Modellauswahl & Optimierung

← Zurück zur LibreChat-Menü

Ziel dieser Seite: Diese Seite kombiniert Modellauswahl (inkl. Reasoning-Modelle) und Parameteroptimierung. Für UI-Details siehe Benutzeroberfläche verstehen, für Workflows Erweiterte Funktionen nutzen, für Schnellfragen die Häufig gestellte Fragen (FAQ).

Übersicht

Für wen ist diese Seite? Fortgeschrittene Benutzer, die gezielt Modelle auswählen und Parameter anpassen möchten, um optimale Ergebnisse zu erzielen.

Was erwartet Sie?

- **Teil 1**: Wann welches Modell wählen – inkl. Reasoning-Modelle und Agenten
- **Teil 2**: Alle Parameter verstehen und optimal einstellen

Für Einsteiger: Wenn Sie neu bei LibreChat sind, **empfehlen wir zunächst Standardwerte zu nutzen** oder einen passenden Agenten zu wählen. Diese Seite ist für erfahrene Nutzer, die mehr Kontrolle wünschen.

Teil 1: Modellauswahl

Standardmodelle für typische Aufgaben

- **Allround / Büroarbeit**: `gpt-4.1-mini` (OpenAI) oder `meta-llama-3.1-8b-instruct` (GWDG, kostenfrei)
- **Technische & wissenschaftliche Analysen**: `qwen3-30b-a3b-instruct-2507` (GWDG)
- **Kreative Inhalte / Bilder**: `llama-3.1-sauerkrautlm-70b`, `gpt-4.1-mini`
- **Aktuelle Informationen**: „GPT-4.1 Web Search„ oder „GPT-5-mini Web Search“ (Agenten)

Spezialisierte KI-Agenten im Überblick

LibreChat bietet verschiedene spezialisierte KI-Agenten, die für spezifische Aufgaben optimiert sind. Agenten kombinieren vorkonfigurierte Modelle, Parameter und Prompts für optimale Ergebnisse.

GPT-4.1 EverydayDraft-Assistent

Ideal für: Übersetzungen, Textbearbeitung, einfache Code-Generierung und Datenanalysen

Verwendung:

- Übersetzungen zwischen verschiedenen Sprachen
- Texte verbessern, umformulieren oder zusammenfassen
- Einfache Programmcode-Generierung
- Basis-Datenanalysen

Praktische Beispiele: Siehe Anwendungsbeispiele - Texte bearbeiten, Programmcode generieren und Daten analysieren

GPT-5 LogicForge Expert

Ideal für: Komplexe Aufgaben wie anspruchsvolle Code-Generierung, tiefgehende Datenanalysen und Dateiüberarbeitungen

Verwendung:

- Komplexe Programmcode-Generierung und Debugging
- Tiefgehende Datenanalysen
- Dateien überarbeiten und mehrere Quellen kombinieren
- Aufgaben, die logische Ableitung und Analyse erfordern

Praktische Beispiele: Siehe Anwendungsbeispiele - Komplexe Aufgaben bearbeiten

Technische Details: Vorkonfigurierter Reasoning-Agent für Mathe, Logik, Code. Prompt-Kern: „Parse the request → Step-by-step plan → Show derivations → Provide tests → End with Next Steps“. Ideal mit GPT-5/o4-mini + Use Responses API aktiviert.

GPT-5-mini DeepWiki Helper

Ideal für: Intelligente Analyse technischer Dokumentationen

Verwendung:

- Technische Dokumentationen durchsuchen und analysieren

- Code-Beispiele und API-Referenzen finden
- Technische Fragen mit Quellen beantworten

Praktische Beispiele: Im Marktplatz auswählen → „Wie funktioniert OAuth2 in Express.js?„ → Agent liefert DeepWiki-Ergebnisse mit Quellen, ohne den Browser zu wechseln.

Technische Details: Kombiniert GPT-5-mini mit DeepWiki für intelligente Analyse technischer Dokumentationen

GPT Bildgenerierung

Ideal für: Erstellung von Bildern basierend auf Textbeschreibungen

Verwendung:

- Bilder aus Textbeschreibungen generieren
- Wissenschaftliche Diagramme erstellen
- Visuelle Konzepte visualisieren

Praktische Beispiele: Agent wählen → Prompt „Erstelle ein Bild von …“ → Bild generieren und herunterladen. Weitere Beispiele: Anwendungsbeispiele - Bilder generieren lassen

GPT-4.1 / GPT-5-mini Web Search

Ideal für: Aktuelle Informationen und Live-Websuche

Verwendung:

- Tagesaktuelle Fragen beantworten
- Neueste Nachrichten und Forschungsergebnisse recherchieren
- Live-Websuche für aktuelle Informationen

Praktische Beispiele:

- **GPT-4.1 web search**: Für tagesaktuelle Fragen („Neueste Nachrichten zu …?„) → Agent startet Echtzeitsuche → liefert aktuellste Infos
- **GPT-5-mini web search**: Gleicher Ablauf, aber schneller und günstiger für leichtere Recherchen
- Weitere Beispiele: Anwendungsbeispiele - Suche/Recherche

