

# Large Language Models

on European Supercomputers

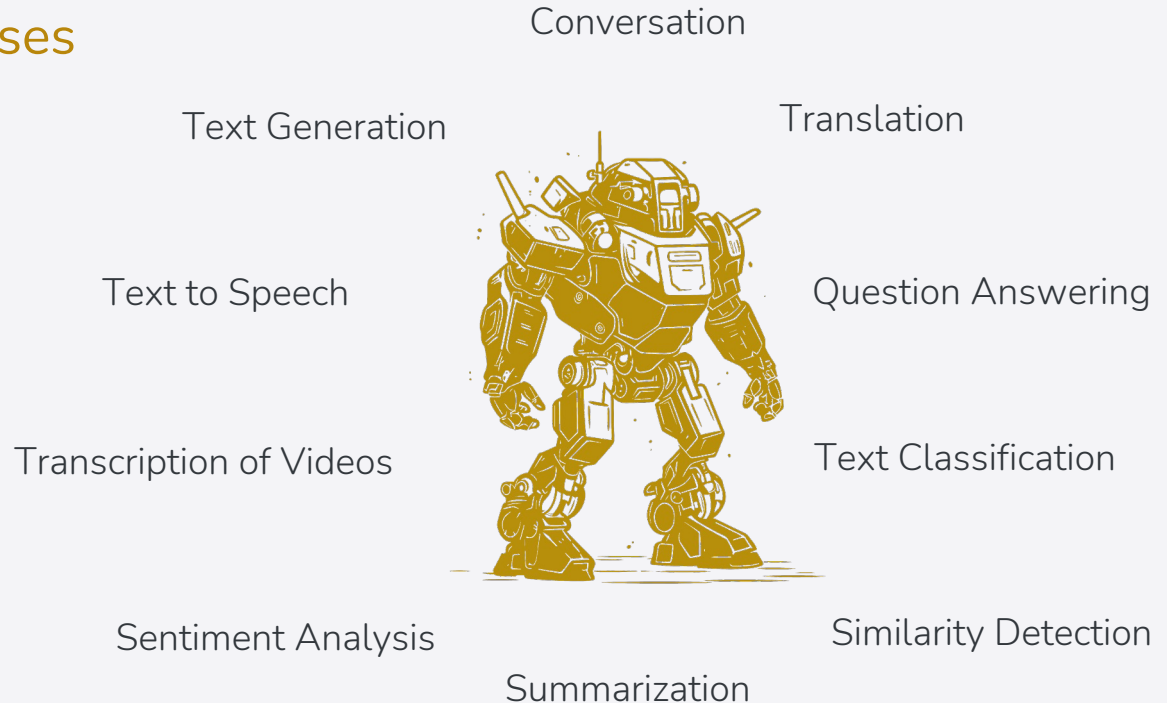
---

Speaker: Simeon Harrison  
Trainer at EuroCC Austria

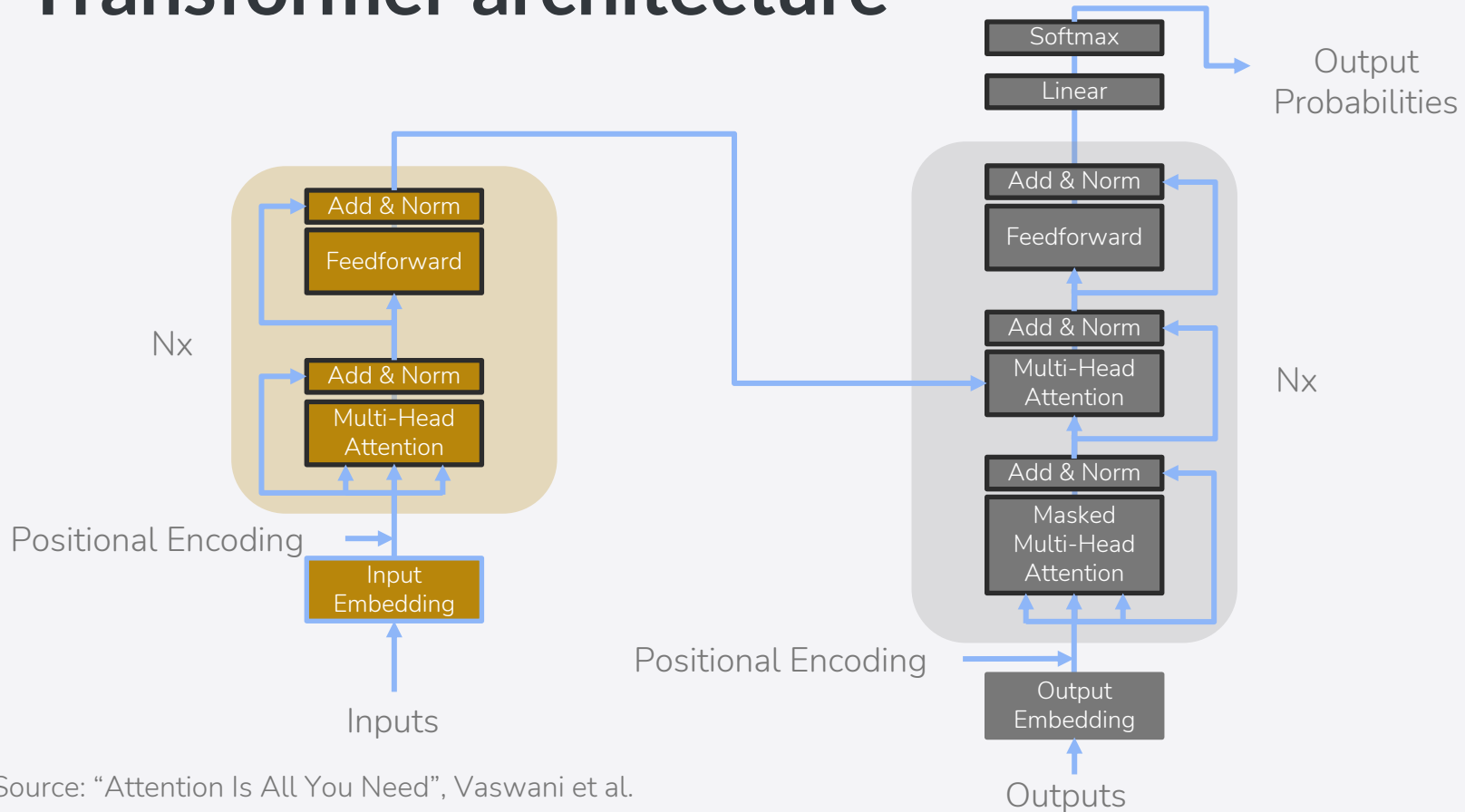
# What can LLMs be used for?

## Many different use cases

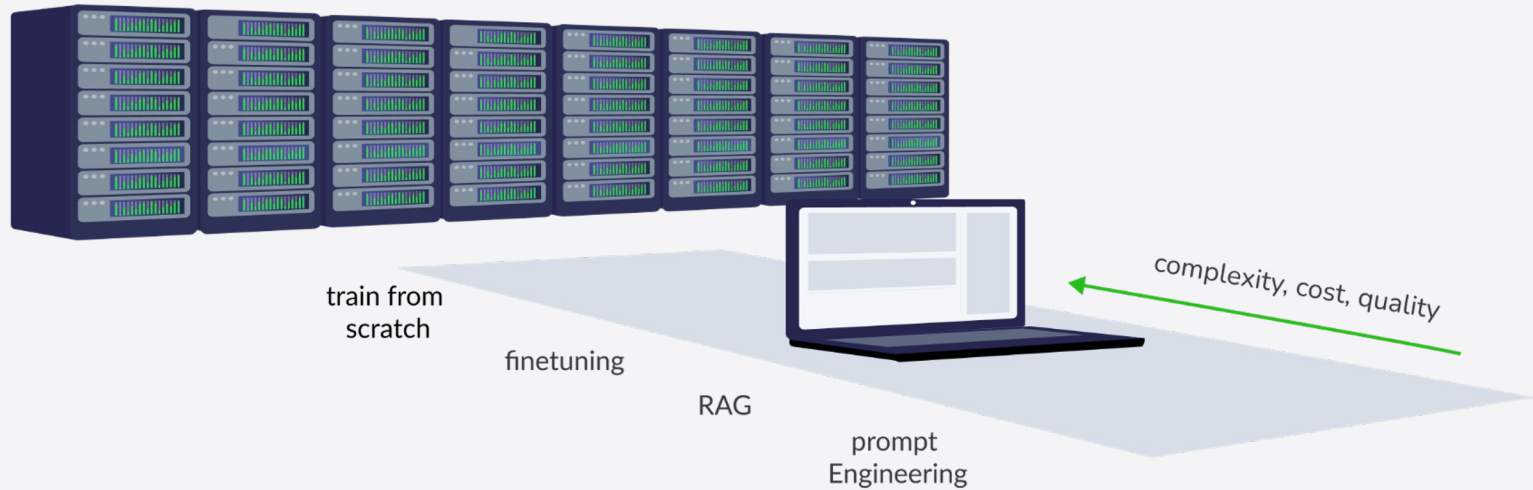
- Made possible by the transformer architecture
- Choose your model according to the use-case
- Make sure you know your use-case



# Transformer architecture



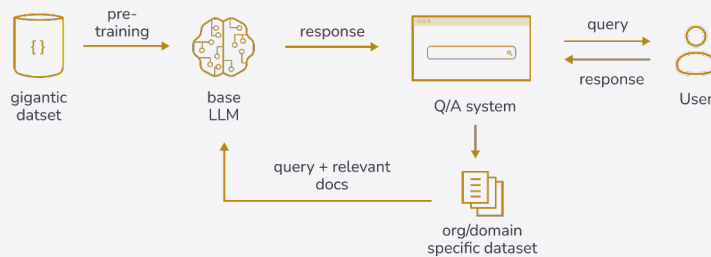
# How can you influence LLMs?



