentenmodus begrenzt nutzbar in Education Version

Zugriff auf online Repositories

Ausblick

Figures generated with ChatGPT 5.3

CALLBACK: WAS IST EIN AGENT?

LLM + Tools + Planung + Autonomie:

ReAct-Schleife: Reason → Act → Observe → Repeat

Die drei Stufen in `Positron` via GitHub Copilot:

Stufe	Modus	Was passiert?
1	Autocomplete	Passiv, vervollständigt beim Tippen
2	Chat (Ask)	Interaktiv, kein Dateizugriff
3	Agent	Liest/schreibt Dateien, nutzt Terminal, plant mehrstufig

POSITRON – WAS IST DAS?

- **Positron** = moderner R/Python-Editor von **Posit** (RStudio-Nachfolger)
- **Rstudio:** Beim reinen R-Programmieren noch deutlich einfacher (Pakete, etc.)

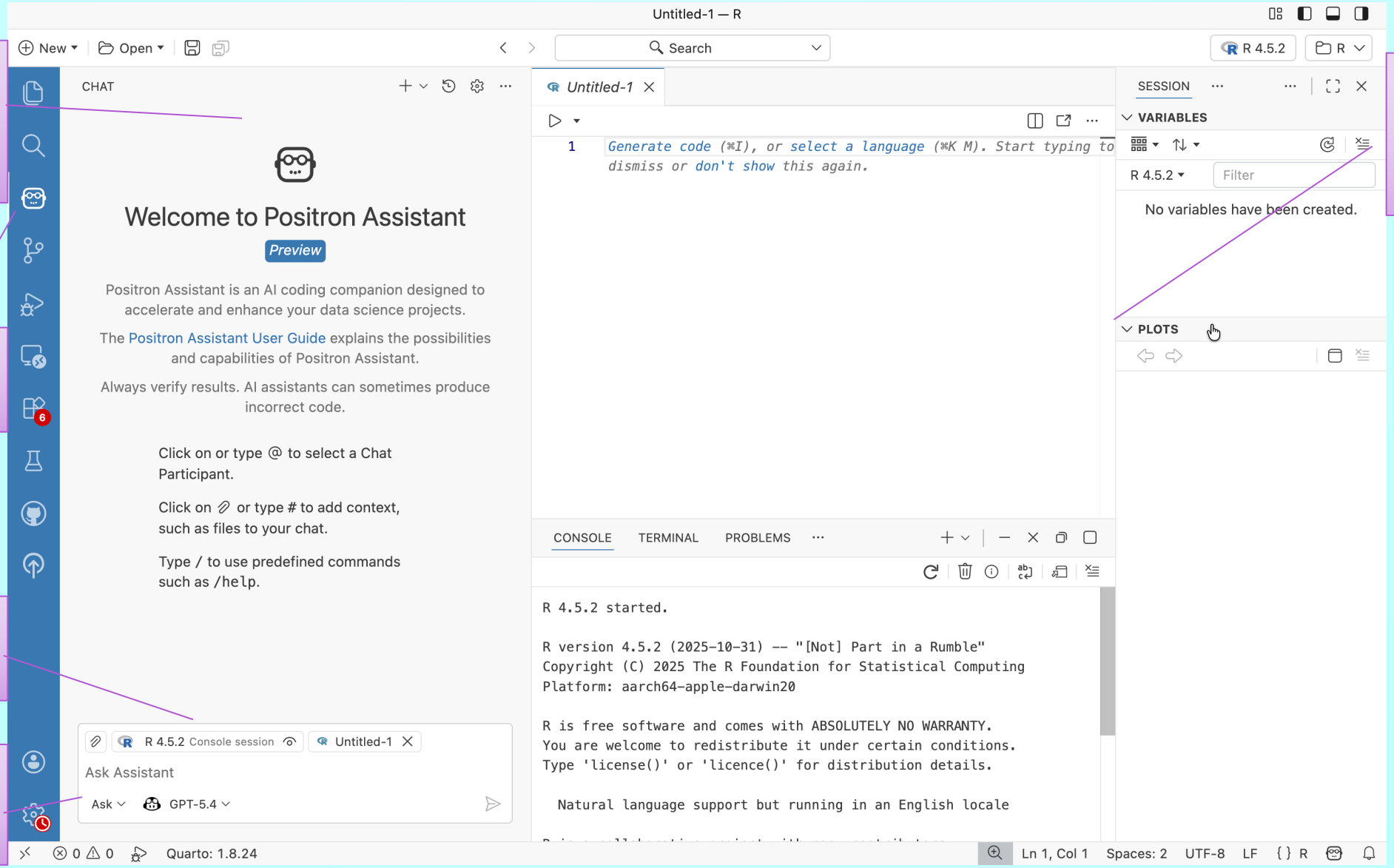
	RStudio	Positron
Stabilität	✅ ausgereift	⚠️ Beta
KI-Integration	Copilot (Autocomplete)	Copilot: Chat + Agent + Autocomplete
Python	eingeschränkt	✅ vollwertig
Oberfläche	vertraut	moderner, VS-Code-ähnlich
Paketmanagement	✅ integriert (Install-Button)	⚠️ manueller
Quarto	✅	✅
Kostenlos	✅	✅

