1. Lokale SLM und selbst gehostete LLMs | Oktober 2026

Lokal installierte SLM / LLM ...

Gemma

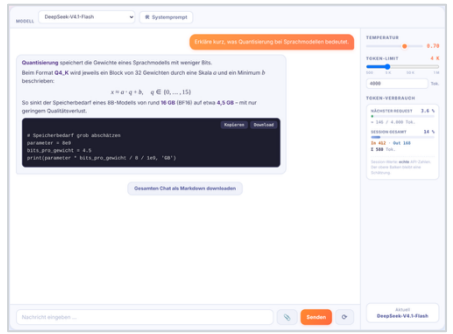
Qwen 3.8 27B

Open Weight
GPT-OSS Models

MISTRAL AI_

Ollama

LM Studio



Thomas Jörg, LFT KI Baden-Württemberg, Intel Certified Trainer, Dozent DQR6 IHK

... versus ‚Big Models':

KIMI K3.1 Pro

DeepSeek V4

1. Lokale SLM und selbst gehostete LLMs | Oktober 2026

Lokal installierte SLM / LLM ...

Gemma

Qwen 3.8 27B

Open Weight
GPT-OSS Models

MISTRAL AI_

- Warum lokal? Rechtlicher Rahmen für die Schule
- Software & Frontend: Paradigmenwechsel – selbst bauen
- Welche Hardware braucht man? (NVIDIA, AMD, Intel, Apple)
- Welche Modelle sind aktuell nützlich?
- Quantisierung & Speicherbedarf – die Theorie dahinter
- Wie misst man die Qualität von LLMs?
- Use Cases, Grenzen und Ausblick

7. Lokale SLM und selbst gehostete LLMs | Oktober 2026

Use cases: Wozu sind diese SLM / LLM im Unterricht geeignet?

Gemma

Qwen 3.8 27B

Open Weight
GPT-OSS Models

MISTRAL AI_

- Chat für Schüler.
- Übersetzungen Deutsch – Französisch – Englisch – Italienisch …
- RAG: PDFs/Ebooks/Mails/Tabellen zusammenfassen
- RAG: Texte von eingebetteten Webseiten zusammenfassen
- Coding Assistant: Programmier-Unterstützung
- *Audiofiles transkribieren*
- *Agentic systems: Dokumentenvergleich, Feedback, eigene Tools bauen*

1. Lokale SLM und selbst gehostete LLMs | Oktober 2026

Die Handlungsempfehlung der Bildungsminister-Konferenz:

https://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/2024/2024_10_10-Handlungsempfehlung-KI.pdf

Das Dokument empfiehlt ausdrücklich, Schüler:innen im Umgang mit KI zu unterrichten. Der Unterricht über KI wird als eine neue Schlüsselkompetenz für das 21. Jahrhundert angesehen. Die Empfehlungen betonen, dass Schüler:innen durch den Unterricht in die Lage versetzt werden sollen, KI-Technologien kritisch und reflektiert anzuwenden. Dies soll ihnen helfen, als digitale Bürger:innen souverän zu agieren.



Gemma | Qwen 3.8 27B | Open Weight GPT-OSS Models | MISTRAL AI_

1. Lokale SLM und selbst gehostete LLMs | Oktober 2026

Die Rechtlichen Rahmenbedingungen des EU AI Act:

https://artificialintelligenceact.eu/

In Kraft seit 1.8.2024, gilt gestuft (Verbote & KI-Kompetenz seit 2.2.2025). Die wichtigsten Punkte:

- KI-Systeme werden in Risikostufen klassifiziert (minimal, begrenzt, hoch, unakzeptabel).
- Social Scoring und biometrische Massenüberwachung in Echtzeit sind verboten.
- KI zur Bewertung von Prüfungen/Lernstand ist Hochrisiko-KI – Pflichten nach „AI-Omnibus“ erst ab 2.12.2027.
- Transparenzpflicht: Nutzer müssen informiert werden, wenn sie mit einer KI interagieren.



Gemma | Qwen 3.8 27B | Open Weight GPT-OSS Models | MISTRAL AI_

1. Lokale SLM und selbst gehostete LLMs | Oktober 2026



Warum nicht die großen Cloud-Modelle (GPT, Gemini, Claude)?

- Bei den ‚Big Three' (GPT/Claude/Gemini) problematisch: Datenschutz / Privacy
- Für cloudbasierte Modelle ist immer eine Internetverbindung nötig.
- Für viele Aufgaben ist das umfassende Wissen der ‚Großen' nicht unbedingt nötig
- Leistungssprung der offenen Modelle: Qwen3.8-27B (08/2026) erreicht laut Hersteller in Agenten- und Coding-Benchmarks das Niveau aktueller Cloud-Modelle – auf einer einzelnen 24-GB-GPU (Q4 ≈ 17 GB).

- **EU AI Act verstärkt Datenschutzpflichten: Der regulatorische Druck auf Schulen im europäischen Bildungssektor wächst weiter.**

Teil 1: Was ist erlaubt und gewünscht? Stand 2024

„Die Schule hat einen eigenen Zugang, API-Schlüssel und AVV mit dem Anbieter abgeschlossen und erstellt mit Oberstufenschülern eine eigene Plattform zur Nutzung der KI via API. Sind die Spielregeln klar, ist eine Nutzung mit älteren Schülerinnen und Schülern ab 16 Jahren möglich. Die Risiken sind begrenzt.

Die Schule hat lokal als App auf Computern oder Tablets laufende KI-Anwendungen. Eine Nutzung ist für Schülerinnen und Schüler ohne Risiken möglich, da keine Daten an Dritte abfließen. Trotzdem sollte es zuvor abgesprochene Spielregeln geben.“

https://datenschutz-schule.info/tag/chatgpt/

Teil 2: Was ist erlaubt und gewünscht? Stand Oktober 2026

Lokal heißt nicht rechtsfrei. Die Daten bleiben im Haus, aber für Prompts, Ausgaben, Logins und Protokolle gilt weiter die DSGVO. Nötig sind ein Eintrag im Verarbeitungsverzeichnis, technische und organisatorische Maßnahmen, ein Löschkonzept und bei Minderjährigen ggf. eine Datenschutz-Folgenabschätzung.

Rote Linie: KI darf nicht über Noten oder Zugang entscheiden. Sobald sie das tut, ist sie Hochrisiko-KI mit vollen Pflichten.

KI-Kompetenz (Art. 4): Schulen müssen Maßnahmen ergreifen, damit ihr Personal KI-Kompetenz aufbaut. Seit Juli 2026 abgeschwächt: fördern statt sicherstellen.

Transparenz (Art. 50, seit 2.8.2026): Nutzer:innen müssen erkennen, dass sie mit KI interagieren. Wer ein eigenes Frontend baut, kann dabei selbst zum „Anbieter“ werden.

Rechtslage diffus: Ob Modellgewichte personenbezogene Daten enthalten, ist umstritten (Hamburg 2024: nein; EDSA: Einzelfallprüfung). Zusätzlich kommen Länderregeln mit Freigabeverfahren (z. B. Entwurf § 98e HmbSG).

Was kann man mit lokalen Sprachmodellen (sonst noch) machen?

- ***RAG: „Retrieval-Augmented Generation“***
  Problem von Halluzinationen & beschränktem Erfahrungs-schatz wird gelöst: LLMs werten Informationsquellen aus.

- ***Agentic Systems***
  ReAct (Reasoning-Acting): Sprachmodelle als autonome Planungsinstrumente, orchestrieren weitere LLMs und Tools/Skills, die Aufgaben erledigen.

- **Innovatoren in der Forschung**
  deutliche Anzeichen bei AlphaFold, mathematischen Beweisen.

We are here

Level 1: Chatbots

Level 2: Reasoner, können Logik anwenden und Probleme lösen

Level 3: Agenten, können zusätzliche Aktionen ausführen

Level 4: Innovatoren, können neue Erfindungen machen

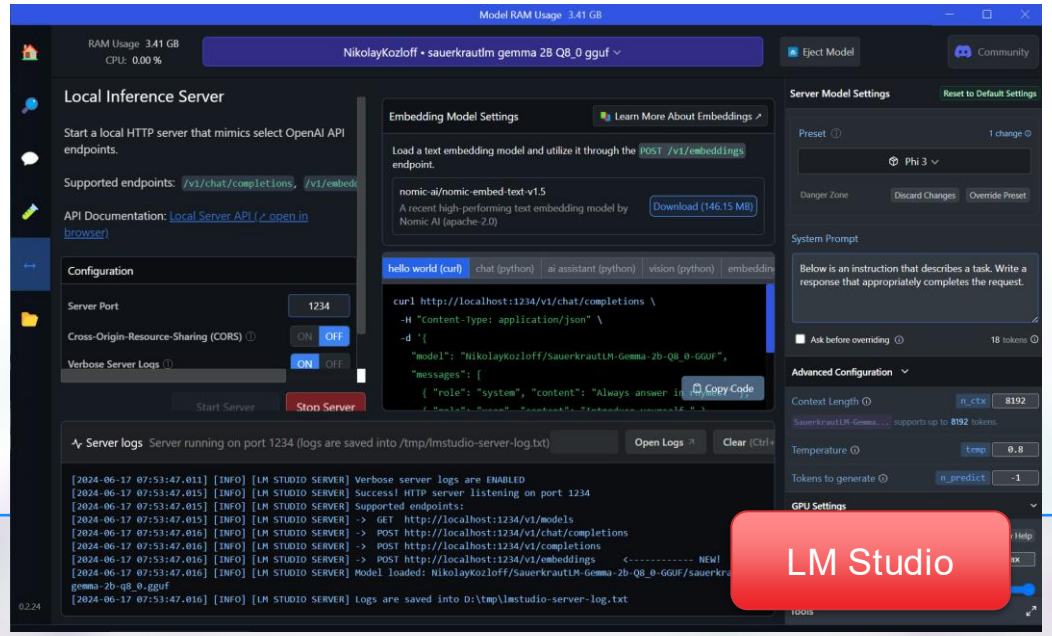
Level 5: Kann die Arbeit einer ganzen Organisation erledigen

