

Vergleichbarkeit von Tokenkosten unterschiedlicher LLM-Konfigurationen in RAG-Systemen

Tom Keller

Institut für Allgemeinen Maschinenbau, Fakultät für Informatik und Ingenieurwissenschaften, TH Köln, Deutschland

Zusammenfassung. Tokenpreise pro Million Input- und Output-Tokens bilden eine transparente Grundlage für den Vergleich von Large Language Models (LLMs). Der KI-ssist LLM Provider Cost Check nutzt diese Grundlage, um Anbieter und Konfigurationen von Retrieval-Augmented-Generation-Systemen (RAG) unter identischen Annahmen gegenüberzustellen. Bei Reasoning-Modellen kommt eine weitere Kostenkomponente hinzu: intern erzeugte Reasoning-Tokens, die als Output abgerechnet werden, deren Umfang aber je nach Modell, Aufgabe und sogar einzelner Ausführung erheblich schwankt. Dieser Beitrag ordnet den Rechner auf Basis aktueller Forschung und Anbieterdokumentation ein, grenzt seinen Nutzen von seinen Grenzen ab und zeigt, für welche RAG-Aufgaben Reasoning einen Mehrwert erwarten lässt und für welche nicht.

1. Ein standardisierter Ausgangspunkt für den LLM-Kostenvergleich

Kommerzielle LLM-Anbieter geben ihre Preise in der Regel getrennt für Input- und Output-Tokens an, jeweils bezogen auf eine Million Tokens, wobei Output-Tokens deutlich teurer sind als Input-Tokens. Diese Preisstruktur hat einen methodischen Vorteil: Werden für alle Modelle dieselben Tokenmengen angenommen, lassen sich die resultierenden Kosten direkt und nachvollziehbar vergleichen.

Auf diesem Prinzip basiert der KI-ssist LLM Provider Cost Check. Der Input einer RAG-Anfrage setzt sich im Wesentlichen aus System Prompt, Nutzeranfrage (Query) und den abgerufenen Textabschnitten (Chunks) zusammen, der Output aus der generierten Antwort. Da der Rechner diese Größen einzeln konfigurierbar macht, unterstützt er zwei Arten von Analysen. Erstens lassen sich Modelle und Anbieter unter identischen Tokenannahmen vergleichen. Zweitens kann der Einfluss einzelner RAG-Parameter isoliert untersucht werden. Über das Anfragevolumen werden diese Effekte schließlich auf den laufenden Betrieb hochgerechnet.

2. Reasoning als zusätzliche und variable Kostenkomponente

Reasoning-Modelle erzeugen vor der eigentlichen Antwort zusätzliche Tokens, in denen sie eine Aufgabe zerlegen sowie Lösungswege prüfen und verwerfen. Diese Reasoning- oder Thinking-Tokens (im Folgenden einheitlich Reasoning-Tokens) sind je nach Modell vollständig, nur als Zusammenfassung oder gar nicht sichtbar. Abgerechnet werden sie dennoch vollständig als Output-Tokens, und sie belegen Platz im Kontextfenster (Anthropic, o. D.-a, o. D.-b; OpenAI, o. D.). Wie viele Reasoning-Tokens anfallen, unterscheidet sich zwischen Modellen, schwankt bei wiederholter Ausführung und steht in keinem festen Verhältnis zur sichtbaren Antwort.

2.1 Unterschiede zwischen Modellen bei identischer Aufgabe

L. Chen et al. (2026) untersuchten systematisch, wie gut Listenpreise die tatsächlichen Kosten von acht aktuellen Reasoning-Modellen auf zwölf Aufgabenbereichen abbilden. In 32 % der paarweisen Modellvergleiche verursachte das Modell mit dem niedrigeren Listenpreis höhere Gesamtkosten. Als dominierenden Faktor bei Aufgaben mit einer einzelnen Anfrage identifizierten die Autoren die stark unterschiedliche Menge an Reasoning-Tokens: Für dieselbe Anfrage kann ein Modell ein Vielfaches der Reasoning-Tokens eines anderen Modells benötigen. Im Extremfall erzeugte ein Modell für eine einzelne Aufgabe aus dem Benchmark MMLU-Pro über 60.000 Reasoning-Tokens, während ein anderes Modell mit 25 auskam. Ein niedrigerer Tokenpreis kann durch einen höheren Tokenverbrauch somit vollständig aufgehoben werden.

