

SUPERCOMPUTERS AND GENAI FOR SMES

Rasmus Larsen

Senior AI Specialist

Dennis Lange Wollbrink

Projektleder, EuroCC2



ALEXANDRA
INSTITUTE



Agenda

1. Background

- Training steps for GenAI models, the “GenAI stack” (data -> base models -> ... -> agent frameworks)
- Why (not) to train?

2. Ecosystem & finding your problem



Step 1: Pre-training

Pile of Data

+

Huge amount of GPUs

(you will probably not do this)

Step 2: Post-training

Technique

Fine-Tuning

- Supervised Fine-Tuning
- Adaptive Fine-Tuning
- Reinforcement Fine-Tuning

Reasoning

- Self-Refine
- RL for Reasoning

Integration and Adaptation

- Multi-modal integration
- Domain Adaptation
- Model Merging

Alignment

- RL with Human Feedback
- RL with AI Feedback
- Direct Preference Optimization

Efficiency

- Model Compression
- Parameter-Efficient Fine-Tuning
- Knowledge Distillation

Data

Dialogue



Reasoning



Multilingual



Code



Question-Answering



Text Generation



Instruction Following



Application

Professional Domain

Legal Assistant



Healthcare and Medical

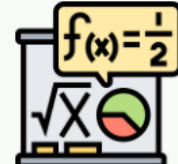


Finance and Economics



Technical and Logical Reasoning

Mathematical Reasoning



Code Generation



Understanding and Interaction

Recommendation System



Speech Conversation



Video Understanding

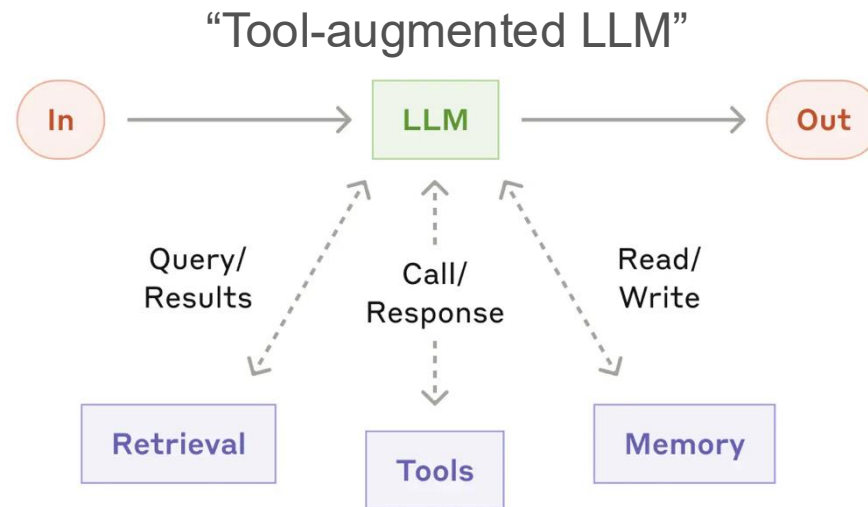


(Tie et al., 2025)

Agents

"[AI research is] the study and design of rational agents."

- Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach* (1995)





Components for training

- Data
- Data! – and not just a “database”
- Compute
- Talent

Why not to train a model

We have compute available

↳ *That's a resource, not a goal*

Everyone else is doing it

↳ *That's peer pressure, not strategy*

AI is the future

↳ *That's a platitude, not a plan*

We want the best model possible

↳ *That's not specific enough to guide decisions*



... so why train a model?

Research

Ecosystem

Business - if...

- large scale (100Ks req/month)
- legal aspects require it
- you can serve the model after training!
- you did the math – GPT-5: 1M requests x 10K input x 1K output = \$22000

Existing models to build on

Chinese

Kimi K2, MiniMax M2, Ling, Meituan, Qwen, (too many to count)

American

OpenAI GPT-OSS, Google Gemma, HF SmolLM, Arcee

European

Mistral, Apertus

Existing software



HF ecosystem – transformers, datasets, trl, ...



PyTorch ecosystem – Torchtitan, Torchforge, ...



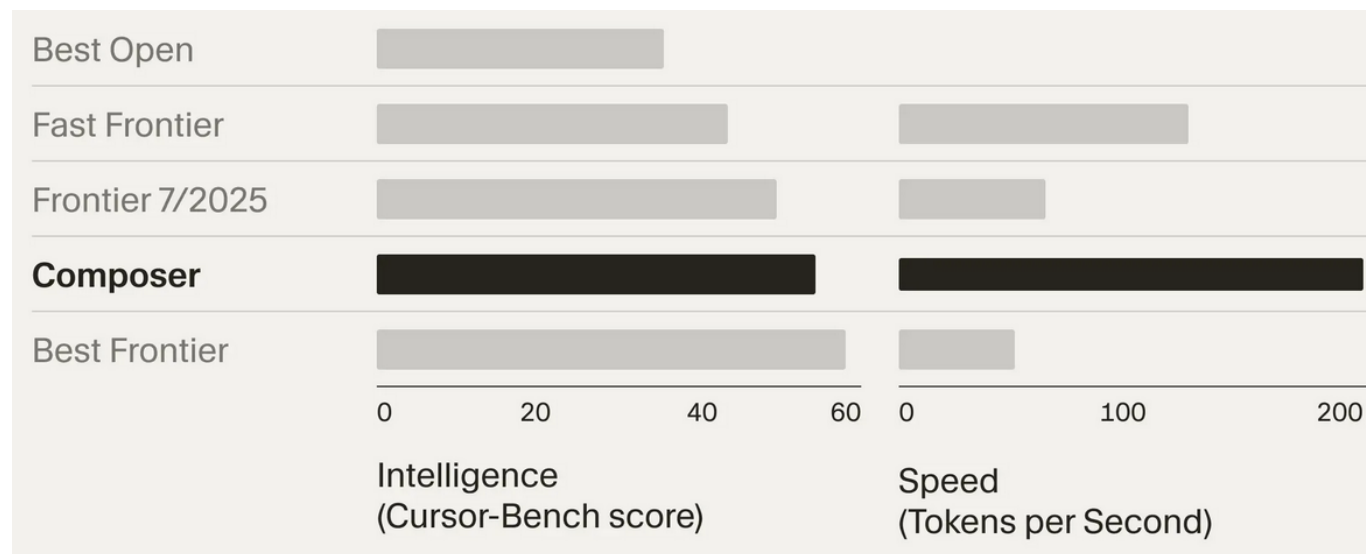
Various others – verifiers, atropos, PipelineRL, ...