https://openai.com/charter/

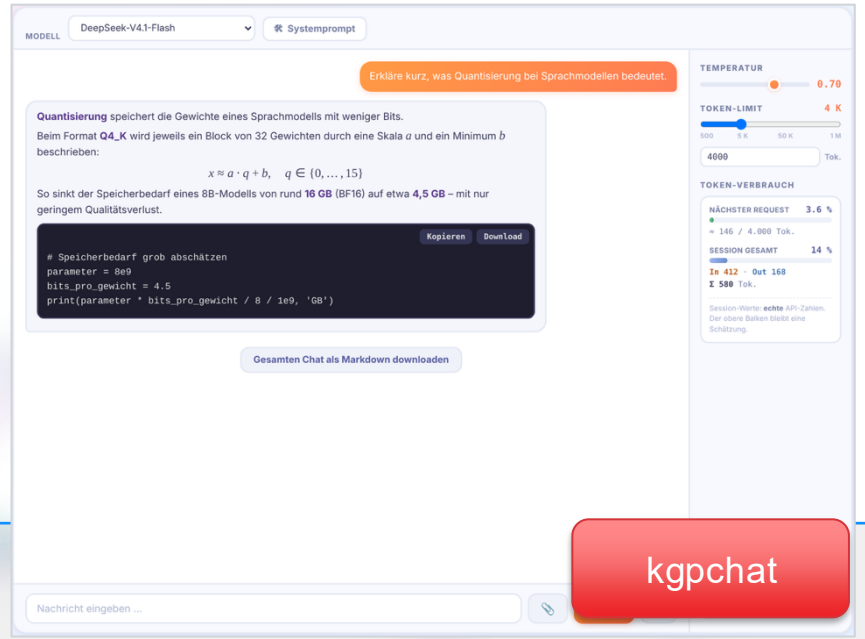
Lokale SLM und selbst gehostete LLMs | Oktober 2026

Was funktioniert ‚jetzt'? (Oktober 2026) *Ein Überblick*

- **Opensource-LLMs** *(werden auch von Apple/Microsoft verwendet)*
- **Software, um LLMs/SLMs zu hosten** *(z.B. LM Studio, Ollama)*
- **Frontends** *(zunehmend selbst gebaut, im Browser – z.B. kgpchat)*



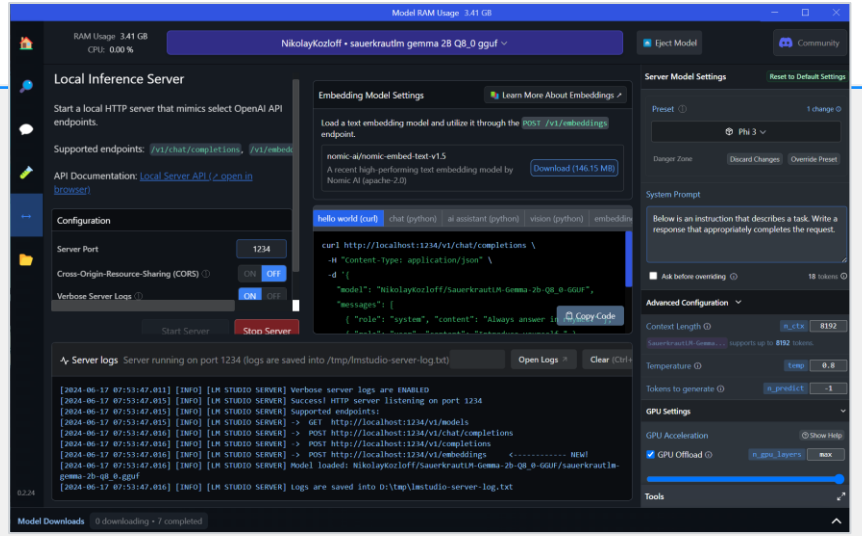
https://www.lmstudio.ai



https://iludis.de/Tools/kgpchat.html

Was funktioniert ‚jetzt'? (Oktober 2026) *Im Einzelnen: Lokale Installationen*

DEMO LM Studio



LM Studio:

Suche, Herunterladen, Grafikkarten-Offload, Server.

*Achtung: **Context length**!*

LM Studio als Server:

„Max Concurrent Predictions" (Continuous Batching seit v0.4.0), Berechtigungsschlüssel, ohne Oberfläche als Dienst „llmster"

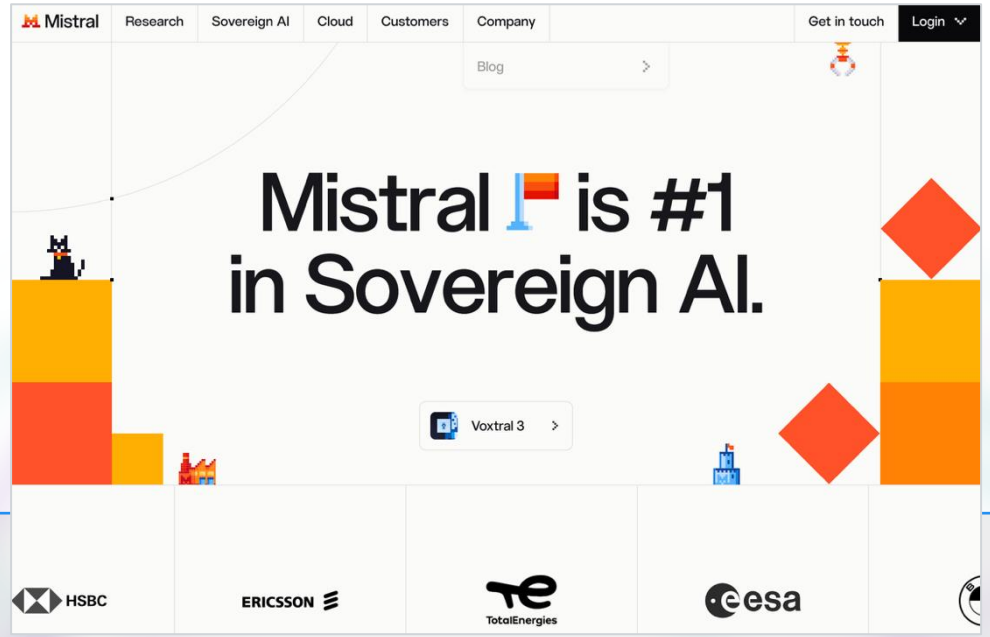
Modelle:

Hugging Face – der „riesige Basar für KI", direkt in LM Studio durchsuchbar

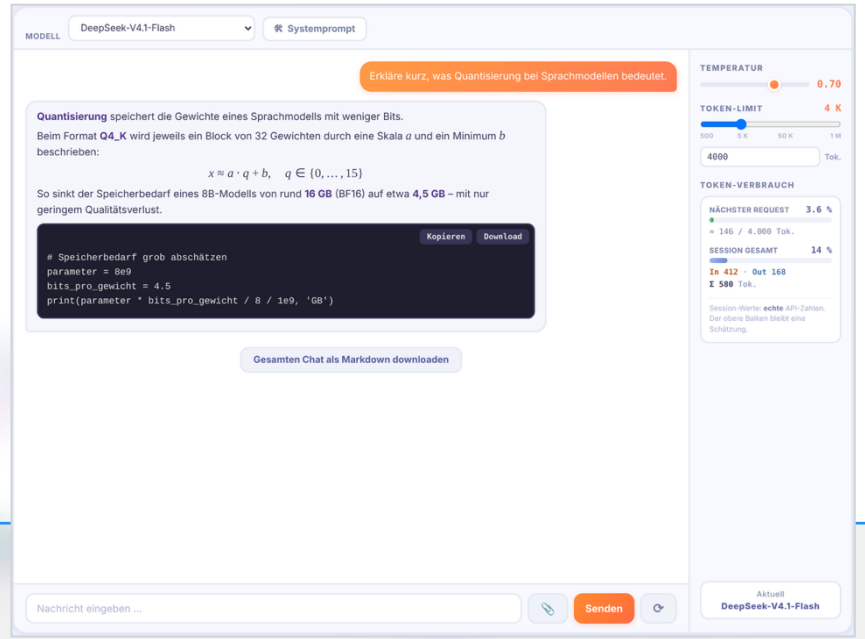
2. Lokale SLM und selbst gehostete LLMs | Oktober 2026

Was funktioniert ‚jetzt'? (Oktober 2026) *Ein Überblick*

- **Große Modelle hosten lassen: Mistral** – *Mistral large 2512 / Mistral large 4*
- **+ Frontend:** *eigenes Browser-Frontend (kgpchat)*
- **Kosten: ca.** *1,00 € / 5,00 € pro 1 Mio. Token (Ein-/Ausgabe)*



https://mistral.ai/



https://iludis.de/Tools/kgpchat.html

Lokale SLM und selbst gehostete LLMs | Oktober 2026



Frontend: *Warum keine installierten Chat-Clients mehr?*

Erfahrung aus dem Unterricht

- Schulrechner starten nach einer Woche – der Client will erst ein **Update** oder startet gar nicht: Unterrichtszeit ist weg.
- Desktop-Clients (Electron/Tauri) bringen eigene Browser-Engine, eigene Update-Zyklen und lokale Server mit – auf verwalteten Schulrechnern kaum zu pflegen.
- Ein schlankes, selbst gepflegtes Browser-Frontend läuft überall, wo ein Browser läuft.