2.2 Streuung innerhalb desselben Modells

Für eine Kostenprognose noch bedeutsamer ist, dass der Aufwand auch bei identischem Modell und identischer Anfrage nicht konstant ist. Bei wiederholter Ausführung derselben Anfrage unterschieden sich Reasoning-Aufwand und Kosten zwischen dem günstigsten und dem teuersten Durchlauf um bis zu Faktor 9,7 (L. Chen et al., 2026). Die Autoren führen dies auf die stochastische Natur der Inferenz zurück, etwa auf Sampling und divergierende Lösungspfade, und folgern, dass die Kosten einer einzelnen Anfrage als Zufallsgröße mit

eigener Verteilung zu betrachten sind. Die Zahl der Reasoning-Tokens lässt sich daher weder aus der Promptlänge noch aus der Länge der sichtbaren Antwort deterministisch ableiten. Einschränkend ist anzumerken, dass die Modelle in dieser Studie mit aktiviertem Reasoning betrieben wurden, die OpenAI-Modelle mit der Effort-Stufe high. Die berichteten Spannweiten beschreiben somit eher reasoning-intensive Einstellungen; bei einfachen Fragen fielen Reasoning-Aufwand und Kostenunterschiede deutlich geringer aus.

2.3 Verhältnis von Reasoning-Tokens zur sichtbaren Antwort

Kumar et al. (2026) weisen für neun Reasoning-Modelle in kontextbasierten Frage-Antwort-Szenarien Reasoning- und sichtbare Output-Tokens getrennt aus. Bereits ohne jede Manipulation erzeugte OpenAI o1 auf dem Datensatz FreshQA im Mittel rund 570 Reasoning-Tokens bei rund 70 sichtbaren Output-Tokens, also etwa das Achtfache. Bei mehrstufigen Schlussfolgerungsaufgaben aus dem Datensatz MuSR waren es bis zu rund 1.300 Reasoning-Tokens bei rund 50 Output-Tokens. Andere Modelle zeigten deutlich geringere Verhältnisse, o4-mini auf FreshQA beispielsweise rund 100 zu 80. Reasoning-Tokens können die sichtbare Antwort somit um ein Vielfaches übersteigen, das Verhältnis ist aber stark modell- und aufgabenabhängig. Der eigentliche Untersuchungsgegenstand der Studie verdeutlicht zudem eine RAG-spezifische Eigenschaft: Werden in den abgerufenen Kontext gezielt Scheinaufgaben eingebettet, steigt der Reasoning-Aufwand um bis zu Faktor 46, während die Antwort korrekt bleibt. Nicht nur die Länge, sondern auch der Inhalt des Kontexts beeinflusst also den Reasoning-Aufwand.

Da Reasoning-Tokens zum Output-Tarif abgerechnet werden, können sie einen erheblichen Anteil der variablen LLM-Inferenzkosten einer Anfrage ausmachen oder diese sogar dominieren. Bei L. Chen et al. (2026) entfielen in den Fällen, in denen das günstiger gelistete Modell tatsächlich teurer war, bei Einzelanfragen über 95 % der Kostendifferenz auf Reasoning-Tokens. Diese Aussage bezieht sich ausschließlich auf die Inferenzkosten der Modellaufrufe. Die Gesamtkosten eines RAG-Systems umfassen darüber hinaus Embeddings, Vektordatenbank, Retrieval, Hosting und Betrieb.