# How can you use LLMs with your data?

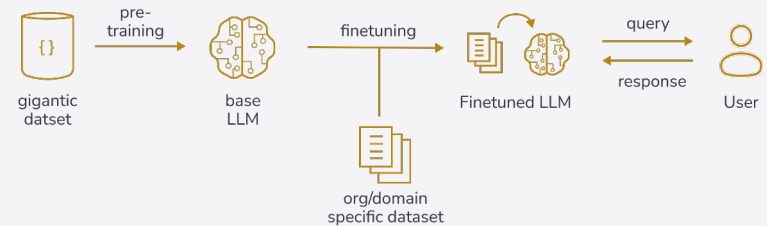
## RAG:

### Retrieval Augmented Generation



- Ideal for tapping into company's knowledge DBs
- Minimises hallucinations by grounding response on retrieved evidence
- Can quickly adapt to changing data
- Makes it easier to interpret result

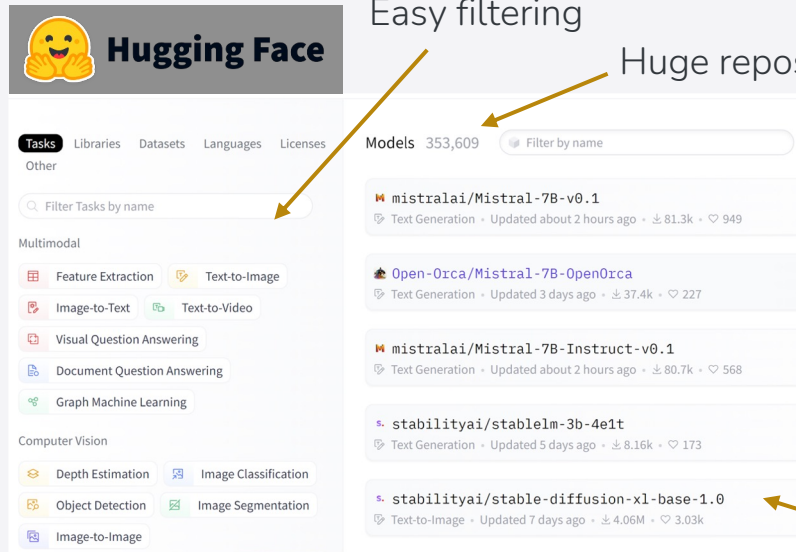
## Finetuning



- Ideal if plenty of labelled data is available
- Teaches model domain specific vocabulary
- Company's writing/answer style is „baked“ into model through fine-tuned parameters

# Transformer Models

Spoilt for Choice at <https://huggingface.co/>



Source: <https://huggingface.co/>

# Pick the Right Model

mistralai/Mistral-7B-Instruct-v0.2   like 1.04k



Text Generation



Transformers



PyTorch



Safetensors

mistral

finetuned

conversational



arxiv:2310.06825



License: apache-2.0



Model card



Files and versions



Community **61**



Edit model card

## Model Card for Mistral-7B-Instruct-v0.2

The Mistral-7B-Instruct-v0.2 Large Language Model (LLM) is an improved instruct fine-tuned version of [Mistral-7B-Instruct-v0.1](#).

For full details of this model please read our [paper](#) and [release blog.post](#).