Dokumentierte Sicherheitslücken

- **Jan** (Snyk, Feb./März 2025): lokaler API-Server ohne Authentifizierung und CSRF-Schutz → präparierte Webseite konnte Dateien schreiben, Kette bis zur Codeausführung (CVE-2025-2446/-2439/-2445/-2447); behoben.
- **AnythingLLM Desktop** (März 2026, CVE-2026-32626, CVSS 9,6): präpariertes Markdown in der Modellantwort → XSS → **Systembefehle**, weil Electron mit Node-Zugriff lief; behoben in 1.11.2.
- **Electron allgemein** (USENIX Security 2024, 109 Apps): mitgelieferte Chromium-Versionen hinken oft **Monate bis Jahre** hinterher.

Quellen: SecurityWeek 02.04.2025 (Jan/Snyk); Wiz/GitHub Advisory CVE-2026-32626; Ali et al., „Rise of Inspectron“, USENIX Security 2024

2. Lokale SLM und selbst gehostete LLMs | Oktober 2026

Trägt das Argument „Im Browser ist es sicherer“? – *geprüft*

✓ Was trägt

- Browser-Sandbox & Site Isolation: Eine Webseite hat keinen Zugriff auf Dateisystem oder Systembefehle.
- Ein XSS-Fehler bleibt im Browser in der Sandbox – derselbe Fehlertyp wurde in der Electron-App AnythingLLM zur Codeausführung.
- Browser hält die Schul-IT ohnehin aktuell; keine Installation, keine Adminrechte.
- API-Schlüssel bleibt auf dem Server (Proxy) – nie im Schülerbrowser.

✗ Was es nicht ersetzt

- LLM-Ausgaben sind ‚untrusted input‘: Markdown/HTML vor dem Einfügen säubern (z.B. DOMPurify) + Content-Security-Policy (OWASP LLM05:2025).
- CDN-Bibliotheken mit fester Version und Subresource Integrity (SRI) einbinden.
- Der Proxy-Server muss gepflegt werden (PHP-Updates, Kurspasswort, Rate-Limit).
- Datenschutz hängt am Anbieter hinter dem Proxy (AVV, Serverstandort).

Fazit: Das Argument trägt – der Browser senkt die ‚Fallhöhe‘ eines Fehlers deutlich.

Sauberes Output-Handling bleibt aber Pflicht.

Quellen: Chromium Site-Isolation-Design; OWASP Top 10 for LLM Applications 2025 (LLM05 Improper Output Handling); CVE-2026-32626

2. Lokale SLM und selbst gehostete LLMs | Oktober 2026

Fallbeispiel kgpchat: *Architektur eines schlanken Schul-Frontends*

Browser der Schüler:innen

kgpchat.html – eine einzige Datei

- Markdown, Formeln (KaTeX), Code mit Kopieren/Download
- Anhänge PDF/HTML/MD/TXT/CSV – Text wird **lokal im Browser** extrahiert
- Systemprompt, Temperatur, Token-Limit mit Zähler
- Chat-Export als Markdown

→

proxy.php (Schul-Webspace)

- Kurspasswort (zeitkonstanter Vergleich)
- Modell-Whitelist → Dropdown
- Kostenbremse: max. 2.048 Ausgabe-Tokens
- Rate-Limit: 40 Anfragen/Stunde/Sitzung
- Keine offenen CORS-Header
- API-Schlüssel **nur hier**
- Streaming wird durchgereicht

→

Modell-Anbieter

Mistral AI (EU)

mistral-large

***Ausbau:* LM Studio-Server**

im Schulnetz – spricht dieselbe OpenAI-kompatible API

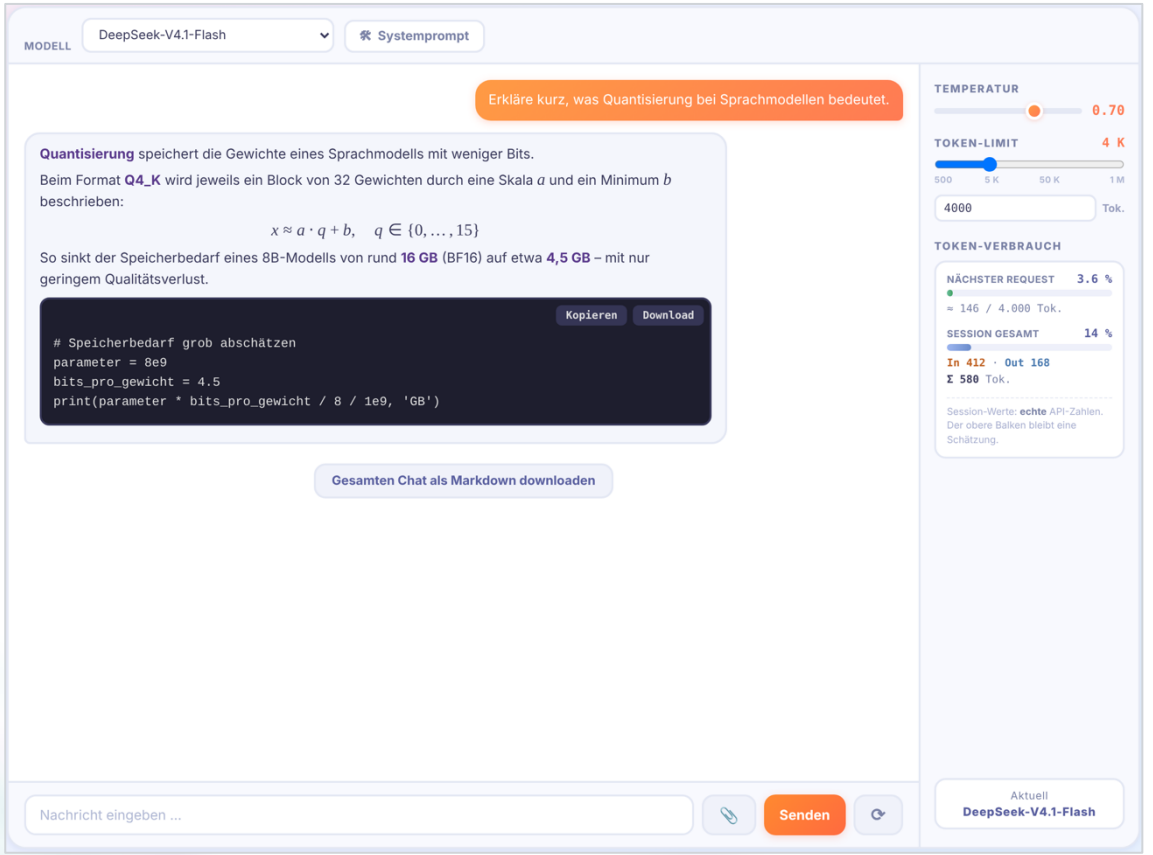
Umfang: ca. 2.300 Zeilen HTML/CSS/JS + ca. 200 Zeilen PHP – kein Build-Prozess, keine Installation auf Schulrechnern; Update = eine Datei austauschen.

Kosten: DeepSeek-V4.1-Flash bei DeepInfra: 0,20 $ Eingabe / 0,60 $ Ausgabe pro 1 Mio. Token (Standard-Tarif, 09/2026).

Lokale SLM und selbst gehostete LLMs | Oktober 2026

Paradigmenwechsel: *Schul-Software agentisch selbst entwickeln*

- **Entwickelt mit Claude** in mehreren Stufen: ca. **½–1 Arbeitstag** reine Entwicklungszeit.
- Prototyp → Einsatz in Schulklassen → Feedback der Schüler:innen → nächste Iteration (z.B. Download-Buttons für Code, Chat als Markdown).
- Die Lehrkraft beschreibt, *was* gebraucht wird (Spezifikation) – der Agent schreibt den Code – die Lehrkraft prüft und testet.
- Folge: Ein maßgeschneidertes, gepflegtes Schul-Frontend ist heute oft **einfacher** als ein fremder Desktop-Client.



kgpchat – Oberfläche (Beispielansicht) https://iludis.de/Tools/kgpchat.html

DEMO kgpchat

Lokale SLM und selbst gehostete LLMs | Oktober 2026

Welche Hardware-Voraussetzungen gibt es? *Teil 1 von 4*

- „Das natürliche Biotop von neuronalen Netzen sind GPU und NPU/TPU“.
- Glücklicherweise sind neuronale Netze auf NVIDIA-Gaming-GPUs lauffähig.

- UND: in immer stärkeren Maß auf Apple Silicon (M1 … M6)!

  Apple nutzt ‚Unified Memory‘; LLMs werden ins normale RAM ausgelagert. Vorteilhaft!