Neue Spalte im Vergleich zu RStudio

Copilot

Chatfenster im Editor

Ask = “nur Chat” kein Agent

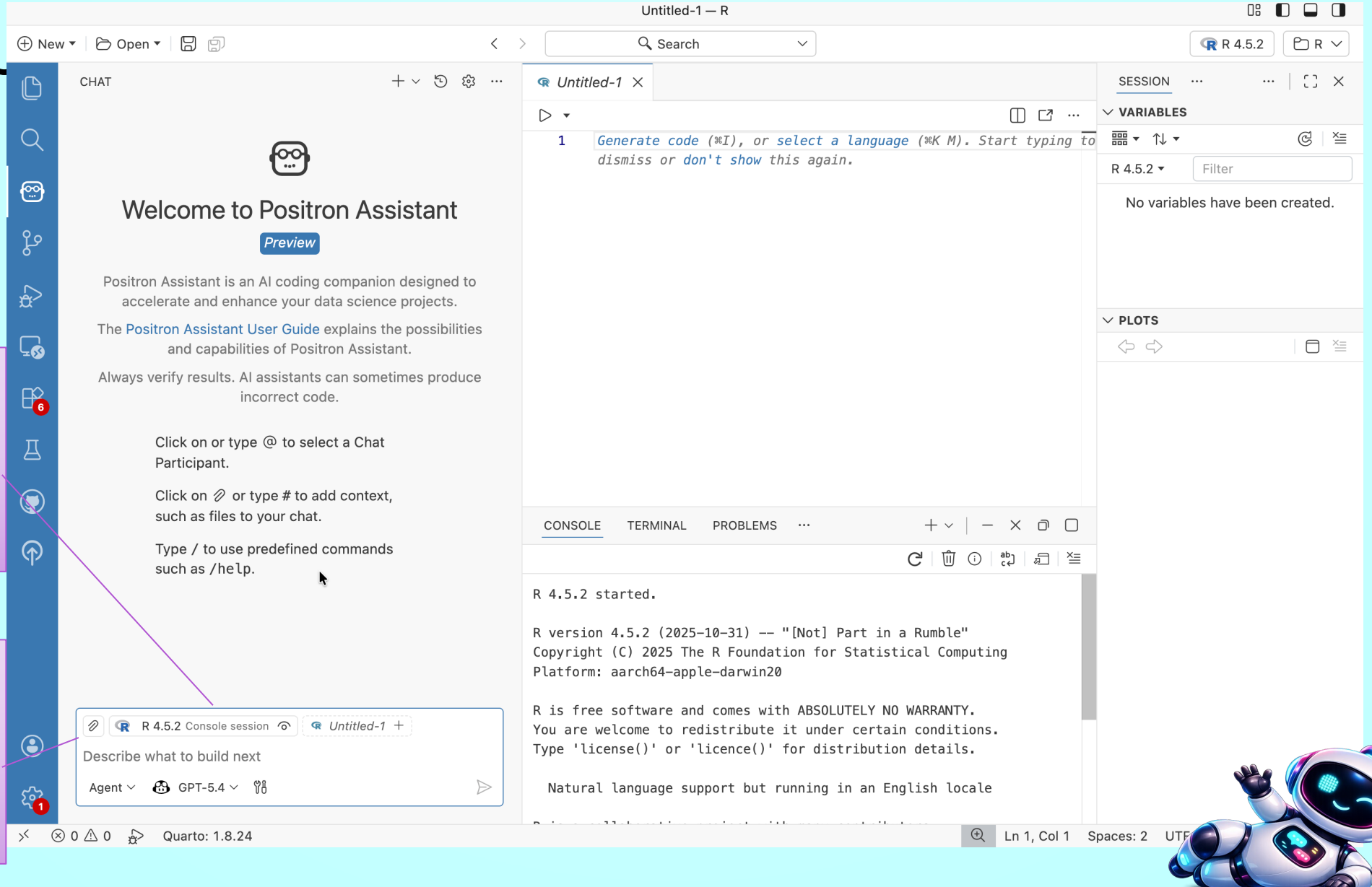


Normalen 4 Kacheln wie bei RStudio

AU

Chatfenster im Editor nutzen, um Agenten Auftrag/Ziel zu geben

Agent = Copilot darf als Agent agieren, fragt aber vorher nach



ZUSAMMENFASSUNG

🧠 **LLMs als Basis aller Assistenzsysteme** – probabilistische Systeme ohne Weltmodell; Halluzination ist kein Bug sondern Systemlogik

⚙️ **Prompt Engineering** – 6 Bausteine + erweiterte Techniken (Refinement, Meta-Prompting, …) erhöhen Outputqualität systematisch

🔬 **KI im R-Workflow** – von Datenaufbereitung über `R-Pakete` bis zum Quarto-Report; KI als Assistenz, nie als (kompletter) Autopilot

🛠️ **Tools kennen** – Chat (Claude/ChatGPT) für Erklärungen & Code, Copilot für Editor-Integration, Positron als Zukunft

✅ **Human in the Loop** – Fachkompetenz ist der entscheidende Multiplikator:
"wer Pakete und Modelle kennt, bekommt besseren Code" (Nestler et al., 2026)

QUELLEN

Bamil, V. (2025). Vibe Coding: Toward an AI-Native Paradigm for Semantic and Intent-Driven Programming. *arXiv*. https://doi.org/10.48550/arXiv.2510.17842

