

# Lokale LLMs

Workshop zu einem (eigenen)  
lokalen Setup



*Image created with DALL-E 3*

# Was ist unser Ziel?

💡 Herauszufinden, wie man LLMs  
lokal auf dem eigenen  
PC, Laptop, etc.  
*zum laufen bringt, wie das geht und*  
**ob das schwierig ist...** 🤔

...Spoiler: **nein!**



© Ollama



© Msty

## Kurze Abfrage

...wer verwendet bereits lokale LLMs, 

...wer nutzt eigens Software dafür, 

...wer ist hier, um zu lernen, wie das geht? 



## ⚠ Disclaimer

...wer einen eigenen PC dabei hat sei sich bitte bewusst, dass Software von dritten installiert wird und zwar auf eigene „Gefahr“ !

 [ollama.com](https://ollama.com)

## Was ist Ollama?

- Ein **LLM-Backend** zur Ausführung und Verwaltung von großen Sprachmodellen (LLMs).
- **CMD-Tool & API** zur einfachen Nutzung im Terminal oder zur Integration in eigene Anwendungen (API).
- Unterhalten eine öffentliche **Modellbibliothek** mit vielen (vortrainierten) open-source KI-Modellen.
- **Optimiert für Coding & KI-Anwendungen**, kann Modelle lokal ausführen.
- **Gute Dokumentation** für einfache Einrichtung und Nutzung.  
<https://github.com/ollama/ollama/tree/main/docs>

# Ollama

Warum Ollama

Pro / Contra

# Ollama

Warum Ollama  
Pro / Contra

## [ollama.com](https://ollama.com)

### Pros

- ✓ Leistungsstarkes **LLM-Backend** für lokale KI-Modellnutzung
- ✓ **Einfache Installation & Nutzung** über Terminal (CMD-Tool) oder API
- ✓ **Keine Cloud-Abhängigkeit**, läuft lokal auf dem eigenen System

### Cons

- ✗ **Hohe Hardware-Anforderungen**, benötigt viel Speicher & Rechenleistung
- ✗ **Keine eigene Nutzer-Oberfläche**, nur CLI & API-Schnittstellen- es gibt aber zahlreiche Lösungen [AnythingLLM](#), [Any LLM](#), [OpenWebUI](#), [Msty](#)



## Was ist Msty?

- Msty ist eine **moderne KI-Chat-UI**, die es ermöglicht, mit verschiedenen KI-Modellen zu interagieren – sowohl **lokal als auch online**.
- **Einfache Nutzung von lokalen und Online-AI-Modellen:** Msty ermöglicht die nahtlose Integration und Nutzung sowohl lokal gehosteter als auch Online-AI-Modelle.
- **Offline-First, Online-Ready:** Das Tool ist für den Offline-Betrieb optimiert, unterstützt aber auch Online-Modelle für maximale Flexibilität.
- **Umfassende Modellunterstützung:** Kompatibel mit Modellen von u.a. HuggingFace, Ollama und Open Router, etc.
- **Datenschutz:** Persönliche Informationen verlassen niemals das eigene Gerät, was höchste Privatsphäre gewährleistet.

**Msty**  
**Warum Msty**  
**Pro / Contra**

# Msty

## Warum Sty

### Pro / Contra

#### [msty.app](#)

##### Pros

- ✓ **Flexibel** – nutzt lokale & Online-KI-Modelle
- ✓ **Hoher Datenschutz** – Daten verlassen nicht das Gerät
- ✓ **Zahlreiche Features**, wie **RAG** (Chat with Documents), **Websuche**, etc.

##### Cons

- ✗ **Freemium:** „always free“ Version für private Nutzung und Premium-Abos
- ✗ **Closed Source** – kein Open-Source-Projekt

<https://docs.msty.app>

# Womit arbeiten wir heute?

Einige Schlagworte...