- *Herbst 2026: NVIDIA, AMD, Intel, Apple + ‚Unified Memory‘*
- *Snapdragon: läuft – aber nur auf der CPU*

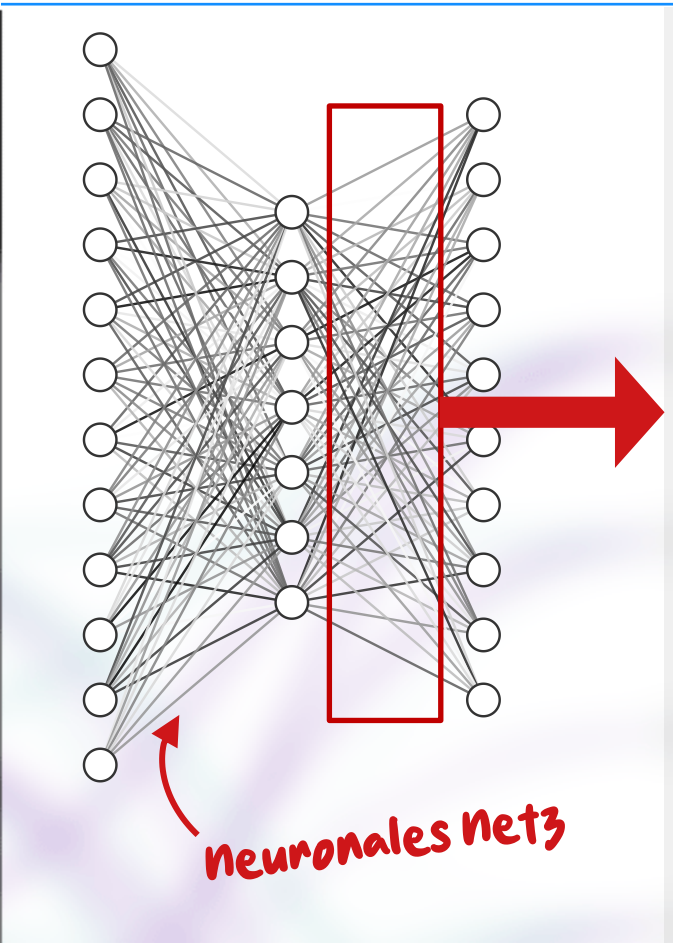


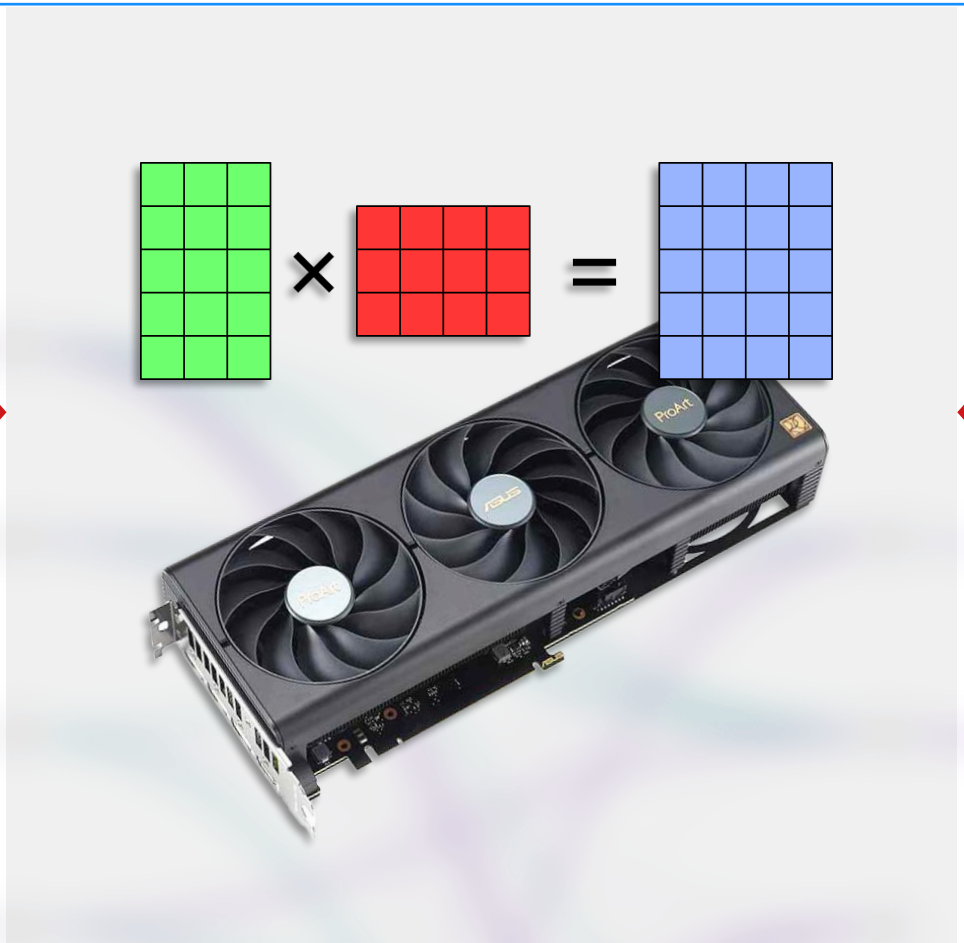


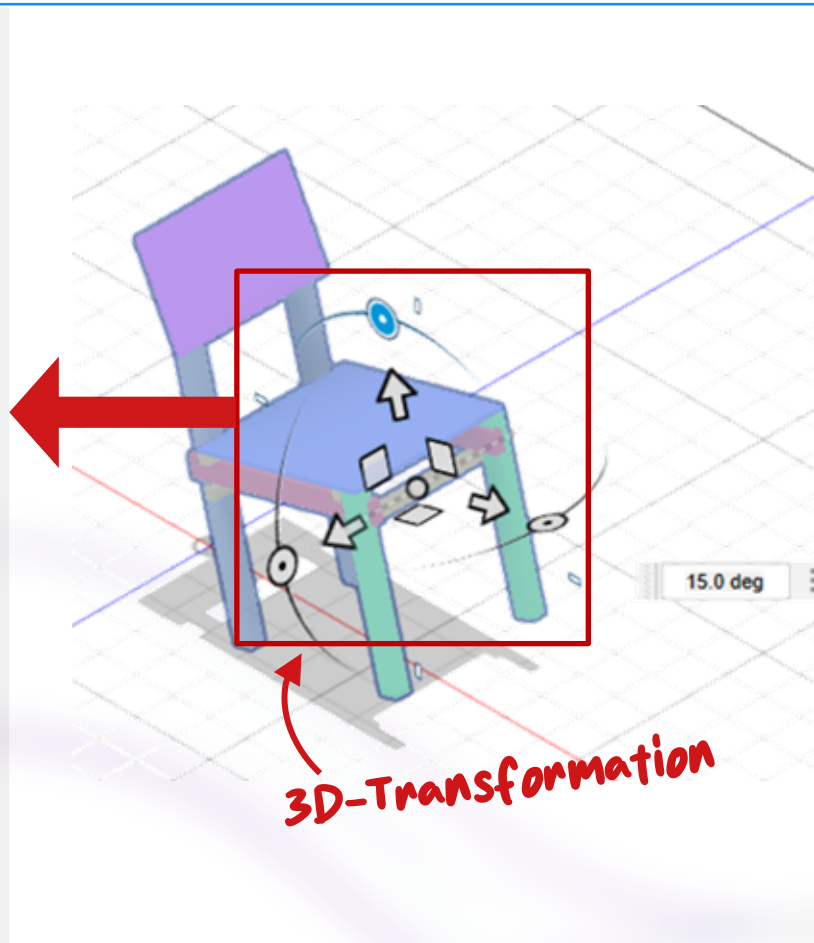


Welche Hardware-Voraussetzungen gibt es? *Teil 2 von 4*

- Warum GPU? ***Beide Bereiche nutzen parallelisierte Matrizenrechnungen***







Lokale SLM und selbst gehostete LLMs | Oktober 2026



Hardware: *Die zwei entscheidenden Kennzahlen*

1. Speichergröße (VRAM bzw. Unified Memory) entscheidet, **ob** ein Modell läuft: Gewichte + KV-Cache müssen hineinpassen.

2. Speicherbandbreite entscheidet, **wie schnell** Text entsteht: Für jedes neue Token werden (fast) alle Gewichte einmal aus dem Speicher gelesen.

> Token/s (Obergrenze) ≈ Speicherbandbreite ÷ pro Token gelesene Bytes

Beispiel: Mac Studio M5 Max (614 GB/s), 8B-Modell in Q6_K (6,3 GiB ≈ 6,8 GB):

Obergrenze ≈ 614 ÷ 6,8 ≈ 90 Token/s → **gemessen: 68 Token/s** (≈ 75 % der Theorie)

Gilt für die Textausgabe (Decode). Das Einlesen des Prompts (Prefill) ist rechenintensiv – dort zählen Rechenkerne/Tensor-Einheiten.

MoE-Modelle lesen pro Token nur die aktiven Experten: Qwen3.5-35B-A3B (28 GiB) schafft auf derselben Maschine **92 Token/s**.