3. Reasoning in RAG-Systemen: notwendig oder optional?

Aus der Kostenrelevanz von Reasoning folgt nicht, dass ein RAG-System grundsätzlich ein Reasoning-Modell benötigt. Betrachtet wird hier ein einfaches RAG-System, in dem zu einer Anfrage einmalig Chunks abgerufen und gemeinsam mit System Prompt und Query in einem einzigen Modellaufruf verarbeitet werden. Während der Generierung werden keine weiteren Informationen abgerufen. Ob Reasoning sinnvoll ist, hängt davon ab, was das Modell mit dem übergebenen Kontext tun muss.

3.1 Aufgaben ohne Bedarf an aufwendigem Reasoning

Enthalten die abgerufenen Chunks die Antwort bereits weitgehend direkt, muss das Modell die relevante Information im übergebenen Kontext nur noch identifizieren, gegebenenfalls extrahieren und sprachlich aufbereiten. Typische Beispiele sind die Wiedergabe einer Angabe aus der Dokumentation, die Extraktion eines Werts, die Zusammenfassung eines eindeutigen Abschnitts oder eine Frage, die sich auf eine einzelne Textstelle bezieht. Sprague et al. (2025) zeigen in einer Metaanalyse von über 100 Studien sowie eigenen Experimenten mit 14 Modellen auf 20 Datensätzen, dass promptbasiertes Chain-of-Thought vor allem bei mathematischen und logischen Aufgaben deutliche Verbesserungen bringt, bei anderen Aufgabentypen dagegen nur geringe. Im Wissensbenchmark MMLU erreichte die direkte Antwort nahezu dieselbe Genauigkeit wie Chain-of-Thought, sofern Frage oder Antwort keine Rechenoperation enthielten. X. Chen et al. (2025) zeigen an mathematischen Aufgaben unterschiedlicher Schwierigkeit, dass Reasoning-Modelle gerade bei einfachen Aufgaben zu Overtinking neigen und redundante Lösungsdurchläufe erzeugen, die kaum zur Genauigkeit beitragen. W.-L. Chen et al. (2026) belegen ergänzend, dass die reine Länge eines Reasoning-Prozesses kein verlässlicher Indikator für dessen Qualität ist. Keine dieser Studien untersucht RAG direkt. Ihre Ergebnisse legen jedoch nahe, dass Reasoning bei Aufgaben, die im Wesentlichen aus dem Erkennen und Wiedergeben vorhandener Informationen bestehen, wenig zusätzlichen Nutzen bringt. Auch die Anbieter tragen dem Rechnung: OpenAI empfiehlt für latenzkritische Aufgaben wie schnelle Informationsabfrage oder Klassifikation, Reasoning zu minimieren oder abzuschalten (OpenAI, o. D.), und bei Anthropic kann das Modell bei einfachen Faktenfragen selbst entscheiden, auf Reasoning zu verzichten (Anthropic, o. D.-b). Für solche Aufgaben kann ein Modell ohne Reasoning ausreichend sein.

3.2 Aufgaben, bei denen Reasoning einen Mehrwert erwarten lässt

Anders verhält es sich, wenn die übergebenen Informationen nicht nur wiedergegeben, sondern weiterverarbeitet werden müssen. Krishna et al. (2025) untersuchten mit dem Benchmark FRAMES Fragen, deren Beantwortung Informationen aus mehreren Wikipedia-Artikeln erfordert. Selbst als Gemini 1.5 Pro alle benötigten Artikel vollständig im Kontext erhielt, beantwortete das Modell rund 27 % der Fragen falsch. Etwa 80 % dieser Fehler betrafen Fragen mit Berechnungen, der Auswertung tabellarischer Daten oder nachgelagerten Verarbeitungsschritten. Die benötigten Fakten lagen also vor, wurden aber nicht korrekt weiterverarbeitet. Dies deckt sich mit Sprague et al. (2025): Neben mathematischen und logischen Aufgaben profitierte dort unter anderem MuSR von Chain-of-Thought, ein Datensatz, der mehrstufiges Schließen über einen vorgegebenen Text erfordert. In einfachen RAG-Systemen betrifft dies etwa Berechnungen auf Basis abgerufener Werte, die Auswertung von Tabellen, Vergleiche zwischen Angaben aus mehreren Chunks oder die Prüfung, ob mehrere Bedingungen aus technischen Vorgaben gleichzeitig erfüllt sind. Ein industrielles Beispiel ist der Abgleich einer Fehlermeldung mit Angaben aus Wartungshandbuch und Ersatzteilkatalog, aus dem eine Maßnahme abgeleitet werden muss. Bei solchen Aufgaben ist der potenzielle Nutzen von Reasoning deutlich höher als bei der reinen Wiedergabe von Informationen.