## Übersicht:

-  **Ollama** – besagtes LLM-Backend, kümmert sich um den Betrieb der LLMs auf unserer Hardware
-  **Terminal / Command Line** – das Eingabefeld auf dem PC oder innerhalb VSCode, in dem wir arbeiten und mit dem wir dem PC Befehle geben werden
-  **API** – Application-Programming-Interface: Schnittstelle, um eine Software mittels Code anzusprechen
-  **Python** – eine Programmiersprache, die sich hervorragend zum Programmieren lernen eignet
-  **Msty** – eine äußerst praktische grafische Nutzeroberfläche für die Arbeit mit LLMs

## Na dann mal los!

Probieren wir es mal aus...

**1** Stellt **Fragen!** Diskutiert... 

**2** Ihr könnt **nichts** kaputt machen, **keine Sorge!** 

**3** Habt **Spaß** 

▷ <https://ollama.com/download>



# Ollama installieren

Wenn alles im Leben so  
einfach wäre...

Im Downloads-Ordner (**Win/Mac**) die  
Installationsdatei ausführen

**Linux** terminal cmd:

`curl -fsSL https://ollama.com/install.sh | sh`

...geschafft!

# Cheatsheet

Übersicht über die wichtigsten Befehle für Ollama...

```
C:\Users\welz\Desktop>ollama --help
Large language model runner

Usage:
  ollama [flags]
  ollama [command]

Available Commands:
  serve      Start ollama
  create     Create a model from a Modelfile
  show       Show information for a model
  run        Run a model
  stop       Stop a running model
  pull       Pull a model from a registry
  push       Push a model to a registry
  list       List models
  ps         List running models
  cp         Copy a model
  rm         Remove a model
  help       Help about any command

Flags:
  -h, --help    help for ollama
  -v, --version Show version information

Use "ollama [command] --help" for more information about a command.
```

**Öffnet eine Kommandozeile od. Terminal und gibt „ollama --help“ ein und klickt „Enter“**

```
C:\Users\welz\Desktop>ollama run llama3.2:1b
>>> /help
Available Commands:
  /set      Set session variables
  /show     Show model information
  /load <model> Load a session or model
  /save <model> Save your current session
  /clear   Clear session context
  /bye     Exit
  /?, /help Help for a command
  /? shortcuts Help for keyboard shortcuts

Use """ to begin a multi-line message.

>>> |Send a message (/? for help)
```

**Nach dem starten eines Modells hilft der Befehl „/help“**

# Ollama starten

Was ist zu tun... 🤔

1. Im Projektordner *Ollama* (Win11):  
Doppelklick auf *ollama-cmd*

2. Es müsste dieser Output erscheinen:

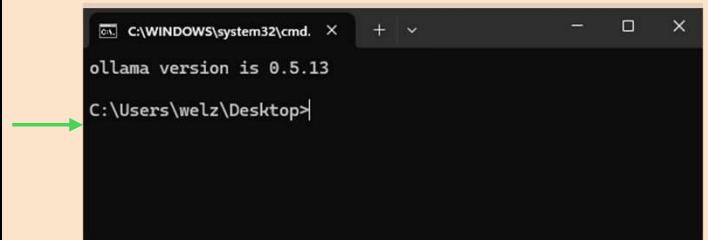
3. Nun können wir mit Ollama interagieren; Gebt mal folgendes ein:

>> *ollama list*

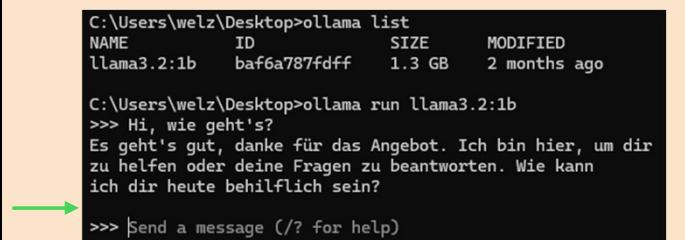
Das listet euch alle auf dem PC vorhandenen Modelle auf