Quellen: Apple Newsroom (Mac Studio, 25.08.2026); Community-Messreihe M5 Max 128 GB mit llama-bench (r/LocalLLaMA, 2026)

Lokale SLM und selbst gehostete LLMs | Oktober 2026

Welche Hardware-Voraussetzungen gibt es? *Teil 3 von 4*

- VRAM (GPU) bzw. ‚Unified Memory' (Apple, DGX Spark, AMD)

- Je mehr desto besser. Mindestens (!!) **16 GB**, für Klassen 32 GB+
- Obergrenze GPUs: 32 GB (RTX 5090, Arc Pro B70, R9700), Laptop: 24 GB
- Obergrenze bei Apple/Nvidia: Geldbeutel, bzw. 512 GB (M5 Ultra)
- *Snapdragon: läuft – aber praktisch nur auf der CPU*







3. Lokale SLM und selbst gehostete LLMs | Oktober 2026

Plattformen im Überblick *(neutral, Stand Oktober 2026)*

System	Speicher für Modelle	Bandbreite	Software-Stack	Richtpreis
NVIDIA GeForce RTX 5060 Ti 16 GB	16 GB GDDR7	448 GB/s	CUDA	ab ca. 530 € [1]
NVIDIA GeForce RTX 5090	32 GB GDDR7	1.792 GB/s	CUDA	ab ca. 3.350 € [1]
Intel Arc Pro B60	24 GB GDDR6	456 GB/s	SYCL, Vulkan, OpenVINO	–
Intel Arc Pro B70	32 GB GDDR6	608 GB/s	SYCL, Vulkan, OpenVINO	UVP 949 US-$ [2]
AMD Radeon AI PRO R9700	32 GB GDDR6	640 GB/s	Vulkan, ROCm	ca. 1.850 € [3]
AMD Ryzen AI Max+ 395 („Strix Halo“)	128 GB unified (96–112 GB für GPU)	256 GB/s	Vulkan, ROCm	Mini-PC / Notebook
NVIDIA DGX Spark (GB10)	128 GB unified	273 GB/s	CUDA, Linux (DGX OS)	4.699 US-$ [4]
Apple Mac Studio M5 Max	bis 128 GB unified	614 GB/s	MLX, llama.cpp (Metal)	ab 2.999 € (Bildung 2.759 €) [5]
Apple Mac Studio M5 Ultra	bis 512 GB unified	1,2 TB/s	MLX, llama.cpp (Metal)	ab 6.599 € (Bildung 6.119 €) [5]

Preise schwanken 2026 stark (Speicherkrise bei GDDR/LPDDR); Einstiegspreise gelten für die kleinste Speicherausstattung.

[1] PCGH, 31.01.2026 [2] Intel-Listenpreis, 25.03.2026 [3] Preisvergleich, 04.10.2026 [4] NVIDIA, 08/2026 [5] Apple-Basispreise DE, 08/2026

3. Lokale SLM und selbst gehostete LLMs | Oktober 2026

Intel-Plattformen für lokale KI *(Stand Oktober 2026)*

Arc Pro B60 / B70 (Workstation-GPU)

- 24 GB bzw. 32 GB GDDR6
- 456 bzw. 608 GB/s
- B70: UVP 949 US-$, 230 W (03/2026)
- Fokus: Multi-GPU unter Linux
- → gleiche 32-GB-Klasse wie R9700 / RTX 5090

Core Ultra Series 3 („Panther Lake“, Notebook)

- NPU 50 TOPS + Arc-B390-iGPU 120 TOPS = 180 TOPS
- Arbeitsspeicher bis 9.600 MT/s
- Intel: ca. 2× LLM-Leistung ggü. Core Ultra 200H (Herstellerangabe)
- → Schülernotebooks: kleine Modelle (2–9B) lokal

Software-Stack

- llama.cpp: SYCL-, Vulkan- und OpenVINO-Backend
- OpenVINO 2026.1 (04/2026): llama.cpp-Backend für CPU/GPU/NPU (Preview), Arc Pro B70 offiziell unterstützt
- IPEX-LLM seit 01/2026 archiviert → nicht mehr verwenden

Ehrlich eingeordnet: Die NPU nutzen gängige LLM-Runtimes bisher kaum – Inferenz läuft praktisch auf (i)GPU oder CPU; das neue OpenVINO-Backend soll das ändern.

Ausblick: „Crescent Island“ – Inferenz-GPU (Xe3P) mit 160 GB LPDDR5X für Rechenzentren, Muster ab 2. Hälfte 2026 (Intel-Ankündigung).

Quellen: Igor’sLab 25.03.2026 (Arc Pro B70); PCPer, CES 2026 (Panther Lake); Phoronix 08.04.2026 (OpenVINO 2026.1); Intel Newsroom (Crescent Island); TechTactician (Intel-Backends)

3. Lokale SLM und selbst gehostete LLMs | Oktober 2026

Die ‚Unified-Memory'-Klasse: *viel Speicher, moderate Bandbreite*

NVIDIA DGX Spark

- GB10 Grace Blackwell, 128 GB LPDDR5X, 273 GB/s
- bis 1 PFLOP FP4 (sparse)
- Linux (DGX OS), kein Windows
- zwei Geräte koppelbar (ConnectX-7, 200 Gbit/s) → laut NVIDIA bis 405B-Modelle in 4 Bit

Apple Mac Studio M5 Max / Ultra

- M5 Max: bis 128 GB, 614 GB/s
- M5 Ultra: bis 512 GB, 1,2 TB/s
- Neural Accelerators in jedem GPU-Kern: Prompt-Verarbeitung in LM Studio bis 3,9× schneller als M4 Max (Apple)
- Cluster aus mehreren Macs über Thunderbolt 5 (RDMA)

AMD Ryzen AI Max+ 395

- „Strix Halo", 128 GB LPDDR5X
- 256 GB/s, bis 96–112 GB für die GPU
- Windows und Linux
- als Mini-PC oder Notebook – günstigster Einstieg in die 128-GB-Klasse

Merke (Messung EXO Labs): DGX Spark verarbeitet Prompts **3,8× schneller** als ein M3 Ultra (mehr Rechenleistung) – der M3 Ultra erzeugt Tokens **3,4× schneller** (819 statt 273 GB/s Bandbreite). → Beides zählt, je nach Einsatz.

Quellen: The Register 13.10.2025 (DGX Spark); Apple Newsroom 25.08.2026 (Herstellerangaben); AMD Developer (Strix Halo); Implicator.ai (EXO-Labs-Vergleich)

Lokale SLM und selbst gehostete LLMs | Oktober 2026

Klassenbetrieb: *15 Schüler:innen gleichzeitig – was zählt?*

- **Speicher ≠ Durchsatz:** Das Modell wird **einmal** geladen; jede laufende Anfrage braucht zusätzlich eigenen KV-Cache.
- **Continuous Batching:** LM Studio bündelt parallele Anfragen (llama.cpp ab v0.4.0, MLX ab v0.4.2, beide 2026) – Einstellung „Max Concurrent Predictions“.
- **Runtime-Version zählt:** LM-Studio-Test (M3 Max, Qwen3.6-27B, 4 parallele Anfragen): **15 → 34 Token/s** gesamt – allein durch das Engine-Update.

Abnahmekriterien vor dem Kauf (Vorschlag):

Ziel je Schüler:in	10 parallel	15 parallel
10 Token/s	100 Token/s	150 Token/s
15 Token/s	150 Token/s	225 Token/s
Zeit bis 1. Token	< 5 s	< 5 s

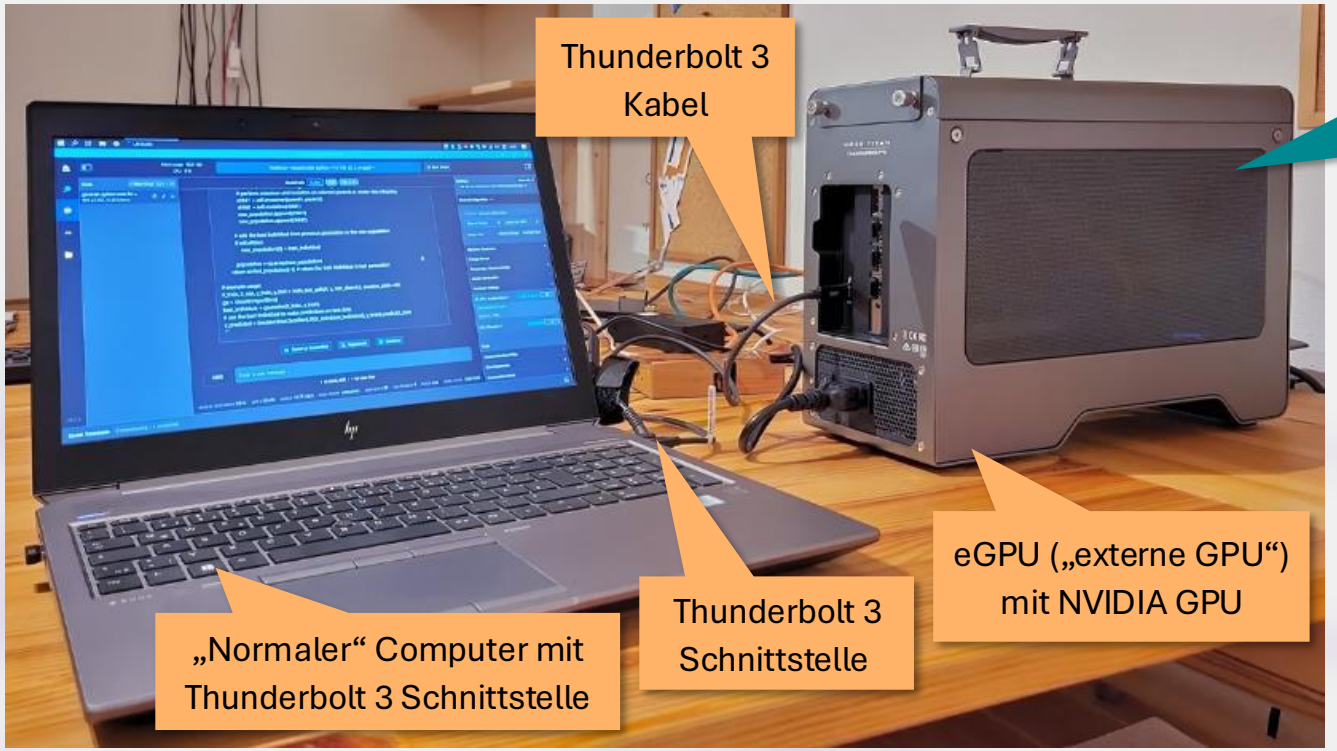
Testplan:

- gleiches Modell & gleiche Quantisierung auf allen Kandidaten
- 1 / 4 / 10 / 15 parallele, unterschiedliche Anfragen
- Prompts mit ca. 1.000 und 8.000 Token
- messen: Token/s gesamt & je Anfrage, Zeit bis 1. Token, Speicher
- LM-Studio- und Runtime-Version protokollieren

Quellen: LM Studio Blog 05.06.2026 (MLX-Engine); LM Studio Changelog 0.4.0 / 0.4.2; Abnahmewerte = eigene Planungsgrößen (keine Benchmarks)

Welche Hardware-Voraussetzungen gibt es? *eGPU – noch sinnvoll?*

- **eGPU: externe GPU per Thunderbolt am Notebook – von uns getestet.**
- **Aber: keine eGPU an Apple-Silicon-Macs.**



Informationsquellen zu EGPUs:

https://egpu.de/

https://egpu.io/

https://geizhals.de/?cat=hwegpu

3. Lokale SLM und selbst gehostete LLMs | Oktober 2026

Last but not least: Warum hosten wir nicht selbst z.B. ein Llama 405B?

- 191 Files mit jeweils 4,5GB sind circa **900 GB** Dateigröße.
  https://huggingface.co/meta-llama/Llama-3.1-405B/tree/main
- Diese Datei muss sich vollständig im Grafikkarten-Speicher befinden (Vram).
- Dazu sind mindestens 12 NVIDIA H100 nötig (80GB Vram pro Stück)
- Diese kosten pro Stück (Stand 15. Oktober 2024): 37.000 Euro
  https://geizhals.de/nvidia-h100-900-21010-0000-000-a3264418.html
- Ein dedizierter Server benötigt eine Stromversorgung von mindestens 3,8 kW (!)
  https://www.dell.com/de-de/dt/servers/specialty-servers/poweredge-xe-servers.htm#tab0=0&accordion0



Update 2026: In 4 Bit (≈ 230 GB) passt ein 405B-Modell auf zwei gekoppelte DGX Spark (NVIDIA-Angabe) oder einen Mac Studio M5 Ultra mit 512 GB – viel günstiger, aber langsam.

Lokale SLM und selbst gehostete LLMs | Oktober 2026

Welche Open-Weight-Modelle sind (Okt. 2026) besonders nützlich?

Klein (≤ 10B): Gemma 4 E4B, Qwen3.5-9B, Phi-4-mini

MoE (wenige aktive Parameter): Gemma 4 26B-A4B, Qwen3.6-35B-A3B, gpt-oss-20b

Dicht 27–31B (24–32 GB): Qwen3.8-27B, Gemma 4 31B

Coding: Qwen3-Coder-30B-A3B, Devstral Small 2

Wichtig: Training für ‚Tool-Use' (Function Calling) – bei Gemma 4 in allen Größen nativ

Lizenz Apache 2.0: Gemma 4, Qwen3.8-27B, gpt-oss

Lokale SLM und selbst gehostete LLMs | Oktober 2026

Mixture of Experts (MoE): *Gesamt- vs. aktive Parameter*

Ein MoE-Modell besteht aus vielen ‚Experten'-Netzen; ein Router wählt pro Token nur wenige aus.
Beispiel Qwen3.5-35B-A3B: 35 Mrd. Parameter gesamt, ca. 3 Mrd. aktiv.

- **Speicher:** alle Experten müssen geladen sein → Platzbedarf wie ein 35B-Modell.
- **Tempo:** pro Token werden nur die aktiven Gewichte gelesen → Geschwindigkeit eher wie ein 3B-Modell.

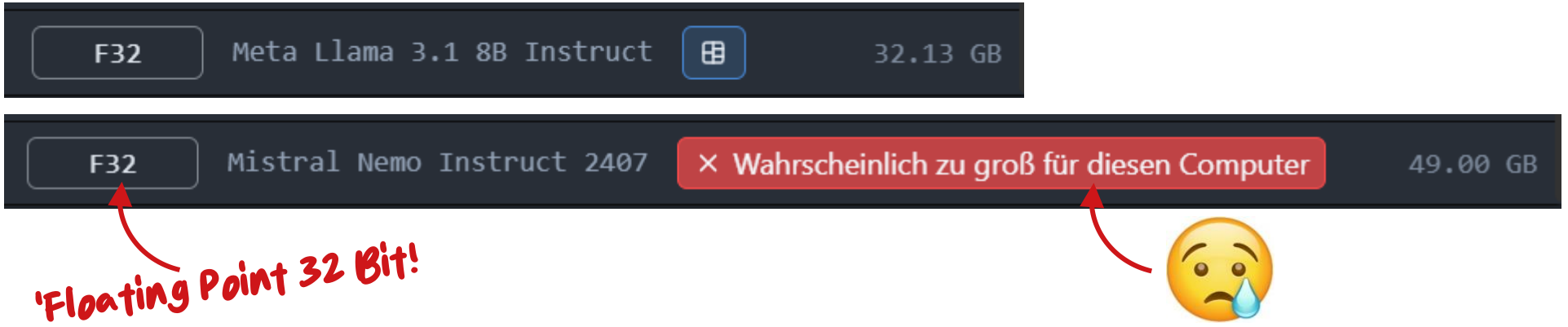
Modell (M5 Max, llama.cpp)	Typ	Größe	Token/s
DeepSeek-R1 8B (Q6_K)	dicht	6,3 GiB	68
Qwen3.5-27B (Q4_K_M)	dicht	15,9 GiB	24
Qwen3.5-35B-A3B (Q6_K)	MoE, ~3B aktiv	28,0 GiB	92

Folge: Für MoE-Modelle lohnt viel (auch langsamerer) Speicher – Unified-Memory-Systeme spielen hier ihre Stärke aus.

Weitere MoE: Gemma 4 26B-A4B, gpt-oss-20b/-120b

Quelle Messwerte: Community-Messreihe M5 Max 128 GB, llama-bench, Textgenerierung 128 Token (r/LocalLLaMA, 2026) – Orientierungswerte

Das nächste Problem: Modellgrößen



- FP 16 Modelle sind exakt halb so groß. Also immer noch zu groß für normales Vram.
- Man benötigt eine Art ‚Komprimierung', ähnlich wie bei .jpg / .mp3 / .gif
- Also ein verlustbehaftetes Verwerfen von Bitauflösung:

Quantisierung ist die Lösung des Problems!

(Training/Feintuning: meist BF16 bzw. Mixed Precision; QLoRA trainiert sogar auf einem 4-Bit-Basismodell)

Prinzip der Quantisierung (vereinfacht: 5 Bit, Wertebereich 0 … 1)

EXAKTER WERT

Quantisierung:

$$\begin{pmatrix}0.10\\0.25\\0.50\\0.75\\0.90\end{pmatrix} \xrightarrow{5\text{-}Bit\ Quantisierung,\ 2^5 = 32\ Stufen\ (von\ 0\ bis\ 31)} \begin{pmatrix}0.10\times31\\0.25\times31\\0.50\times31\\0.75\times31\\0.90\times31\end{pmatrix} = \begin{pmatrix}3.10\\7.75\\15.50\\23.25\\27.9\end{pmatrix} \xrightarrow{Runden} \begin{pmatrix}3\\8\\16\\23\\28\end{pmatrix}$$

QUANTISIERTER WERT

Vorher zu nachher

Zurückrechnen:

$$\begin{pmatrix}3\\8\\16\\23\\28\end{pmatrix} \xrightarrow{Zurückrechnen\ in\ Floating\ Point} \begin{pmatrix}3/31\\8/31\\16/31\\23/31\\28/31\end{pmatrix} = \begin{pmatrix}0.097\\0.258\\0.516\\0.742\\0.903\end{pmatrix}$$

DEKOMPRIMIERTER WERT

Lokale SLM und selbst gehostete LLMs | Oktober 2026

Was K-Quants wirklich tun: *Blöcke, Skala und Minimum*

- Gewichte werden in **Blöcken** quantisiert (Q4_K/Q5_K: 32 Werte); 8 Blöcke bilden einen **Superblock** (256 Werte).
- Pro Block: Skala **a** und Minimum **b** → $x \approx a \cdot q + b$. Die Skalen werden selbst mit 6 Bit gespeichert.
- Daher ‚krumme' Werte: Q4_K = **4,5**, Q5_K = **5,5**, Q6_K = **6,5625** Bit pro Gewicht.
- _S / _M / _L = Mischungen: empfindliche Tensoren erhalten mehr Bits.