Technische Details:

- **GPT-4.1 web search**: Kombiniert GPT-4.1 mit Live-Websuche für aktuelle Informationen
- **GPT-5-mini web search**: Kombiniert GPT-5-mini mit Live-Websuche für schnelle Recherchen

Reasoning-Agenten (Math & Logic, Code Debugging, Deep Analysis)

Ideal für: Komplexe mathematische Beweise, Logikrätsel, tiefes Debugging

Verwendung:

- Mathematische Beweise und Logikrätsel lösen
- Komplexes Code-Debugging
- Tiefgehende Analysen mit nachvollziehbaren Begründungen

Technische Details: Bereitgestellte Reasoning-Presets für komplexe Fälle. Modelle, Parameter und Prompts sind bereits vorkonfiguriert.

Weitere Informationen: Siehe Abschnitt „Reasoning-Modelle“ weiter unten und FAQ - Reasoning-Modelle

Best Practices für Agenten

- **Gezielt einsetzen**: Einfache Aufgaben weiterhin normalen Modellen überlassen, Agenten für spezialisierte Aufgaben nutzen (siehe oben „Ideal für“ bei jedem Agenten)
- **Beta-Funktionen im Blick behalten**: Bildgenerierung kann kurzfristig deaktiviert werden
- **Agenten kombinieren**: Für komplexe Aufgaben können Sie Agenten abwechselnd einsetzen (z.B. LogicForge Expert für Analyse, EverydayDraft-Assistent für Formatierung)

Reasoning-Modelle: Wann lohnt sich Reasoning?

Warum Reasoning trotzdem ein Thema bleibt:

1. **Standardnutzer**: Wählen einfach den passenden Agenten (z.B. „Deep Analysis“, „Math & Logic„, „Code Debugging“). Dort sind Modelle, Parameter und Prompts bereits vorkonfiguriert.
2. **GPT-5/GPT-5.1**: Erkennen die Komplexität Ihrer Frage automatisch und aktivieren bei Bedarf das interne „Thinking„. Sie müssen nichts einstellen.
3. **Fortgeschrittene**: Können weiterhin in den KI-Einstellungen `Reasoning Effort`, `Reasoning Summary` und `Verbosity` anpassen, wenn sie volle Kontrolle möchten.

Wann lohnt sich Reasoning?:

1. Mathematische Beweise, Logikrätsel, komplexes Debugging, mehrstufige Recherchen oder Entscheidungsbäume.
2. Aufgaben, die sich sauber in Zwischenschritte aufteilen lassen („Zeige mir Schritt für Schritt…“).
3. Wenn Sie nachvollziehbare Begründungen oder alternative Szenarien benötigen.

Nicht nötig ist Reasoning bei kurzen Faktenfragen, normalem Textschreiben oder schnellen Brainstorming-Sessions – dort sind `gpt-4.1-mini` oder leichte GWDG-Modelle deutlich schneller.

Empfohlene Reasoning-Modelle:

1. **Komplexe Logik, Mathe, Debugging**: Reasoning-Agenten oder `o4-mini`, `qwen3-30b-a3b-thinking-2507`, `deepseek-r1`

Wenn Reasoning-Antworten lange dauern

1. Vergewissern Sie sich, dass wirklich Reasoning erforderlich ist.
2. Reduzieren Sie den `Reasoning Effort` (Medium/Low) oder setzen Sie `Reasoning Summary` auf *Concise*.
3. Geben Sie eine klare Aufgabenbeschreibung bzw. laden Sie relevante Dateien hoch.
4. Nutzen Sie erst ein leichtes Modell zum Ideensammeln und schalten Sie für die finale Analyse auf einen Reasoning-Agenten.

Ausführlichere Tipps finden Sie im Abschnitt „Reasoning-Modelle: Häufige Fragen„ der FAQ.

Teil 2: Parameteroptimierung

Wo finde ich die Modellparameter?

- Öffnen Sie das **rechte Seitenmenü** (siehe Benutzeroberfläche)
- Klicken Sie auf **„KI-Einstellungen“**
- Hier sehen Sie alle verfügbaren Parameter

Zuerst die Grundlagen lernen? Lesen Sie zuerst die Benutzeroberfläche, um zu verstehen, wo Sie diese Einstellungen finden.

KI-Einstellungen im Überblick

Basis & Kontext

- **Benutzerdefinierter Name**: Vergeben Sie einen eindeutigen Namen für die aktuelle Konfiguration (z. B. „Formelle E-Mails“), damit Presets leichter zugeordnet werden können.
- **Benutzerdefinierte Anweisungen**: Dauerhafte Vorgaben wie „Antworte formell“ oder „Handle als Python-Experte“; Standard leer lassen, wenn keine Sonderrolle erforderlich ist.
- **Max. Kontext-Token**: Obergrenze für das „Chat-Gedächtnis“. Nur bei extrem langen Dokumenten erhöhen.
- **Max. Antwort-Token**: Steuerung der Antwortlänge – Richtwerte: 500 (kurz), 2000 (Analyse), >4000 (sehr lang).