4. Laden und Ausführen eines lokalen LLMs geht mit:

>> *ollama run <name-des-LLM>*



C:\WINDOWS\system32\cmd. x  
ollama version is 0.5.13  
C:\Users\welz\Desktop>



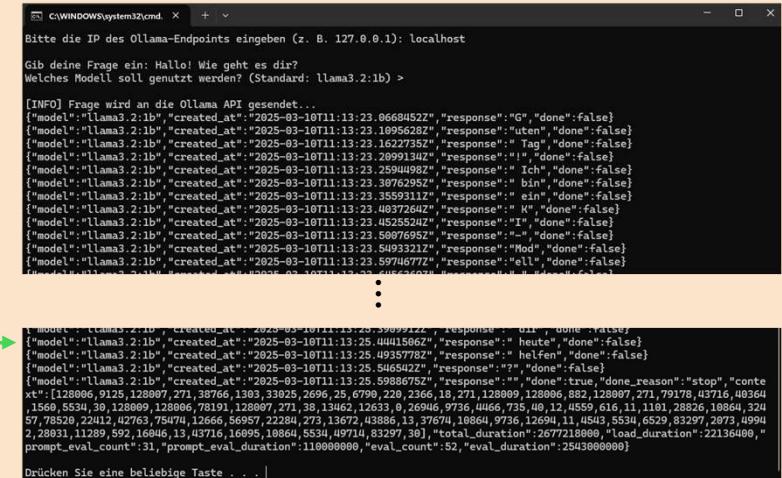
```
C:\Users\welz\Desktop>ollama list
NAME          ID        SIZE      MODIFIED
llama3.2:1b   baf6a787fdff  1.3 GB   2 months ago

C:\Users\welz\Desktop>ollama run llama3.2:1b
>>> Hi, wie geht's?
Es geht's gut, danke für das Angebot. Ich bin hier, um dir zu helfen oder deine Fragen zu beantworten. Wie kann ich dir heute behilflich sein?
>>> Send a message (/? for help)
```

# Ollama via API

Was ist zu tun... 🤔

1. Im Projektordner *Ollama* (*Win11*):  
Doppelklick auf *ollama-api*
2. Es müsste dieser Output  
erscheinen:



```
C:\WINDOWS\system32\cmd.exe + -> Welches Modell soll genutzt werden? (Standard: llama3.2:1b) >
[INFO] Frage wird an die Ollama API gesendet...
{"model": "llama3.2:1b", "created_at": "2025-03-10T11:13:23.06684527", "response": "G", "done": false}
{"model": "llama3.2:1b", "created_at": "2025-03-10T11:13:23.10956287", "response": "Utan", "done": false}
{"model": "llama3.2:1b", "created_at": "2025-03-10T11:13:23.16227357", "response": "Tap", "done": false}
{"model": "llama3.2:1b", "created_at": "2025-03-10T11:13:23.28991342", "response": "!", "done": false}
{"model": "llama3.2:1b", "created_at": "2025-03-10T11:13:23.29544092", "response": " Ich", "done": false}
{"model": "llama3.2:1b", "created_at": "2025-03-10T11:13:23.38762952", "response": " bin", "done": false}
{"model": "llama3.2:1b", "created_at": "2025-03-10T11:13:23.35593117", "response": " ein", "done": false}
{"model": "llama3.2:1b", "created_at": "2025-03-10T11:13:23.40372642", "response": " X", "done": false}
{"model": "llama3.2:1b", "created_at": "2025-03-10T11:13:23.45255242", "response": "I", "done": false}
{"model": "llama3.2:1b", "created_at": "2025-03-10T11:13:23.50976952", "response": "-", "done": false}
{"model": "llama3.2:1b", "created_at": "2025-03-10T11:13:23.59746772", "response": "Mod", "done": false}
{"model": "llama3.2:1b", "created_at": "2025-03-10T11:13:23.65623607", "response": "ell", "done": false}
Drücken Sie eine beliebige Taste . . . |
```

# Ollama via API

Was ist zu tun... 🤞

1. Im Projektordner *Ollama (Win11)*:  
Doppelklick auf  
**start-ollama-chat**
2. Netzwerk wechseln:  
**scadsai-meetup**
3. Es startet eine WebApp