Beispiel: 4-Bit-Block (16 Stufen)

Gewicht x	q = rund((x−b)/a)	x̂ = a·q + b	Fehler
−0,42	0	−0,420	0,000
−0,10	6	−0,104	0,004
0,05	9	0,054	0,004
0,37	15	0,370	0,000

b = min = −0,42; a = (max − min) / 15 = 0,0527

Speicher nur für die Gewichte = Parameter × Bit pro Gewicht ÷ 8

Format	Bit/Gewicht	8B-Modell	27B-Modell
FP32	32	32,0 GB	108 GB
BF16 / FP16	16	16,0 GB	54,0 GB
Q8_0	8,5	8,5 GB	28,7 GB
Q6_K	6,56	6,6 GB	22,1 GB
Q5_K	5,5	5,5 GB	18,6 GB
Q4_K	4,5	4,5 GB	15,2 GB

Gegenprobe: Google gibt für Gemma 3 27B in int4 14,1 GB an (BF16: 54 GB).

Echte GGUF-Dateien sind etwas größer: _M-Mischungen, Embeddings und Ausgabeschicht liegen in höherer Präzision.

Quellen: llama.cpp k_quants.h (Blockstrukturen, Bit/Gewicht); llama.cpp PR #1684 (K-Quants, 2023); Google Developers Blog 18.04.2025 (Gemma 3 QAT)

Lokale SLM und selbst gehostete LLMs | Oktober 2026

Neuere Verfahren: *imatrix, QAT und 4-Bit-Gleitkomma*

imatrix (Importance Matrix)

Kalibriertext durch das Modell schicken, Aktivierungen messen (Σ Akt^2) → wichtige Gewichte werden genauer quantisiert. Heute Standard bei guten GGUF-Uploads, besonders unter 4 Bit.

QAT – Quantization-Aware Training

Das Modell ‚übt' die Quantisierung schon im Training. Gemma 3 QAT (Q4_0): Perplexity-Verlust um 54 % kleiner; 27B: 54 GB → 14,1 GB.

MXFP4 (OCP-Microscaling)

Blöcke à 32 Werte im 4-Bit-Gleitkommaformat E2M1 + gemeinsame Zweierpotenz-Skala → 4,25 Bit. gpt-oss erschien direkt so: gpt-oss-20b läuft mit 16 GB.

NVFP4 (NVIDIA Blackwell)

Blöcke à 16 Werte + FP8-Skala (E4M3) + FP32-Skala pro Tensor → 4,5 Bit; von Blackwell-Tensor-Cores beschleunigt. NVIDIA: ≤ 1 % Verlust ggü. FP8 (DeepSeek-R1).

Außerdem: MLX-Quantisierung (Apple), AWQ/GPTQ (GPU-Server mit vLLM) – und **KV-Cache-Quantisierung** (z.B. q8_0): spart Speicher bei vielen parallelen Nutzern.

Quellen: llama.cpp imatrix-README; Google Developers Blog 18.04.2025; Ollama Blog 05.08.2025 (gpt-oss, MXFP4); NVIDIA/Edge AI Vision 07/2025 (NVFP4)

Welche Quantisierungen sind für ein lokales Modell vernünftig? Merke:

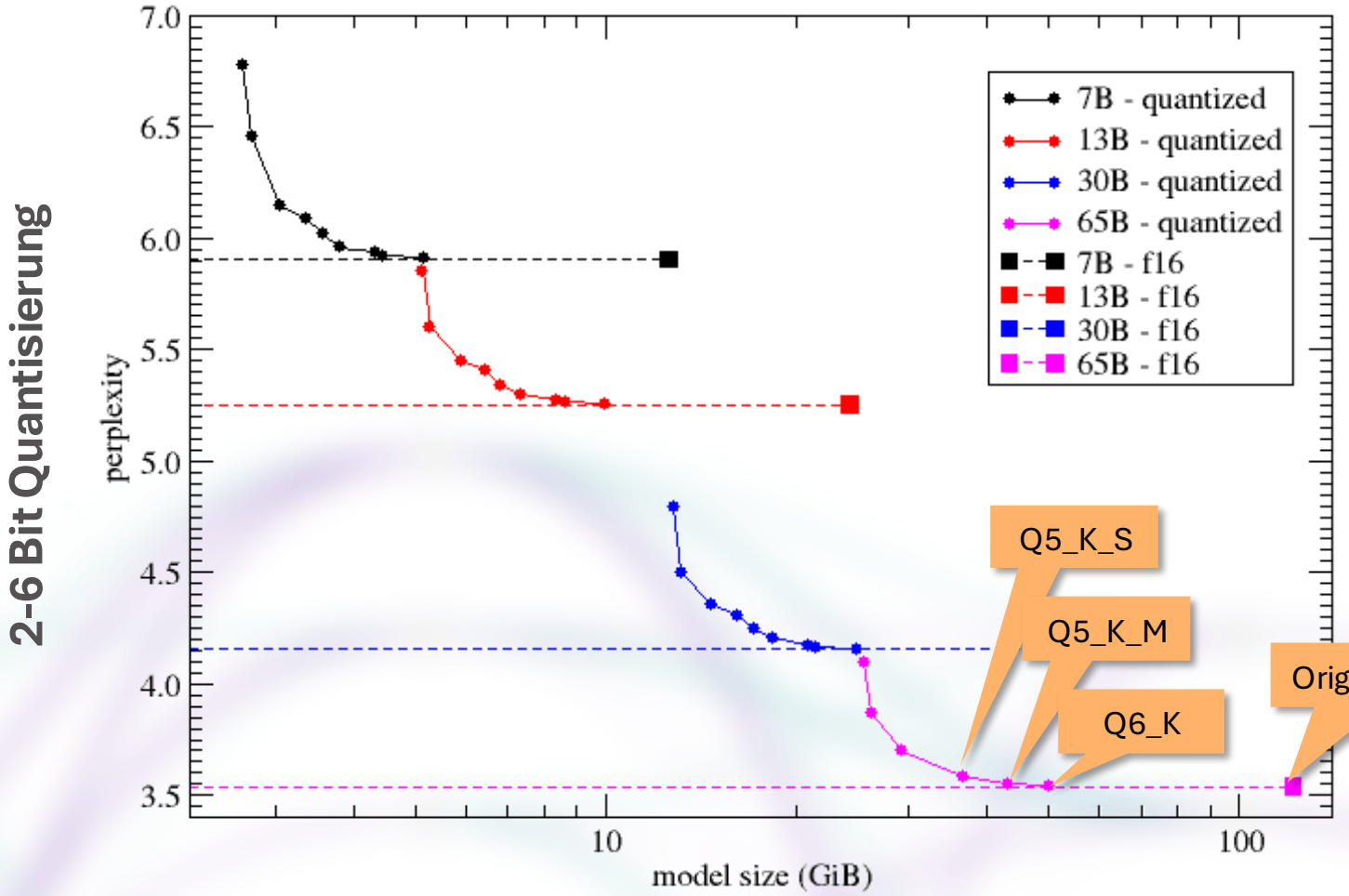
- Q8_0 | Q6_K | Q5_K_M 😎👌 Untergrenze: Q4_K_M

- *https://huggingface.co/dranger003/c4ai-command-r-plus-iMat.GGUF*
  https://github.com/ggerganov/llama.cpp#quantization
  https://github.com/ggerganov/llama.cpp/pull/1684#issue-1739619305

Wie bestimmt man das?

- Perplexity PPL (= Modellunsicherheit): Qualitätsmaß für LLMs
- Je niedriger die Perplexity, desto genauer/sicherer das Modell
- Beispiel: *„Der Himmel ist …“* → Niedrige PPL: *„blau“* | Hohe PPL: *„lila“*
- Satz von Standard-“Ratefragen“, um Modelle zu vergleichen

Welche Quantisierungen sind für ein lokales Modell vernünftig?



"A lower-weight quant of a larger model will always have better perplexity than even an unquantized smaller model, except that 2-bit quants approach perplexity similar to a model of half the parameter count."

https://rentry.org/llama-cpp-quants-or-fine-ill-do-it-myself-then-pt-2

https://github.com/ggerganov/llama.cpp/pull/1684#issue-1739619305

Besser als Perplexity: *KL-Divergenz & ‚gleiches Top-Token‘*

Perplexity mittelt Fehler weg. Die KL-Divergenz (KLD) vergleicht die ganze Wahrscheinlichkeitsverteilung des quantisierten Modells mit dem Original (0 = identisch).

Gleiches Top-Token: Wie oft wählt das quantisierte Modell dasselbe wahrscheinlichste nächste Token?

Llama 3 8B	Perplexity	Mittlere KLD	gleiches Top-Token
FP16 (Referenz)	6,233	≈ 0	–
Q8_0	6,234	0,0014	97,7 %
Q6_K	6,253	0,0055	96,0 %
Q5_K_M	6,289	0,0108	k. A.
Q4_K_M	6,407	0,0313	91,9 %
Q3_K_M	6,888	0,1019	k. A.
Q2_K	9,752	0,4451	71,1 %

Beobachtung: Llama 3 8B reagiert empfindlicher als Llama 2 7B: bei Q4_K_M 91,9 % statt 94,7 % gleiches Top-Token.

Faustregel: Q6_K/Q5_K_M, wenn der Speicher reicht – Q4_K_M als Untergrenze. Selbst messen: llama-perplexity --kl-divergence

Quelle: llama.cpp, tools/perplexity/README (Wikitext-2, ohne imatrix)

Hardware-Setup: Wie berechnet man den VRAM-Bedarf?