4. Aussagekraft und Grenzen des KI-ssist Cost Checks

Für die Einordnung der Ergebnisse des KI-ssist Cost Checks sind sein Einsatzbereich und die Größen relevant, die er methodisch bewusst offenlässt.

4.1 Was der Rechner leistet

Der Rechner ermöglicht einen standardisierten und transparenten Vergleich von Input- und Outputkosten über Modelle und Anbieter hinweg, für unterschiedliche RAG-Konfigurationen und Anfragevolumina. Er beantwortet damit Fragen wie: Was passiert mit den Kosten, wenn statt drei zehn Chunks an das Modell übergeben werden? Wie wirken sich Chunkgröße, Top-k, Promptlänge und Antwortlänge aus? Wie unterscheiden sich Anbieter unter denselben Tokenannahmen? Für Modelle ohne Reasoning oder Konfigurationen, in denen Reasoning deaktiviert oder minimiert ist, ist diese Betrachtung unmittelbar aussagekräftig. Bei aktivem Reasoning beschreibt sie den Kostenanteil, der unabhängig vom Reasoning entsteht.

4.2 Was der Rechner bewusst nicht abbildet

Reasoning-Tokens werden weder zusätzlich angenommen noch prognostiziert. Dies ist eine bewusste methodische Entscheidung. Wie Abschnitt 2 zeigt, hängt ihre Anzahl vom Modell, von der Aufgabenkomplexität, von der Reasoning-Konfiguration, von der konkreten Anfrage und sogar von der einzelnen Ausführung ab. Ein pauschaler Aufschlag würde eine Präzision suggerieren, die empirisch nicht gedeckt ist, und die Vergleichbarkeit eher verschlechtern, da derselbe Faktor auf Modelle angewendet würde, deren Reasoning-Verbrauch sich bei identischer Anfrage um ein Vielfaches unterscheidet.

Hinzu kommt, dass Reasoning nicht bei allen Modellen gleich konfigurierbar ist. Bei Anthropic lässt sich Thinking für Claude Sonnet 5 vollständig deaktivieren, für Claude Opus 5 nur bis einschließlich der Effort-Stufe high. Claude Sonnet 5.5 erlaubt als niedrigste Einstellung lediglich das Abschalten des vorgelagerten Thinking, während Claude Opus 5.5 und Claude Fable 5.1 eine Deaktivierung grundsätzlich nicht zulassen. Bei adaptivem Thinking entscheidet das Modell je Anfrage selbst, ob und in welchem Umfang es Reasoning einsetzt; der Effort-Parameter ist dabei eine weiche Vorgabe ohne garantierte Tokenzahl (Anthropic, o. D.-a, o. D.-b). Auch bei OpenAI sind die zulässigen Werte des Parameters reasoning.effort modellabhängig; GPT-6 Astra unterstützt beispielsweise keine Einstellung ohne Reasoning (OpenAI, o. D.). Eine einheitliche Reasoning-Annahme für alle Modelle wäre daher nicht sachgerecht. Ebenfalls nicht abgebildet sind mehrstufige agentische Abläufe, deren Kosten zusätzlich von der Zahl der Modellaufrufe und dem wiederholt übergebenen Kontext abhängen (L. Chen et al., 2026).

