# Verhält sich die Schul-KI u.U.anders als das Original?

Übersicht von **möglichen** LLM-Unterschieden in behördlichen & schulischen Umgebungen

Autor: Thomas Jörg, 6.3.2026

## Ausgangslage

**Lehrkräfte und andere Beschäftigte im öffentlichen Dienst berichten regelmäßig, dass ein von der Behörde bereitgestelltes Sprachmodell (z. B. „ChatGPT 5“ auf telli) sich spürbar anders verhält als die öffentliche Originalversion desselben Modells. Die Antworten wirken steifer, weniger kreativ – und sind manchmal sogar fehlerhaft.**

> *Fazit: Ja, das ist absolut plausibel – und fast schon erwartbar: Die behördlich gehostete Version kann sich deutlich anders verhalten, auch wenn sie auf demselben oder einem sehr ähnlichen Basismodell aufsetzt. Der Grund ist die zugrundeliegende Konfiguration und Governance. Viele technische Ebenen können das beobachtbare Verhalten verändern – vom System-Prompt über Quantisierung bis zum Kontextfenster-Management.*

## Im Detail: Ebenen des Unterschieds

1. **System-Prompt und Rollen-Vorgaben („Leitplanken“)**

Bevor eine Nutzereingabe das Modell erreicht, steht ein unsichtbarer System-Prompt voran, z. B.: „Du bist ein schulischer Assistent. Antworte sachlich, politisch neutral, vermeide kreative Ausschweifungen.“ Dieser Prompt hat Vorrang vor der Nutzereingabe (sog. Developer Instructions). Schon eine veränderte Rollenbeschreibung, eine Vorgabe zu Gendern oder politischer Zurückhaltung verschiebt das Antwortverhalten deutlich. Für Nutzende sieht es aus, als sei „das Modell anders“ – es ist aber nur die Steuerung.

2. **Zusätzliche Sicherheits- und Compliance-Filter**

Behördliche Umgebungen schalten häufig strengere Moderationspipelines vor und nach das eigentliche Modell: Content-Filter, Blacklists, domainspezifische Heuristiken (z. B. „Schulkontext“), Logging- und Policy-Schichten. Diese können Antworten abschneiden, umformulieren oder komplett blockieren. Auch Azure/OpenAI-Dienste dokumentieren, dass Antworten und Eingaben durch solche Filter beeinflusst werden. Das Ergebnis: Antworten wirken defensiver, ausweichender oder pauschaler als beim Original.

3. **API-Anbindung statt Webprodukt – fehlende Features**

Die öffentliche Web-Version (z. B. chatgpt.com) ist ein hochoptimiertes Produkt mit zahlreichen Hilfsprozessen im Hintergrund: Websuche, Code-Interpreter, Bilderzeugung (DALL-E), automatische Bilderkennung, Dateiverarbeitung, Memory-Funktionen, Canvas-Modus u. v. m. Behördeninstanzen basieren hingegen meist auf der reinen API-Schnittstelle und bieten nur Text-Chat. Viele Fähigkeiten, die Nutzer vom Original kennen, fehlen daher schlicht.

4. **Inferenz-Parameter (Temperatur, Top-p, Token-Limits)**

Antwortstil und Varianz hängen stark an Sampling-Parametern. Über die API kann die Temperatur (Kreativität) eingestellt werden: Bei 0 wirkt das Modell deterministisch und „roboterhaft“, bei 0,7–1,0 (typisch für die Web-Version) deutlich lebendiger. Ebenso beeinflussen Top-p, Frequency Penalty und maximale Antwortlänge das Verhalten. Wenn Behörden konservative Werte setzen, sinkt die wahrgenommene Kreativität und Detailtiefe.

5. **Quantisierung und Modellkompression**

Wird das Modell auf eigenen Servern (On-Premise) betrieben, muss enorme Rechenpower bereitgestellt werden. Um Kosten und Hardware zu sparen, werden Modelle häufig quantisiert: Die mathematische Präzision wird z. B. von FP16 (16-Bit) auf INT8 oder INT4 (4-Bit) reduziert. Das Modell wird kleiner und schneller, verliert aber an „Feinschliff“ – häufigere Flüchtigkeitsfehler in der Logik, subtilere Sprachnuancen gehen verloren. Auch der Einsatz distillierter (verkleinerter) Modellvarianten ist möglich, die zwar kosteneffizienter, aber weniger leistungsfähig sind.