Kreativität & Vielfalt

- **Temperature**: Steuert Kreativität – 0.0-0.3 fokussiert, 0.4-0.7 Standard, 0.8-1.0 kreativ.
- **Top P**: Alternative zur Temperature; Standard 1.0 lassen und primär Temperature nutzen.
- **Frequency Penalty**: Reduziert Wiederholungen einzelner Wörter (typisch 0-0.5).
- **Presence Penalty**: Fördert neue Themen; für Brainstorming 0.3-0.6, sonst 0.0.

Dateien, Bilder & Suche

- **Stop-Sequenzen**: Optionale Stop-Zeichen (z.B. „Ende“, „\n\n“) für Spezialfälle.
- **Anhänge erneut senden**: Sendet Anhänge bei jeder Nachricht erneut – nur aktivieren, wenn häufig darauf verwiesen wird.
- **Bild-Detail**: Niedrig = schnell/grob, Hoch = detailliert, Auto = automatisch; nur bei Bild-Uploads relevant.
- **Web-Suche**: Bindet Live-Suchergebnisse ein; oft sind vorkonfigurierte Web-Suchagenten bequemer.

Reasoning & Antworten-API

Wichtig: Aktivieren Sie **„Responses-API nutzen“**, bevor Sie Reasoning-Parameter anpassen. Nur dann sind die erweiterten Modellfunktionen verfügbar.

- **Responses-API nutzen**: Aktiviert die neue Antworten-API, Voraussetzung für Reasoning-Parameter und spezielle Funktionen.
- **Denkaufwand (Reasoning Effort)**: Nur bei GWDG-Reasoning-Modellen (Low = schnell, Medium = Standard, High = maximale Tiefe).
- **Zusammenfassung des Nachdenkens (Reasoning Summary)**: None/Knapp = kurze Ausgaben, Detailed = vollständige Denkpfade (nur bei Reasoning-Modellen).
- **Ausführlichkeit (Verbosity)**: Steuert die Gesamtlänge (Low = knapp, Medium = Standard, High = ausführlich).

Streaming & Limits

- **Streaming deaktivieren**: Aktiviert = Ausgabe erfolgt erst am Ende, Standard = Ausgabe live; nur bei Bedarf ändern.
- **Datei-Token-Limit**: Begrenzung der Tokens aus hochgeladenen Dateien; nur bei sehr großen Dateien erhöhen.

Typische Presets für häufige Aufgaben

1. **Fakten & Recherche**: Temperature 0.2, Top P 1.0, keine Penalties → präzise Antworten ohne

Ausschweifen.
2. **Code & Debugging**: Temperature 0.2–0.3, Custom Instructions „Du bist ein Python-Experte„, Max Output ≥ 2000 Tokens.
3. **Kreatives Schreiben**: Temperature 0.8–1.0, Presence/Frequency Penalty 0.3–0.5 → vielfältige Ideen.
4. **Formelle Texte**: Temperature 0.5, Custom Instructions „Antworte in formeller Sprache“, Frequency Penalty 0.2.
5. **Zusammenfassungen**: Temperature 0.3, Max Output ca. 500 Tokens, Custom Instructions „Fasse prägnant zusammen„.
6. **Reasoning (schnell)**: Use Responses API aktivieren, Reasoning Effort low, Reasoning Summary Concise, Verbosity none.
7. **Reasoning (ausgewogen)**: Use Responses API aktivieren, Reasoning Effort medium, Reasoning Summary Auto, Verbosity middle.
8. **Reasoning (tiefgehend)**: Use Responses API aktivieren, Reasoning Effort high, Reasoning Summary Detailed, Verbosity middle.

Nützliche Funktionen

Modellparameter zurücksetzen

- Setzt alle Werte auf Werkseinstellungen zurück – ideal nach Experimente.

Als Voreinstellung speichern

- Speichert die aktuelle Konfiguration (z.B. „Formelle E-Mails“, „Code-Hilfe„, „Reasoning-Analyse“) für schnellen Wechsel zwischen Aufgaben.

Weiterführende Links

- Anwendungsbeispiele - Erste Schritte mit LibreChat
- Benutzeroberfläche von LibreChat
- FAQ – Schnellhilfe für LibreChat
- Modellauswahl & Optimierung
- Erste Schnitte
- Erweiterte Nutzungsanleitung
- Sprach-Ein- und Ausgabe in LibreChat
- Temporäre Chats

← Zurück zur Startseite

Direkt-Link:
https://doku.tu-clausthal.de/doku.php?id=sonstige_dienste:ki-dienste:librechat:modellauswahl-optimierung&rev=1764755015

Letzte Aktualisierung: **09:43 03. December 2025**