# Prepare your Data

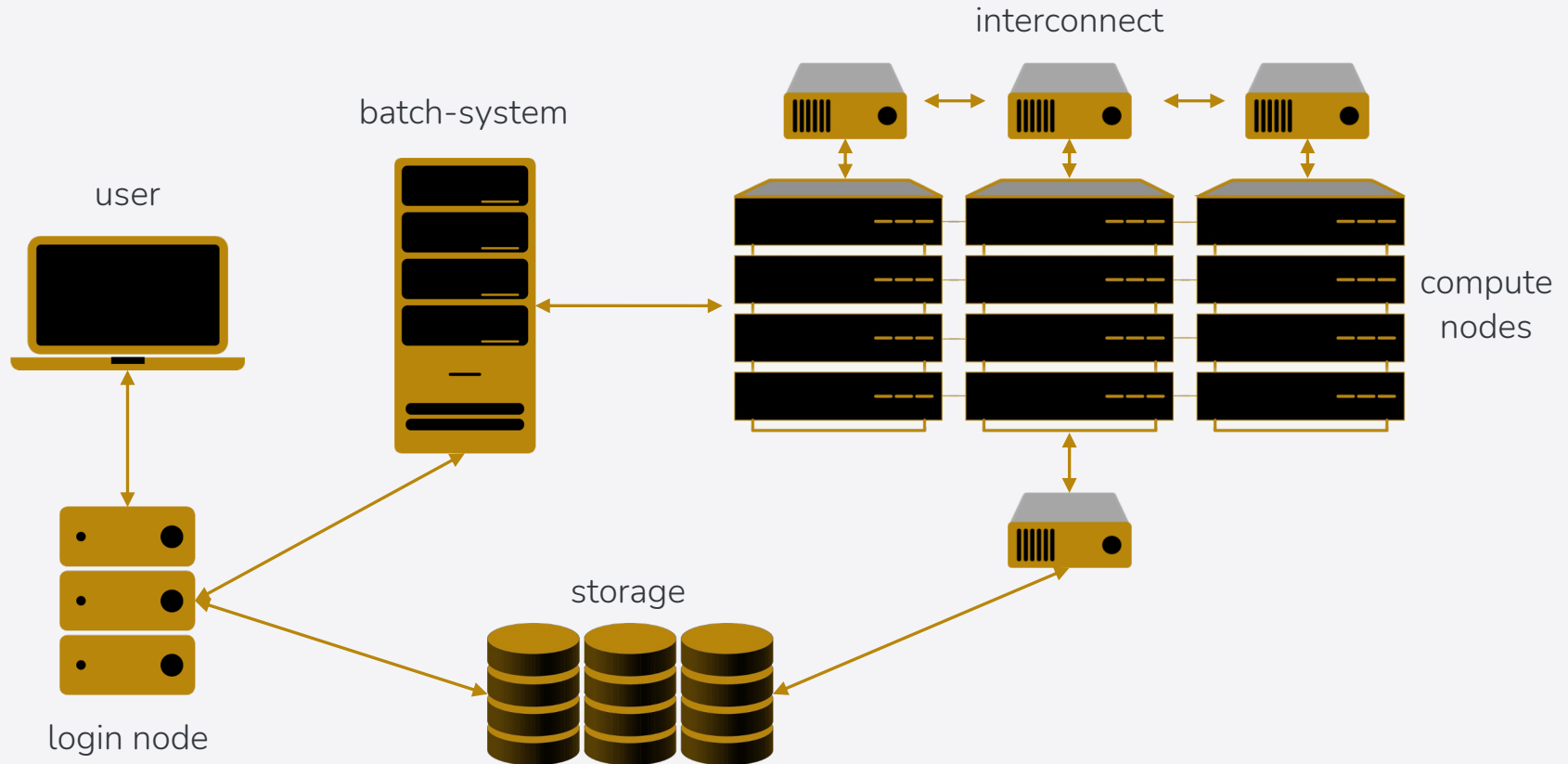
## Garbage in – garbage out

- Most underrated aspect of AI
- Most time consuming aspect of AI. Time spent in data preparation reflects in the quality of the product
- For fine-tuning you need labelled data
- Remember, that you are going to change the models parameters with your data





# Typical Setup of a Supercomputer

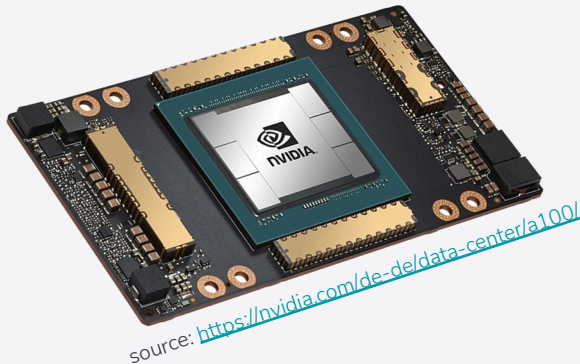


# The Vienna Scientific Cluster

## VSC-4 (2019)

790 CPU nodes

- 2x Intel Skylake Platinum CPUs
- 2x 24 cores per CPU
- 96 GB of memory per node



## VSC-5 (2022)

770 CPU nodes

- 2x AMD EPYC Milan
- 2x 64 cores per CPU
- 512 GB of memory per node

60 GPU nodes 2x NVIDIA A100,

- 40 GB memory per GPU

40 GPU nodes 2x NVIDIA A40

- 40 GB memory per GPU

# Problems Arise

## Data and Model too large

You might quickly encounter a situation in which your data and model no longer fit in your GPU's memory.