Binz, M., Alaniz, S., Roskies, A., Aczel, B., Bergstrom, C. T., Allen, C., Schad, D., Wulff, D., West, J. D., Zhang, Q., Shiffrin, R. M., Gershman, S. J., Popov, V., Bender, E. M., Marelli, M., Botvinick, M. M., Akata, Z., & Schulz, E. (2025). How should the advancement of large language models affect the practice of science? *Proceedings of the National Academy of Sciences, 122*, e2401227121. https://doi.org/10.1073/pnas.2401227121

Blackwell, R. E., Barry, J., & Cohn, A. G. (2024). *Towards reproducible LLM evaluation: Quantifying uncertainty in LLM benchmark scores* (arXiv:2410.03492). arXiv. https://arxiv.org/abs/2410.03492

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., … Amodei, D. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems, 33*, 1877–1901.

Bubeck, S., Szegedy, C., Tassiulas, L., et al. (2024). Response generated by large language models depends on the structure of the prompt. *Journal of Medical Internet Research, 26*, e51866. https://doi.org/10.2196/51866

He, J., Rungta, M., Koleczek, D., Sekhon, A., Wang, F. X., & Hasan, S. (2024). *Does Prompt Formatting Have Any Impact on LLM Performance?* arXiv. https://doi.org/10.48550/arXiv.2411.10541

Liu, X., Chen, Y., & Li, J. (2025). *A systematic survey of prompt engineering in large language models: Techniques and applications* (arXiv:2402.07927). arXiv. https://arxiv.org/abs/2402.07927

Lott, M. (2025). Tracking AI: Monitoring artificial intelligence [Dataset/Website]. *Maximum Truth Project*. https://www.trackingai.org/home