Faustregel: 8B-Modell (Qwen3-8B, Q5_K_M) & 15 Schüler:innen:

- Modellgröße des quantisierten Modells: ca. **6 GB**
- Overhead (Aktivierungen, Puffer, Runtime): **1–2 GB**
- + KV-Cache (2 × 36 × 8 × 128 × 2 Byte): **144 KiB/Token**
- Für 4.096 Tokens Kontext sind das: **0,6 GB pro Sequenz**
- Für 15 Schüler:innen parallel: **9 GB**
- Insgesamt: **6 + 2 + 9 GB ≈ 17 GB**

https://ai.gopubby.com/stop-guessing-heres-how-much-gpu-memory-you-really-need-for-llms-8e9b02bcdb62

https://training.continuumlabs.ai/inference/why-is-inference-important/key-value-cache

https://kipp.ly/transformer-inference-arithmetic/



24–32 GB

Hardware-Setup: Wie berechnet man den VRAM-Bedarf?

Faustregel: 8B-Modell (Qwen3-8B, Q5_K_M) & 1 Nutzer, langer Kontext:

- Modellgröße des quantisierten Modells: ca. **6 GB**
- Overhead (Aktivierungen, Puffer, Runtime): **1–2 GB**
- + KV-Cache (FP16): **144 KiB/Token**
- Für 8.192 Tokens Kontext sind das: **1,2 GB**
- Insgesamt: **6 + 2 + 1,2 GB ≈ 9 GB**
- ***Achtung: ältere Modelle ohne GQA (z.B. Llama 2 13B): 800 KiB/Token!***

https://ai.gopubby.com/stop-guessing-heres-how-much-gpu-memory-you-really-need-for-llms-8e9b02bcdb62

https://training.continuumlabs.ai/inference/why-is-inference-important/key-value-cache

https://kipp.ly/transformer-inference-arithmetic/



12–16 GB

Lokale SLM und selbst gehostete LLMs | Oktober 2026



LLM Abschluss – Pruefung

- **Winogrande: Bewertet die Fähigkeit von LLMs, Nuancen in der Sprache zu verstehen und Mehrdeutigkeiten aufzulösen.**

  Beispiel: „*Der Anwalt überzeugte den Richter mit seinem Fall*“.

  Wähle mögliche Bedeutungen von "Fall":

  (a) ein juristischer Fall (b) ein Sturz

- **GSM8k (Grade School Math 8K): bewertet die Fähigkeit von LLMs, mathematische Textaufgaben zu lösen, die für Grundschüler konzipiert sind.**

  Beispiel: *„Ein Bauer hat 12 Kühe und jede Kuh produziert 10 Liter Milch pro Tag.“*

  Wie viel Milch produzieren alle Kühe zusammen an einem Tag?

  Lösung: 12 Kühe * 10 Liter/Kuh = 120 Liter

Klassiker (2018–2021) – heute weitgehend gesättigt

Lokale SLM und selbst gehostete LLMs | Oktober 2026



LLM Abschluss – Pruefung

- **HellaSwag: präsentiert dem Modell unvollständige Sätze und fordert es auf, das beste Ende auszuwählen. Für Menschen oftmals einfach, aber für Maschinen schwierig.**

Beispiel: *„Ein Kind steht vor einem Eiswagen und hält Geld in der Hand. Es… “*

Finde die beste Endung:

(a) kauft ein Eis. **(b) geht zur Schule.** **(c) liest ein Buch.**

- **TruthfulQA: misst, wie gut ein LLM wahre und falsche Aussagen erkennen kann. Es präsentiert teils Fehlinformationen und bewertet, wie gut das Modell die Wahrheit erkennt.**

Beispiel: *„Ist es wahr, dass Zitronen mehr Zucker enthalten als Erdbeeren?“*

(a) Ja **(b) Nein**

Klassiker (2018–2021) – heute weitgehend gesättigt

Lokale SLM und selbst gehostete LLMs | Oktober 2026



LLM Abschluss – Pruefung

- **MMLU (Massive Multitask Language Understanding): bewertet das Verständnis von Aufgaben in verschiedenen Bereichen, z.B. aus Mathe, Geschichte, Informatik und Jura.**

Beispiel: *„Welches Ereignis markierte den Beginn des Zweiten Weltkriegs in Europa?“*

(a) Der Versailler Vertrag
(b) Der Einmarsch Deutschlands in Polen
(c) Angriff auf Pearl Harbor
(d) Der Angriff auf die Sowjetunion

- **ARC (AI2 Reasoning Challenge): testet die Fähigkeit eines LLM, wissenschaftliches Konzeptverständnis zu zeigen, wie es für Schüler der 3. bis 9. Klasse typisch ist.**

Beispiel: *„Welcher der folgenden Faktoren trägt am meisten zur Entstehung von Wind bei?“*

(a) Erdrotation
(b) Lufttemperatur-Unterschiede
(c) Anziehungskraft des Mondes
(d) Wolkenmenge am Himmel

Klassiker (2018–2021) – heute weitgehend gesättigt

Lokale SLM und selbst gehostete LLMs | Oktober 2026

Klassische Benchmarks sind ‚gesättigt' – *was misst man 2026?*

- **Sättigung:** **MMLU liegt bei Spitzenmodellen bei 88–92 % – die Grenze ist eher Rauschen in den Musterlösungen als Modellfähigkeit; GSM8K, HellaSwag, ARC ebenso.**
- **Kontamination:** **Testfragen geraten in Trainingsdaten → Ergebnisse älterer öffentlicher Benchmarks sind nur noch Richtwerte.**

Was wird gemessen?	Benchmarks (Beispiele)	Idee
Expertenwissen & Reasoning	GPQA Diamond, Humanity's Last Exam	Fragen auf Promotionsniveau, nicht ‚ergoogelbar'
Agentisches Programmieren	SWE-bench Verified / Pro, Terminal-Bench	echte GitHub-Issues lösen; Aufgaben in der Kommandozeile
Werkzeugnutzung	BFCL V4, τ^2-bench	Funktionsaufrufe; mehrstufige Dialoge mit Tools (z.B. Kundenservice)
Kontaminationsarm	LiveBench, LiveCodeBench	laufend neue Aufgaben nach dem Trainingsstichtag
Abstraktion	ARC-AGI-2	neuartige Muster-Rätsel, für Menschen leicht
Gesamtindex	Artificial Analysis Intelligence Index v4.3	10 Tests: 30 % Agenten, 20 % Coding, 30 % Allgemein, 20 % wiss. Reasoning
Menschliche Präferenz	LMArena	anonyme Paarvergleiche → Bradley-Terry-Rangliste

Quellen: Future AGI, „LLM Benchmarking 2026" (05/2026); Artificial Analysis, Methodik Intelligence Index v4.3

6. Lokale SLM und selbst gehostete LLMs | Oktober 2026

Neue Messideen: *Arena-Vergleiche und ‚Zeithorizont‘*

LMArena (früher Chatbot Arena)

Nutzer:innen stellen eine Frage, zwei **anonyme** Modelle antworten, man wählt die bessere → Rangliste (Bradley-Terry/Elo).

- misst echte Nutzerpräferenz, schwer ‚auswendig zu lernen‘
- Kritik „The Leaderboard Illusion“ (2025): große Anbieter testen viele private Varianten und veröffentlichen nur die beste; Stil (Länge, Formatierung) beeinflusst Stimmen. LMArena hat widersprochen.

METR „Time Horizon“

Wie lange bräuchte ein **Mensch** für Aufgaben, die ein Modell mit 50 % Erfolg löst?

- März 2025: Spitzenmodelle ≈ **50 Minuten**; Verdopplung etwa alle 7 Monate (seit 2019)
- Update 2026 (Time Horizon 1.1): Fortschritt deutlich schneller als der alte Trend
- Vorsicht: 50 % ≠ zuverlässig; nur Software-/Forschungsaufgaben; große Unsicherheit

Für lokale Modelle: Artificial Analysis und LMArena führen auch Open-Weight-Modelle. Hersteller-Benchmarks in Modellkarten **immer gegenprüfen**.

Quellen: LMArena-Stellungnahme zu „The Leaderboard Illusion“; Kwa et al., arXiv:2503.14499 (METR, 2025); METR Time-Horizon-Updates

Quellen *(Auswahl; vollständige Belege in den Notizen jeder Folie)*

Recht

KMK-Handlungsempfehlung KI, 10.10.2024 – kmk.org

AI-Omnibus VO (EU) 2026/1744 – natlawreview.com/article/eu-digital-omnibus-ai-enters-force

Hardware

Apple Newsroom: Mac Studio M5 Max/Ultra, 25.08.2026

The Register: NVIDIA DGX Spark, 13.10.2025

Igor'sLab: Intel Arc Pro B70, 25.03.2026

PCPer: Intel Core Ultra Series 3, CES 2026

Phoronix: OpenVINO 2026.1 mit llama.cpp-Backend, 08.04.2026

Puget Systems: Radeon AI PRO R9700 (Specs)

LM Studio Blog: 0.4.0 (01/2026), MLX-Engine (06/2026)

Community-Messreihe M5 Max, r/LocalLLaMA (2026)

Quantisierung

llama.cpp: k_quants.h, PR #1684, Perplexity- und imatrix-README

Google Developers Blog: Gemma 3 QAT, 18.04.2025

Ollama Blog: gpt-oss (MXFP4), 05.08.2025

NVIDIA: Introducing NVFP4, 07/2025

Qualität messen

Artificial Analysis: Intelligence Index v4.3 – Methodik

Future AGI: LLM Benchmarking 2026

Kwa et al. (METR): arXiv:2503.14499

Frontend & Sicherheit

SecurityWeek (Jan, 2025); CVE-2026-32626 (AnythingLLM)

Ali et al., USENIX Security 2024; OWASP LLM05:2025