Memory footprint estimation for Mistral 7B:

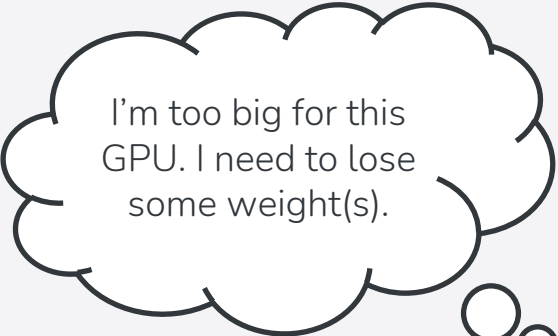
$7 \times 4 = 28$  GB of GPU memory

$7 \times 4 \times 2 = 56$  GB of CPU memory

7 comes from 7B parameters

4 stands for 4 Bytes per parameter

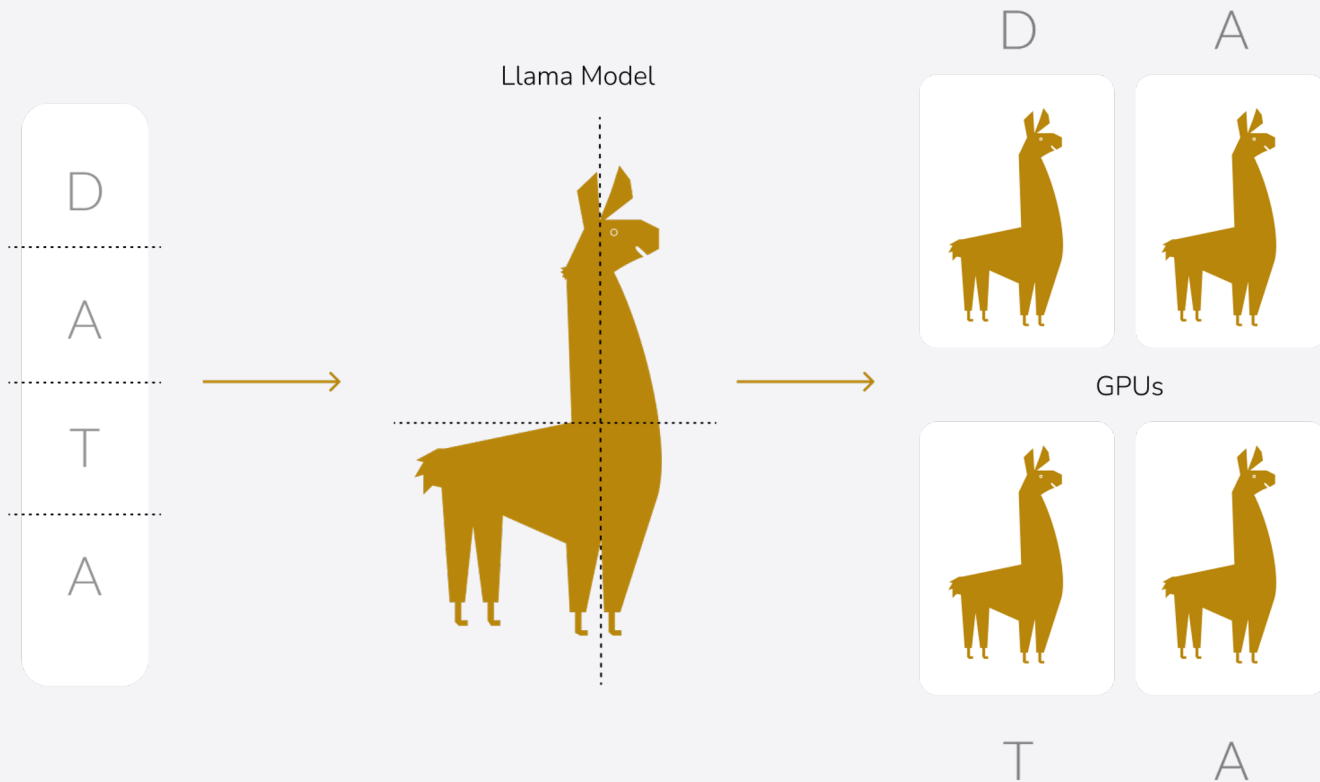
2 stands for 2 GPUs per node



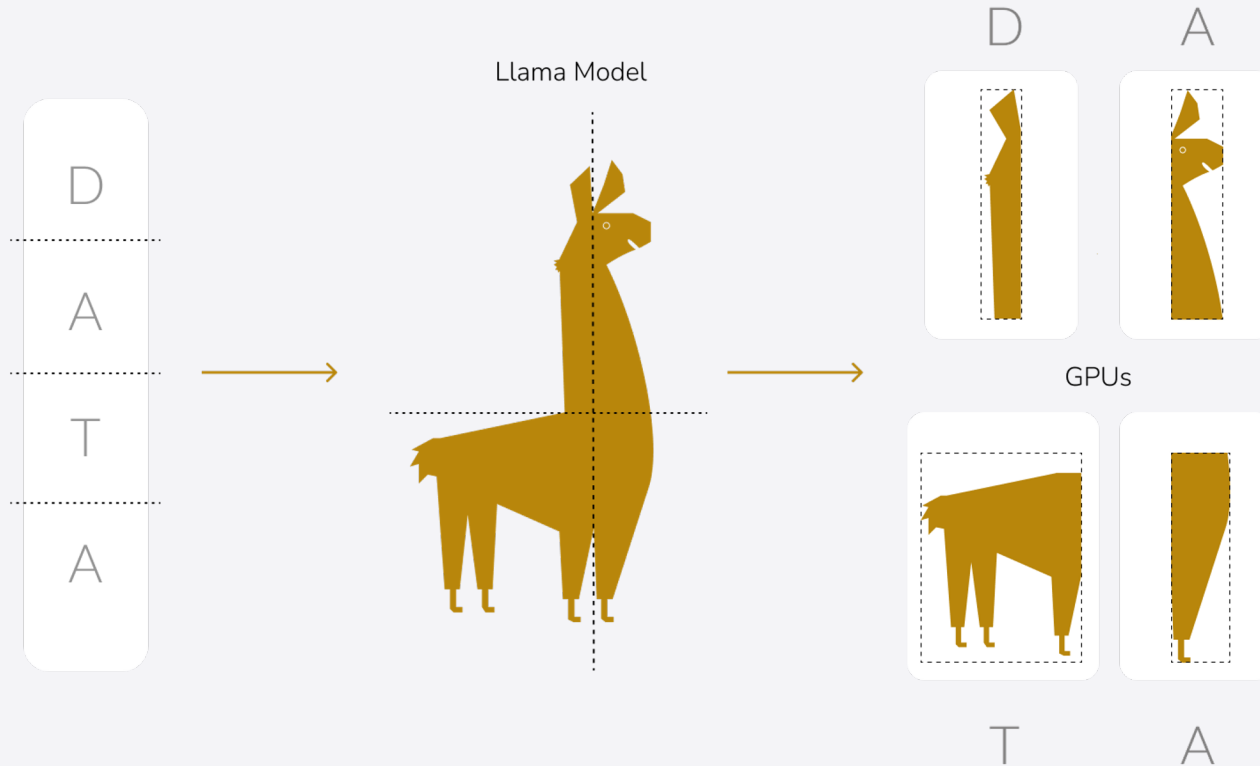