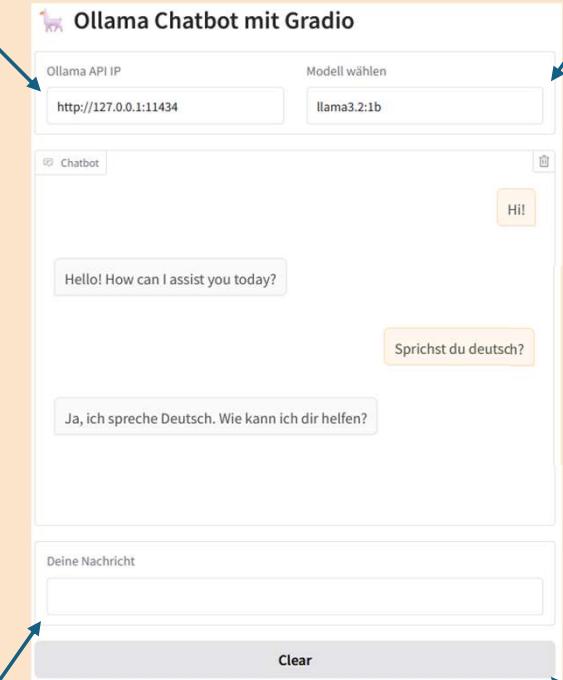
im Browser: **localhost:7860**  
eingeben

Texteingabe (Bestätigen mit *Enter*)

Reset

Adresse des (GPU) PCs für LLM-Inferenz

Modellbezeichnung



**...durchatmen** 🧘  
Schon richtig was geschafft!



...weiter geht's



# Msty installieren

Einfacher geht kaum...

► <https://docs.msty.app/getting-started/download>

Getting Started > Download

## Download

Download Msty for Windows, Mac, and Linux

Find the right Msty installer for your operating system and hardware. Whether you're on Windows, Mac, or Linux, we offer versions optimized for both CPU and GPU setups.

Windows

Choose between two versions depending on your hardware.

CPU-Only Version

Ideal for systems without a dedicated GPU.

↳ Download Msty (x64 CPU)

GPU Version

Optimized for systems with a compatible AMD or NVIDIA GPU, offering improved performance.

↳ Download Msty (x64 GPU)

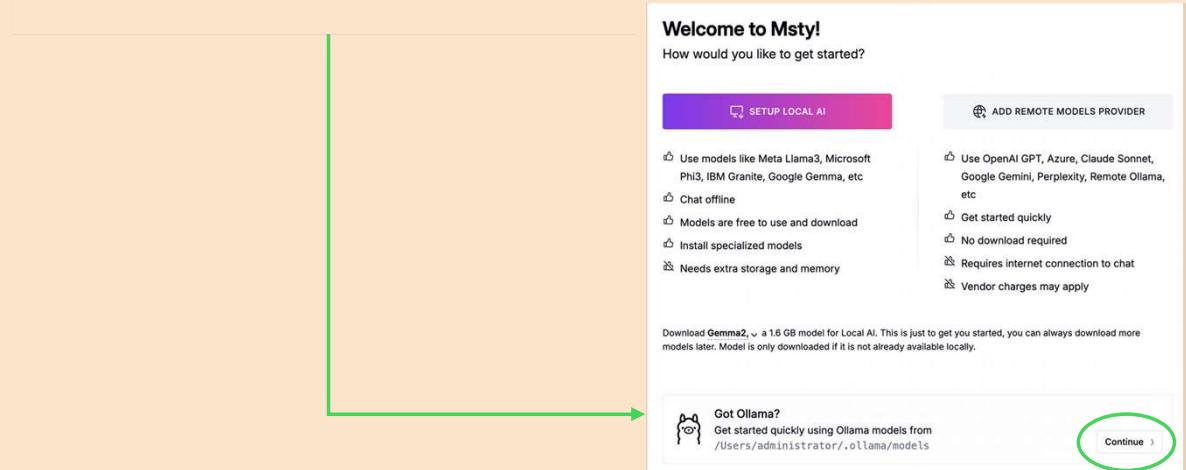
Im Downloads-Ordner die  
Installationsdatei ausführen

# Msty einrichten

Alles vorbereitet... 