Finding your problem

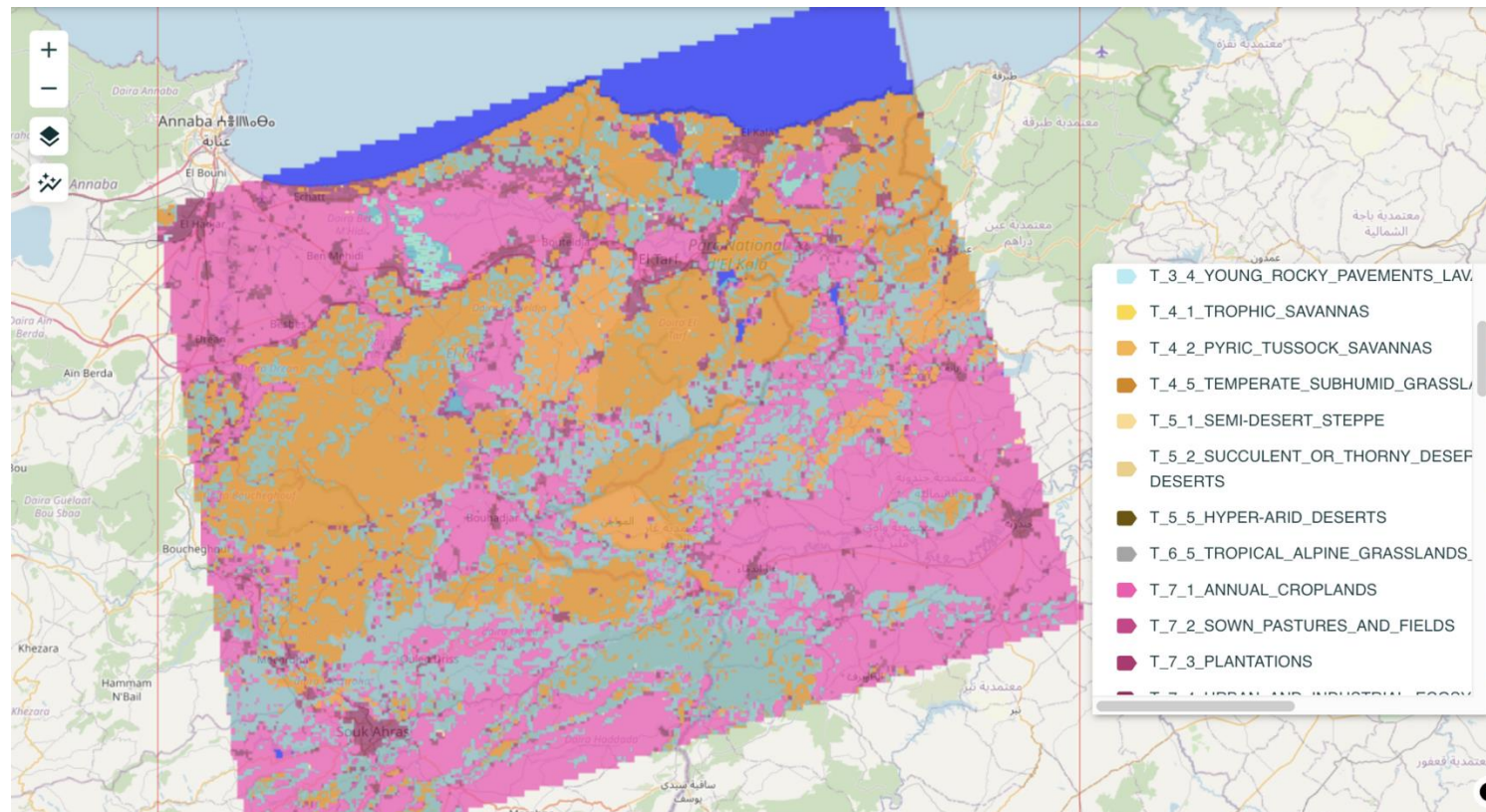
- Specializing a strong, open model for a vertical is valuable
- I would perhaps look for things that are...
 - text/voice/image heavy – think emails, contracts, transcripts, objects
 - variable, but with structure – not solved by simpler methods, not too hard
 - bottlenecked by reading and writing
 - long-tail knowledge
 - high volume and low stakes

Example: Cursor 2.0 / Composer

- Take open source (Chinese) model
- Post-train (RL) on custom data for coding <-- EuroHPC!
- Flywheel: get more, better data -> train better model
- Build more evals to understand performance



Example: OlmoEarth



“applying a foundation model to a novel task requires data gathering, alignment, pre-processing, labeling, fine-tuning, and running inference”

Rasmus Larsen · rasmus.larsen@alexandra.dk