6. **Modellversionen und Update-Rückstand**

OpenAI aktualisiert die Web-Version laufend per „Rolling Updates“. Behördeninstanzen werden hingegen oft auf einer spezifischen, stabilen Version „eingefroren“ (z. B. gpt-5-0314), um Vorhersehbarkeit und Zertifizierung zu gewährleisten. Während das öffentliche Modell bereits verbessert wurde, bleibt die Behörden-Version auf dem Stand der Installation. Subtile Änderungen in den Gewichten oder Safety-Layern machen sich im Alltag schnell bemerkbar.

7. **Fine-Tuning und domänenspezifische Anpassung**

OpenAI bietet Fine-Tuning und Reinforcement Fine-Tuning an, um ein Basismodell auf bestimmte Aufgaben und Antwortmuster zuzuschneiden. Behörden oder Dienstleister können das Modell mit eigenen Beispieldialogen nachtrainieren (z. B. „immer auf DSGVO hinweisen“, „pädagogisch formulieren“). Schon relativ kleine Fine-Tuning-Datensätze können das Verhalten weit über den eigentlichen Spezialbereich hinaus verschieben.

8. **Kontextfenster-Management**

Das Kontextfenster (die maximale Anzahl Tokens, die ein Modell gleichzeitig verarbeiten kann) ist eine der zentralen Betriebsgrößen. Die Modellgrenze selbst ist fest vorgegeben – aber Betreiber konfigurieren, wie viel davon tatsächlich genutzt wird. Die Auswirkungen sind vielfältig:

- **Früheres „Vergessen“:** Wird der verfügbare Kontext klein gehalten oder aggressiv gekürzt, fallen ältere Gesprächsteile heraus. Das Modell wirkt inkonsistent oder wiederholt Fragen.
- **Verdrängung durch Richtlinien:** Umfangreiche Systemanweisungen, Sicherheitsregeln und Dokumentauszüge belegen Kontext-Tokens, die dann für die eigentliche Nutzereingabe fehlen. Antworten werden steifer und weniger kontextsensitiv.

> *Merke: Nicht nur welches Modell läuft, sondern auch wie der Kontext verwaltet wird, bestimmt das beobachtbare Verhalten.*

# Ergänzende Faktoren

## UI- und Produktlogik

Die öffentliche ChatGPT-Oberfläche verarbeitet Nutzereingaben häufig vor, bevor sie an das Modell gehen: automatische Prompt-Erweiterungen, Kontextkomprimierung, Routing an spezialisierte Sub-Modelle (z. B. für Code oder Bildanalyse). In einer Behörden-API fehlt diese Produktlogik – der Prompt geht „roh“ ans Modell, was zu weniger optimierten Ergebnissen führen kann.

## Kosten- und Latenz-Optimierung

Mehr Kontext und größere Modelle kosten mehr Rechenaufwand und erhöhen die Latenz. Betreiber haben daher einen starken Anreiz, den nutzbaren Kontext zu begrenzen, Prompt Caching einzusetzen oder auf kleinere Modellvarianten auszuweichen. All dies beeinflusst die Antwortqualität.

## Fehlende Transparenz der Konfiguration

In der Praxis erfahren Nutzende fast nie, welche System-Prompts, Filter, RAG-Quellen, Temperaturwerte oder Modellversionen tatsächlich im Einsatz sind. Die Aussage „Das ist ChatGPT 5“ reicht technisch nicht aus – für das beobachtbare Verhalten sind mindestens viele Konfigurationsebenen relevant.

# Vergleichsübersicht

| Merkmal | OpenAI Original (Web) | Behörden-Instanz |
|---|---|---|
| Kreativität / Temperatur | Dynamisch, eher hoch (ca. 0,7–1,0) | Oft konservativ / niedrig (0–0,3) |
| Zusatzfunktionen | Websuche, Code-Interpreter, DALL-E, Bilderkennung, Memory, Canvas | Meist nur reiner Text-Chat |
| Modellpräzision | Maximale Rechenleistung (FP16/BF16) | Ggf. reduziert durch Quantisierung (INT4/8) oder Destillation |
| Sicherheitsfilter | Standard-OpenAI-Filter | Zusätzliche, oft strengere Behörden-Filter |
| Modellversion | Laufend aktualisiert (Rolling Updates) | Oft auf stabilem Release eingefroren |
| Kontextfenster | Großzügig, automatisch verwaltet | Häufig eingeschränkt, Richtlinien belegen Tokens |
| Wissensquellen | Allgemeines Vorwissen + Websuche | Ggf. RAG mit internen Dokumenten |
| Fine-Tuning | Standard-Training von OpenAI | Möglicherweise domänenspezifisch nachtrainiert |
| System-Prompt | OpenAI-Standard | Behördenspezifisch (Rollen, Regeln, Tonfall) |
| Antwortlänge | Flexibel, oft ausführlich | Häufig begrenzt (Kosten/Latenz) |