Klick auf „Ollama Continue“



Welcome to Msty!

How would you like to get started?

**SETUP LOCAL AI**

- Use models like Meta Llama3, Microsoft Phi3, IBM Granite, Google Gemma, etc
- Chat offline
- Models are free to use and download
- Install specialized models
- Needs extra storage and memory

**ADD REMOTE MODELS PROVIDER**

- Use OpenAI GPT, Azure, Claude Sonnet, Google Gemini, Perplexity, Remote Ollama, etc
- Get started quickly
- No download required
- Requires internet connection to chat
- Vendor charges may apply

Download Gemma2, a 1.6 GB model for Local AI. This is just to get you started, you can always download more models later. Model is only downloaded if it is not already available locally.

Got Ollama?  
Get started quickly using Ollama models from /Users/administrator/.ollama/models

**Continue >**

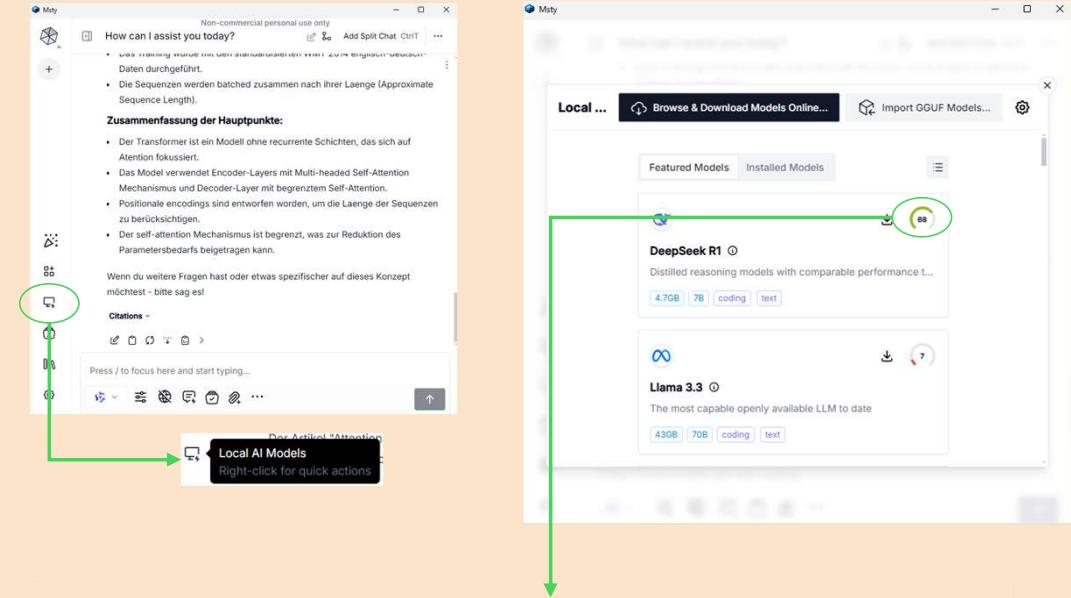
 Du wirst nun durch das **Setup** geführt und anschließend heißt es...

...start chatting! 

# Msty produktiv



Unendliche Möglichkeiten...