5. Fazit

Der KI-ssist LLM Provider Cost Check beantwortet eine bewusst standardisierte Kostenfrage: Welche Kosten entstehen bei gegebenen Input- und Output-Tokenmengen und einem definierten Anfragevolumen? Für diese Frage ist er ein sinnvolles und transparentes Werkzeug, um Anbieter zu vergleichen und die Kostenhebel eines RAG-Designs wie Chunkanzahl, Chunkgröße, Promptlänge und Antwortlänge zu verstehen. Bei Reasoning-Modellen hängen die tatsächlichen Kosten zusätzlich vom internen Reasoning-Aufwand ab, der zwischen

Modellen, Aufgaben und sogar einzelnen Ausführungen erheblich variiert. Für eine konkrete Produktionsentscheidung, insbesondere bei reasoning-intensiven Anwendungen, sollte die Kalkulation daher durch Messungen mit repräsentativen realen Anfragen ergänzt werden. Der KI-ssist LLM Provider Cost Check bietet damit eine belastbare Grundlage für die Modellwahl bei Aufgaben ohne Reasoning-Bedarf, sollte bei komplexeren RAG-Anfragen mit Reasoning aber nicht die alleinige Entscheidungsbasis darstellen.

Literatur

- Anthropic. (o. D.-a). *Thinking*. Claude Platform Docs. Abgerufen am 7. Oktober 2026, von <https://platform.claude.com/docs/en/build-with-claude/thinking>
- Anthropic. (o. D.-b). *Steering thinking*. Claude Platform Docs. Abgerufen am 7. Oktober 2026, von <https://platform.claude.com/docs/en/build-with-claude/thinking-steering-and-cost>
- Chen, L., Zhang, C., He, Y., Stoica, I., Zaharia, M., & Zou, J. (2026). *The price reversal phenomenon: When cheaper reasoning models cost more* (Version 2) [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2603.23971>
- Chen, W.-L., Peng, L., Tan, T., Zhao, C., Chen, J., Lin, Z., Go, A., & Meng, Y. (2026). Think deep, not just long: Measuring LLM reasoning effort via deep-thinking tokens. In *Proceedings of the 43rd International Conference on Machine Learning* (Proceedings of Machine Learning Research, Bd. 306, S. 16342–16360). PMLR. <https://proceedings.mlr.press/v306/chen26di.html>
- Chen, X., Xu, J., Liang, T., He, Z., Pang, J., Yu, D., Song, L., Liu, Q., Zhou, M., Zhang, Z., Wang, R., Tu, Z., Mi, H., & Yu, D. (2025). Do NOT think that much for $2+3=?$: On the overthinking of long reasoning models. In *Proceedings of the 42nd International Conference on Machine Learning* (Proceedings of Machine Learning Research, Bd. 267, S. 9487–9499). PMLR. <https://proceedings.mlr.press/v267/chen25bx.html>
- Krishna, S., Krishna, K., Mohananey, A., Schwarcz, S., Stambler, A., Upadhyay, S., & Faruqui, M. (2025). Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies* (Bd. 1: Long Papers, S. 4745–4759). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2025.naacl-long.243>
- Kumar, A., Roh, J., Naseh, A., Karpinska, M., Iyyer, M., Houmansadr, A., & Bagdasarian, E. (2026). *OverThink: Slow-down attacks on reasoning LLMs* (Version 5) [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2502.02542>
- OpenAI. (o. D.). *Reasoning models*. OpenAI API documentation. Abgerufen am 7. Oktober 2026, von <https://developers.openai.com/api/docs/guides/reasoning>
- Sprague, Z., Yin, F., Rodriguez, J. D., Jiang, D., Wadhwa, M., Singhal, P., Zhao, X., Ye, X., Mahowald, K., & Durrett, G. (2025). To CoT or not to CoT? Chain-of-thought helps mainly on math and symbolic reasoning. In *The Thirteenth International Conference on Learning Representations (ICLR 2025)*. https://proceedings.iclr.cc/paper_files/paper/2025/hash/ead542f13a38179d1b55b88610f959a1-Abstract-Conference.html