Nestler, S., Humberg, S., Debelak, R., Heck, D. W., Henninger, M., Voelkle, M. C., Frick, S., Irmer, J. P., Scharf, F., & Frey, A. (2026). *Automating the scientist? Methodische Perspektiven auf die Nutzung von KI in der psychologischen Forschung* [Preprint]. https://doi.org/10.31234/osf.io/dqxsu_v1

Patil, R., Heston, T. F., & Bhuse, V. (2024). Prompt Engineering in Healthcare. *Electronics, 13*(15), 2961. https://doi.org/10.3390/electronics13152961

Poole, D. L., & Mackworth, A. K. (2017). *Artificial intelligence: Foundations of computational agents*. Cambridge University Press.

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2025). A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. *arXiv*. https://doi.org/10.48550/arXiv.2402.07927

Schulhoff, S., Ilie, M., Balepur, N., Kahadze, K., Liu, A., Si, C., Li, Y., Gupta, A., Han, H., Schulhoff, S., Dulepet, P. S., Vidyadhara, S., Ki, D., Agrawal, S., Pham, C., Kroiz, G., Li, F., Tao, H., Srivastava, A., … Resnik, P. (2025). The Prompt Report: A Systematic Survey of Prompt Engineering Techniques. *arXiv*. https://doi.org/10.48550/arXiv.2406.06608

Sarkar, A., & Drosos, I. (2025). Vibe coding: Programming through conversation with artificial intelligence. *arXiv*. https://doi.org/10.48550/arXiv.2506.23253

Shojaee, P., Mirzadeh, I., Alizadeh, K., Horton, M., Bengio, S., & Farajtabar, M. (2025). The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity. *arXiv*. https://doi.org/10.48550/arXiv.2506.06941

Sperl, M. F. J., Baumgärtner, L., Bach, K. M., Bamberg, C., Behlau, C., Bergmann, B., Bienefeld, M., Bleckmann, E., Danböck, S. K., Dreston, J. H., Eckardt, V. C., Frick, S., Friehs, M.-T., Handke, L., Hein, I., Hutmacher, F., Irmer, J. P., Kause, A., Kern, M., … Neef, N. E. (2026). *Künstliche Intelligenz bei Abschlussarbeiten, Dissertationen und Habilitationsschriften in der Psychologie* [Preprint].

Su, J. (2023, August 1). *Master the perfect ChatGPT prompt formula (in just 8 minutes)* [Video]. YouTube. https://www.youtube.com/watch?v=jC4v5AS4RIM

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. ukasz, & Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems, 30*.

Walter, Y. (2024). Embracing the future of Artificial Intelligence in the classroom: The relevance of AI literacy, prompt engineering, and critical thinking in modern education. *International Journal of Educational Technology in Higher Education, 21*(1), 15. https://doi.org/10.1186/s41239-024-00448-3

White, J., Fu, Q., Hays, S., Sandhu, J., Olea, C., Hays, M., Elnashar, A., Goyal, A., & Schmidt, D. C. (2023). *A prompt pattern catalog to enhance prompt engineering with ChatGPT*. arXiv. https://arxiv.org/abs/2302.11382

QUELLENAUSZUG FÜR EXKURS ZU IQ VON KI

- Gigazine. (2024, 7. März). Report that Claude-3, who appeals to exceed GPT-4, achieved IQ over 100 for the first time in AI. Gigazine. https://gigazine.net/gsc_news/en/20240308-claude-3-chat-gpt-iq-test/
- Lott, M. (2024). AIs ranked by IQ: AI passes 100 IQ for first time, with release of Claude-3. Maximum Truth. https://www.maximumtruth.org/p/ais-ranked-by-iq-ai-passes-100-iq
- Lott, M. (2025). *Tracking AI: Monitoring artificial intelligence* [Dataset/Website]. Maximum Truth Project. https://www.trackingai.org/home
- Vellum AI. (2025). GPT-5.x / GPT-5.2 benchmarks (explained). Vellum.ai. https://www.vellum.ai/blog/gpt-5-2-benchmarks


Ende
R
Figures generated with ChatGPT 5.3