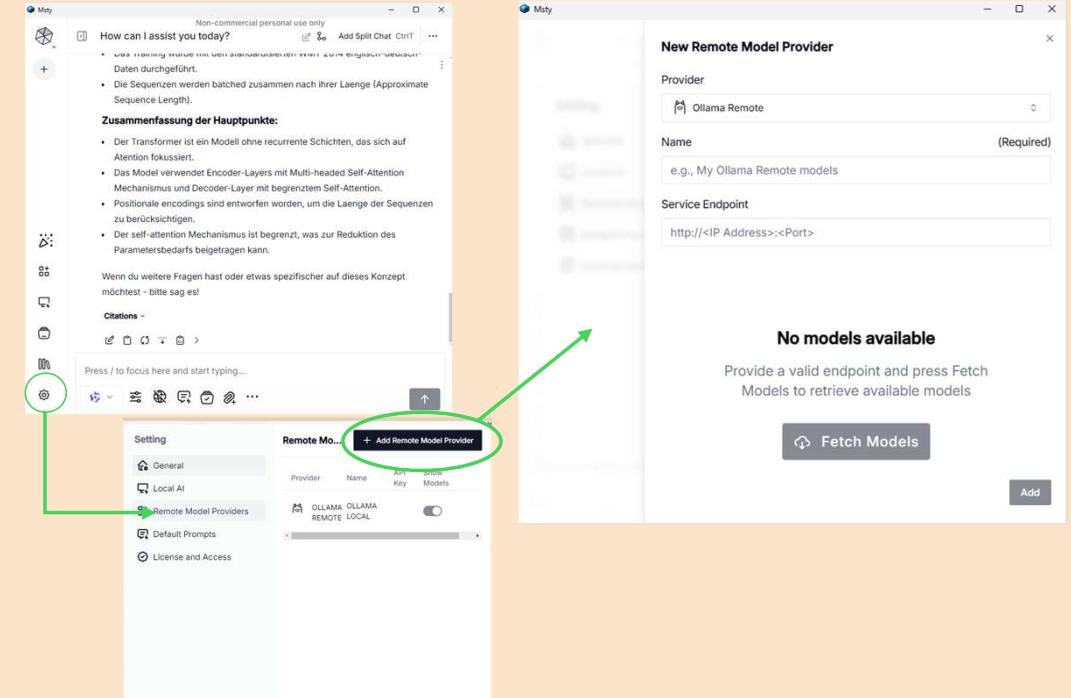


**Weitere Modelle (inkl. Anzeige, wie gut deine Hardware geeignet ist, um das Modell auszuführen)**

# Msty produktiv



Unendliche Möglichkeiten...



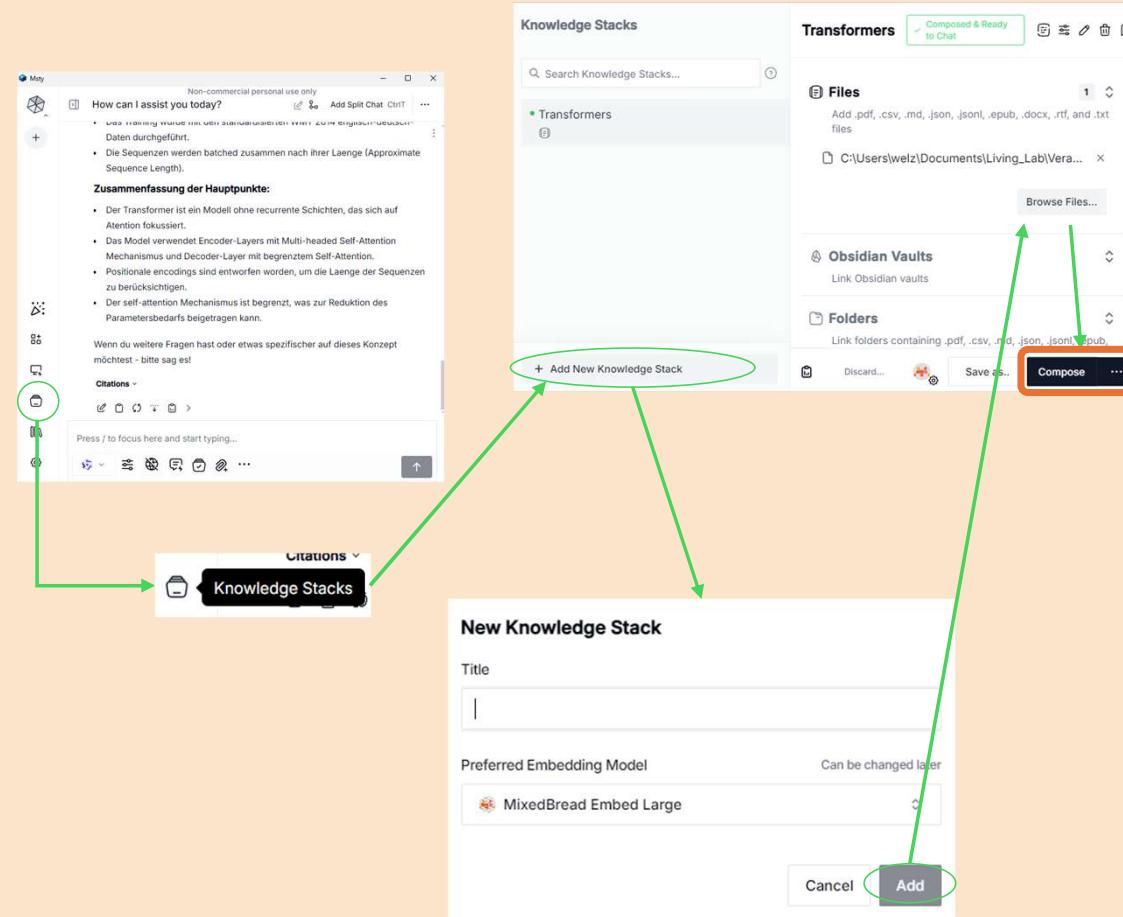
**Weitere Modell-API-Endpunkte (ähnlich wie bei Ollama via API)**

# Msty produktiv



Chat with Documents...

## Msty Knowledge Stacks

The screenshot shows the Msty application interface. On the left, a chat window displays a document about Transformers. A green arrow points from the 'Knowledge Stacks' button in the bottom right of the chat window to the 'Add New Knowledge Stack' button in the 'Knowledge Stacks' sidebar. Another green arrow points from the 'Compose' button in the 'Transformers' sidebar to the 'Add' button in the 'New Knowledge Stack' dialog box.

**Knowledge Stacks**

How can I assist you today?

Zusammenfassung der Hauptpunkte:

- Der Transformer ist ein Modell ohne recurrente Schichten, das sich auf Attention fokussiert.
- Das Modell verwendet Encoder-Layers mit Multi-headed Self-Attention Mechanismus und Decoder-Layer mit begrenztem Self-Attention.
- Positionale encodings sind entworfen worden, um die Länge der Sequenzen zu berücksichtigen.
- Der self-attention Mechanismus ist begrenzt, was zur Reduktion des Parametersbedarfs beigetragen kann.

Wenn du weitere Fragen hast oder etwas spezifischer auf dieses Konzept möchtest - bitte sag es!

Citations

+ Add New Knowledge Stack

**New Knowledge Stack**

Title

Preferred Embedding Model

MixedBread Embed Large

Add

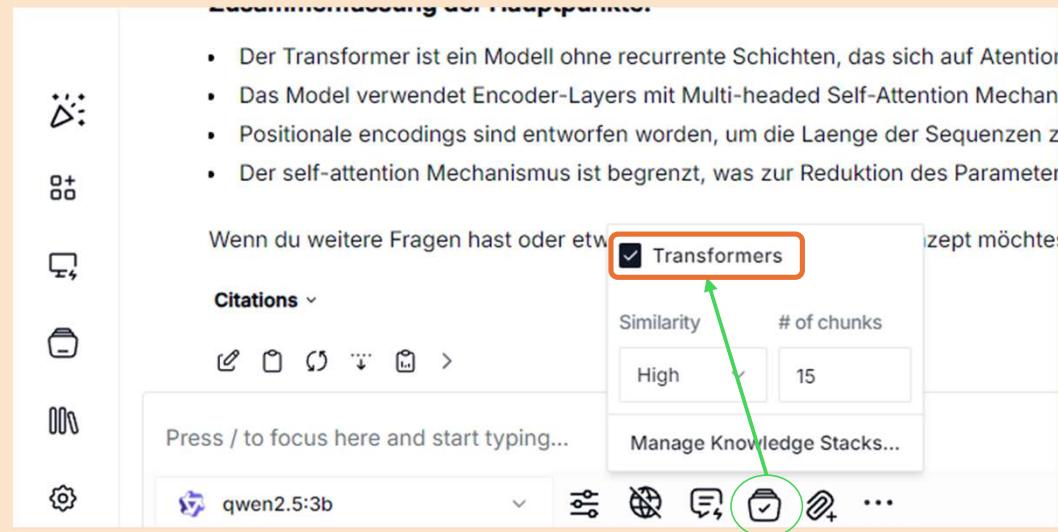
! Dokumente werden eingelesen und mittels **Embedding-Modellen** von **Text zu Tokens** umgerechnet, sprich in eine Sprache gewandelt, mit der die LLMs umgehen können. Die Dokumente sind dann im **Kontext** des LLMs und man kann fragen stellen- sprich, mit den Dokumenten *chatten*.

# Msty produktiv



Chat with Documents...

## Msty Knowledge Stacks



The screenshot shows the Sty Msty interface with a sidebar containing icons for document management and settings. The main area displays a list of bullet points about Transformers. Below this is a section for "Citations" with a dropdown menu. At the bottom, there's a search bar, a model selection dropdown set to "qwen2.5:3b", and a toolbar with various icons. A green arrow points from the "Transformers" checkbox in the dropdown menu to the "Manage Knowledge Stacks..." button in the toolbar.

- Der Transformer ist ein Modell ohne recurrente Schichten, das sich auf Attention konzentriert.
- Das Modell verwendet Encoder-Layers mit Multi-headed Self-Attention Mechanismen.
- Positionale encodings sind entworfen worden, um die Länge der Sequenzen zu handhaben.
- Der self-attention Mechanismus ist begrenzt, was zur Reduktion des Parameterbedarfs führt.

Wenn du weitere Fragen hast oder etwas über das Thema erfahren möchtest, schreibe mir eine Nachricht.

**Transformers**

Citations

Similarity # of chunks

High 15

Manage Knowledge Stacks...

Press / to focus here and start typing...

qwen2.5:3b

...

! Der Knowledge Stack kann nun **unterhalb** der Chat-Eingabe „enabled“ werden

Probiert's mal aus und findet etwas über unsere **Schwarmintelligenz Demo** heraus

## Hall of Fame

Wer hat was gemacht?



## Key-Takeways?

# Hardwareanforderungen für deinen LLM Use-Case



Image created with DALL-E 3

💻 Für **kleine Use-Cases** (dedizierte Aufgaben/Agenten, z.B. für Wetter-API o.ä.) eignen sich schon **handelsübliche Laptops** (~1-3B Parameter Modelle)

💸 Braucht es **leistungsfähigere LLMs** (ab ~8B Parameter aufwärts), sollte über eine **eGPU** oder je nach Vorhaben auch über ein **GPU Edge-Device** nachgedacht werden:  
Beispiele:

**eGPU: Grafikkarte** (z.B. RTX3060 od. RTX4060Ti)

mit **12 bzw. 16GB VRAM**

$\Sigma \sim 800\text{€}$

**eGPU-Case: z.B. Razor oder Sonnet**

**GPU Edge PC:** z.B. NVIDIA Jetson Orin Nano Super

od. NVIDIA Jetson AGX Orin (~70B Parameter Modelle möglich)

$\Sigma \sim 300\text{€} - 3000\text{€}$

## Diskussion

Wie war's,  
was habt ihr erreicht,  
für was würdet oder werdet ihr lokale LLMs  
einsetzen...

Was denkt Ihr?



# Danke!

**ScaDS.AI**  
DRESDEN LEIPZIG  
LIVING LAB



**Inspiration:** Lokales LLM auf einem USB-Stick  
© Global Science Network Channel on YouTube