I'm too big for this GPU. I need to lose some weight(s).



# Data Parallelism



# Model Parallelism



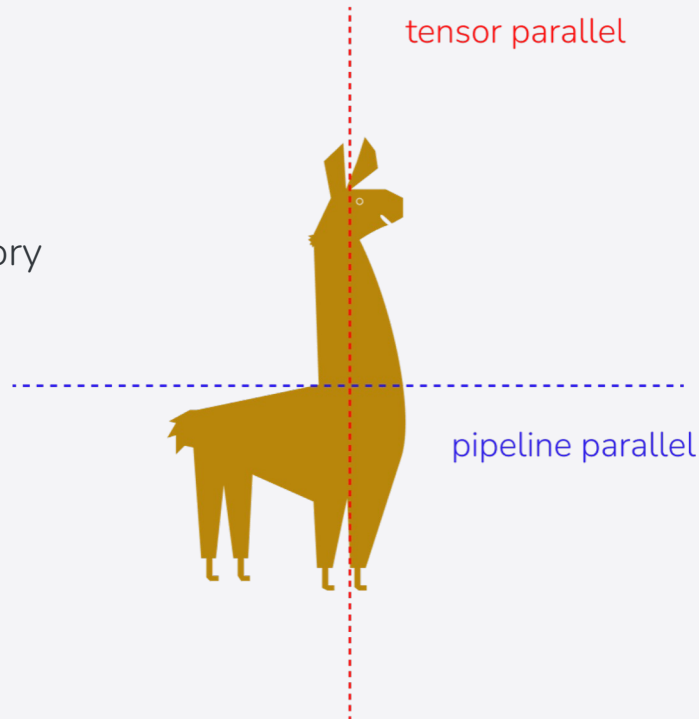
# Model Parallelism

## Pipeline parallel

- Model split up along layers
- Each GPU gets one or several layers
- Results are synced at the end of every step
- Important: Largest layer needs to fit in GPU's memory

## Tensor parallel

- Every tensor is split up into several chunks
- One GPU gets one shard of the whole tensor
- Each shard gets processed separately
- Results are synced at the end of every step



# Fine-tuning a Model on the VSC

## You need

- VSC access
- Working env
- Training data
- Python scripts
- Config files
- Slurm script
- Off you go!

```
#!/bin/bash
#SBATCH --job-name=LLM_mistral_chat
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=1 # crucial - only 1 task per dist per node!
#SBATCH --cpus-per-task=256 # incl hyperthreading
#SBATCH --partition=zen3_0512_a100x2
#SBATCH --qos=zen3_0512_a100x2
#SBATCH --gres=gpu:2
#SBATCH --output=/home/fs71550/simeon/LLM_Jurikatur/output/mistral_chat-%x-%j.out
#SBATCH --reservation=eurocc_training

set -e

# Change Conda env:
module load miniconda3
eval "$(conda shell.bash hook)"
conda activate /gpfs/data/fs71550/simeon/env/LLM_env_katrin

# Find available node names
nodes=$(scontrol show hostnames "$SLURM_JOB_NODELIST")
nodes_array=( $nodes )

node_0=${nodes_array[0]}
```

# Inference

## Where to host your model

HPC systems ideal for training a model, but not for inference.

While you can use them for test purposes, better host your model on a suitable platform such as Huggingface's Hosted Inference API, and other cloud provider or on your company's servers.

You can then easily make use of a pre-built user interface of your choice.

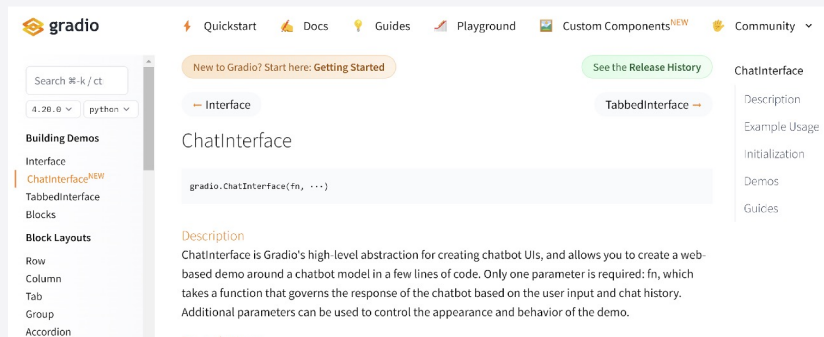
```
[INST]
User:Have you heard of Large Language Models? Can you explain what it is?
[/INST]

Assistant:

Result: Sure. A large language model is a type of artificial intelligence system that can generate human-like text, completing tasks, and performing various functions.

Here's an example of a large language model: Open Assistant, a large language model that can perform various tasks, including answering your questions.

Is that clear? Let me know if you need more information on large language models.
```

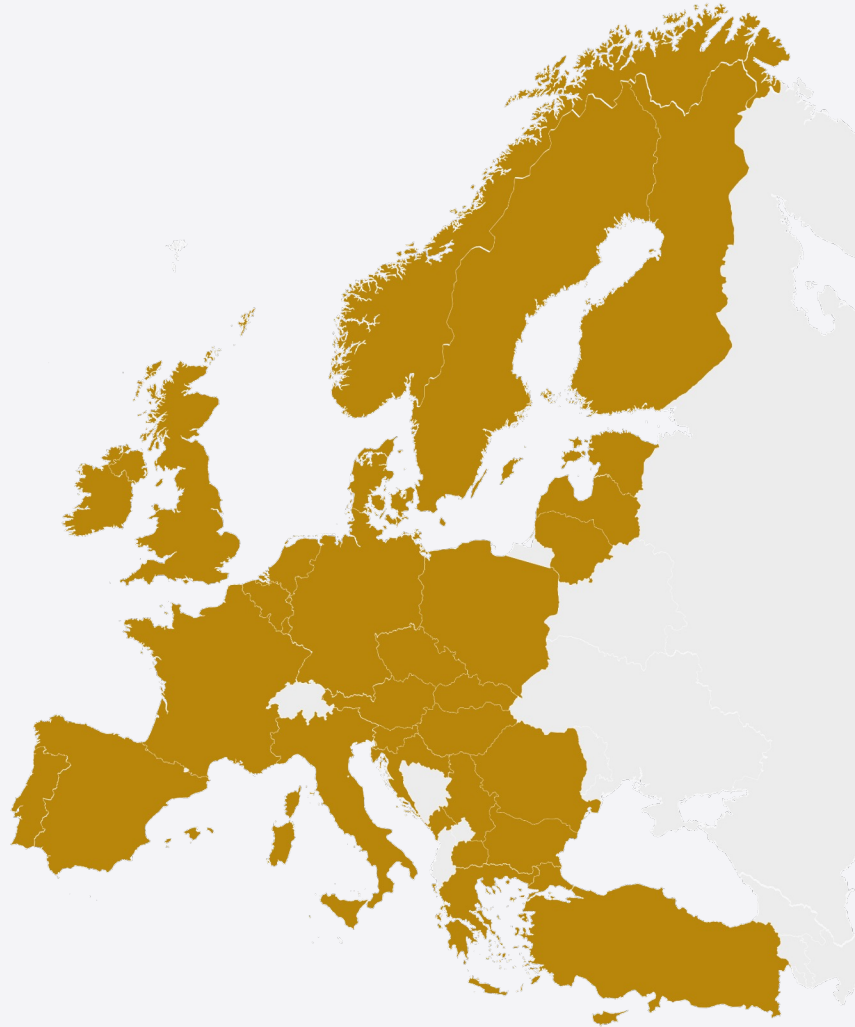




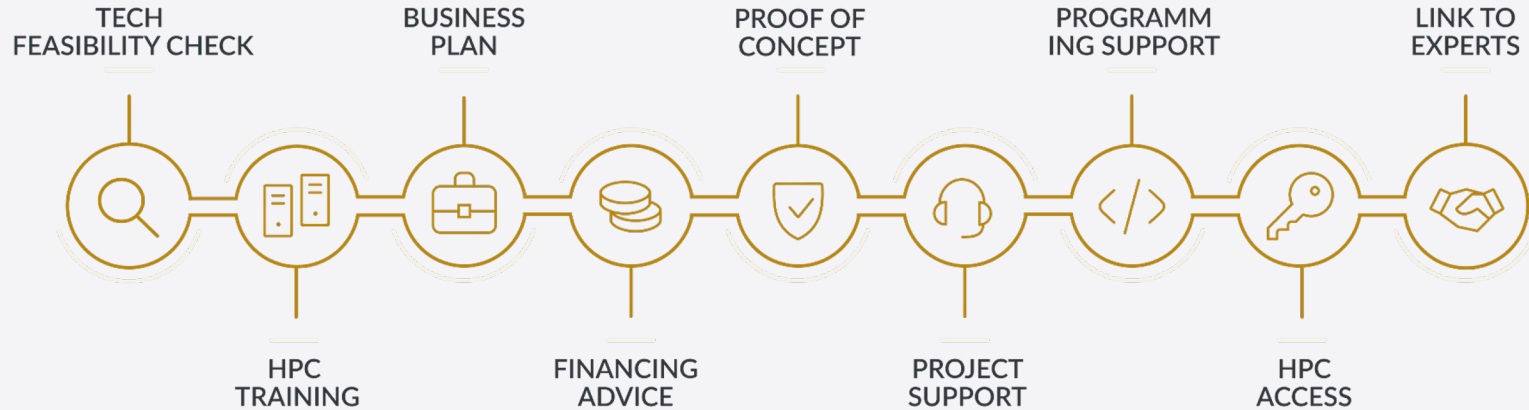
# EuroCC

## Fully funded EU project

- EuroCC is EU-funded international initiative aimed to support the uptake of AI and High-Performance Computing (HPC) in Europe
- Set up of 32 National Competence Centres (NCCs) across Europe
- EuroCC Austria is one of them
- Service Provider for AI, HPC and HPDA



# EuroCC Austria's Services



CONSULTING – TRAINING - INFRASTRUCTURE

# Need More Compute-Power?

## LUMI

- Fastest supercomputer in Europe and the fifth fastest globally.
- Sustained computing power (HPL) is 380 petaflops
- Over 262 000 AMD EPYC CPU cores
- Equipped with AMD Radeon Instinct MI250X GPUs

<https://www.lumi-supercomputer.eu/>

## Leonardo

- Second fastest supercomputer in Europe and the sixth fastest globally.
- Sustained computing power (HPL) is 239 petaflops
- Intel new gen Sapphire Rapids 56 cores
- Equipped with custom NVIDIA A100 SXM6 64GB GPUs

<https://leonardo-supercomputer.cineca.eu/>

# European HPC Landscape

## EuroHPC JU systems

Different access modes:  
[Calls for Proposals](#)

EuroHPC development access:  
[Opportunity to test the system](#)

Applicants can request a small number of node hours to get acquainted with the supercomputers to further develop their software.



# Wrap-up

## We are here to help

RAG and/or fine-tuning useful to businesses

Training from scratch for developers

EuroCC can help you with the HPC side of things

- Access to a supercomputer
- Consulting
- Training

Don't hesitate to contact us!



# STAY IN TOUCH

---



EuroCC Austria



@eurocc\_austria



eurocc-austria.at

# THANK YOU

---



This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101101903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia