

SMARTHEP

REAL-TIME ANALYSIS FOR
SCIENCE AND INDUSTRY

TARP: Twikis for ATLAS via RAG Protocols

Overview:

- Fine-tuning
- Multi-modal LLMs
- Tool-suite
- AI dev tools
- Outlook

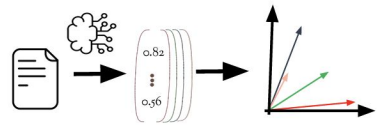


SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

RAG vs LLM Fine-tuning

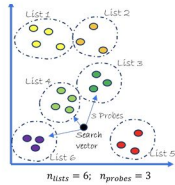
RAG Workflow

01 Embeddings



Transformer model converts paragraphs into vectors encoding semantic meaning.

02 Retrieval



A Cosine Similarity Search via Approximate Nearest Neighbors finding the closest matching vectors.

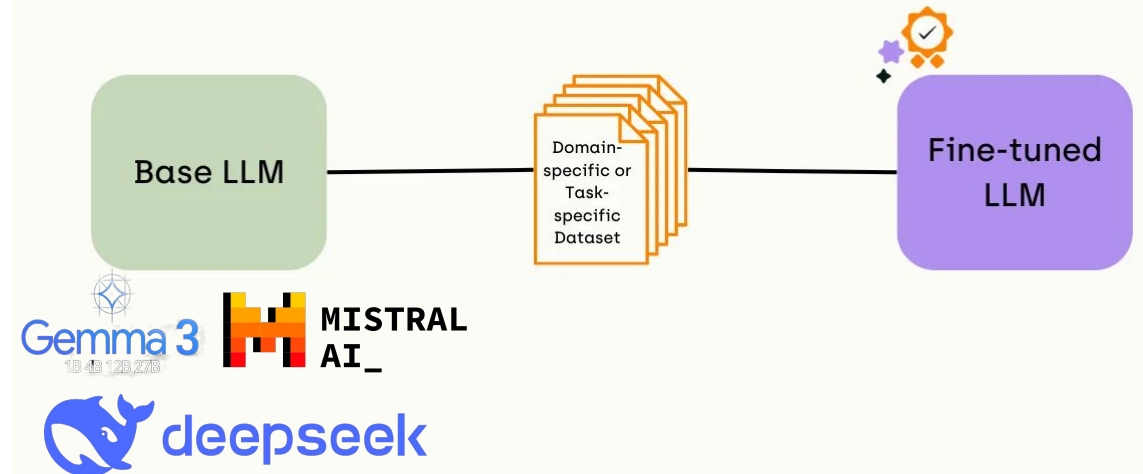
03 Generation



LLM uses retrieved documents to answer the question asked

[Source](#)

Fine-tuning an LLM

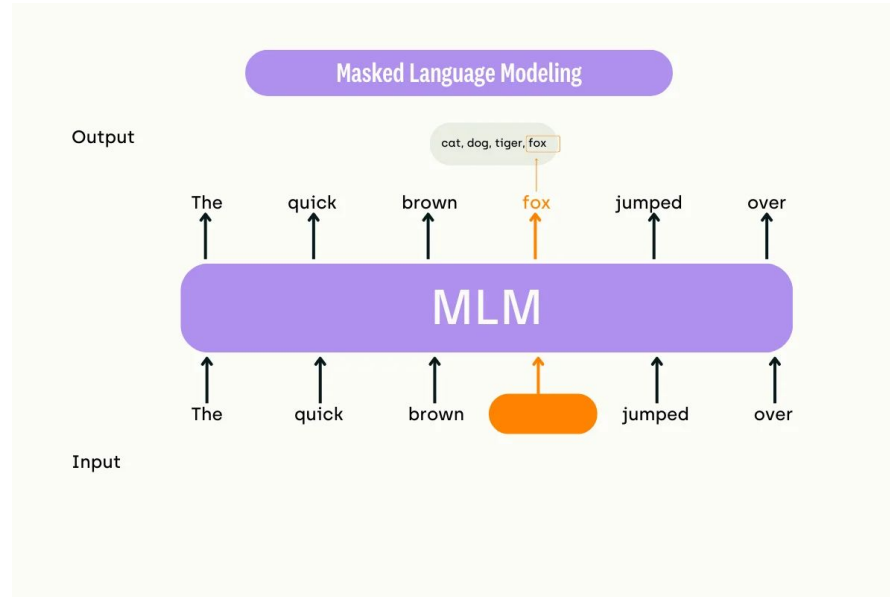


[Source](#)

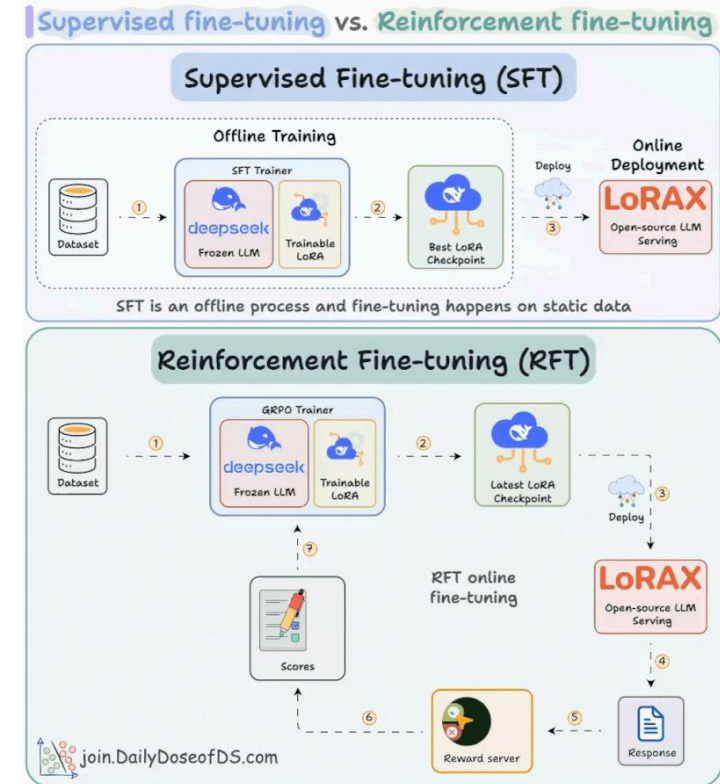


How do you fine-tune?

- Unsupervised:
 - Masked Language Models
- Supervised:
 - Generate Labeled Dataset
- Reinforcement:
 - Setup a feedback loop with reward chains



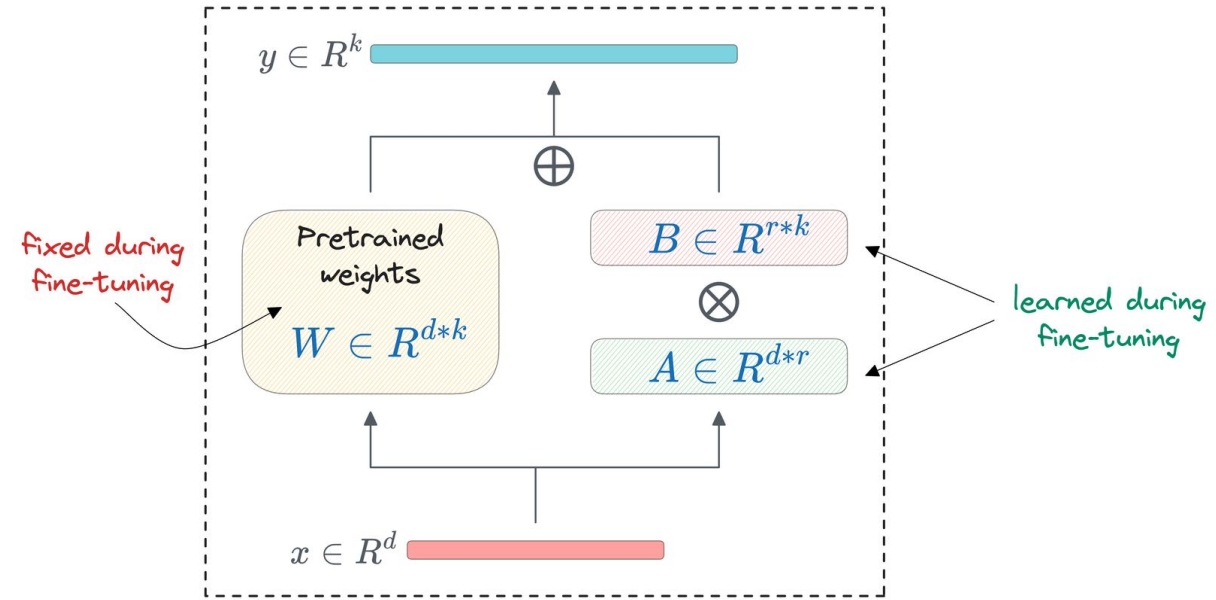
[Source](#)



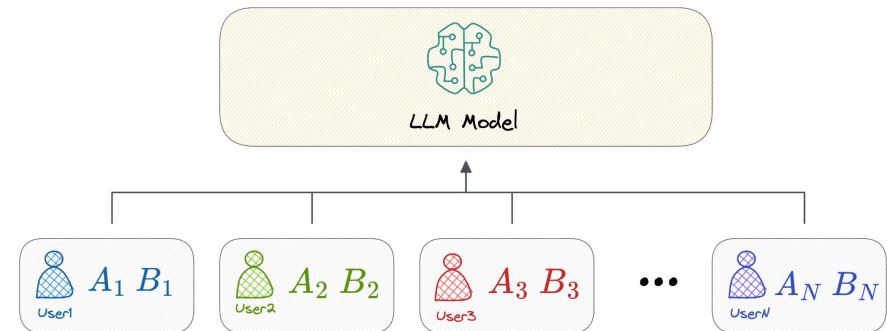
[Source](#)

What do you fine-tune?

- Fine-tune the whole base model
 - Backprop through all weights
 - “Small” models have $O(1B)$ params - Too expensive!
- **Low Rank Adaptation (LoRA)**
 - Fix the massive base model
 - Create small low-rank adapters
 - Append adapter to base and train on target dataset (say SM analyses)
 - Create new adapter for new target (say Exotics analyses) ...
 - Possibly 100s of adapters each tuned on a different concept/group

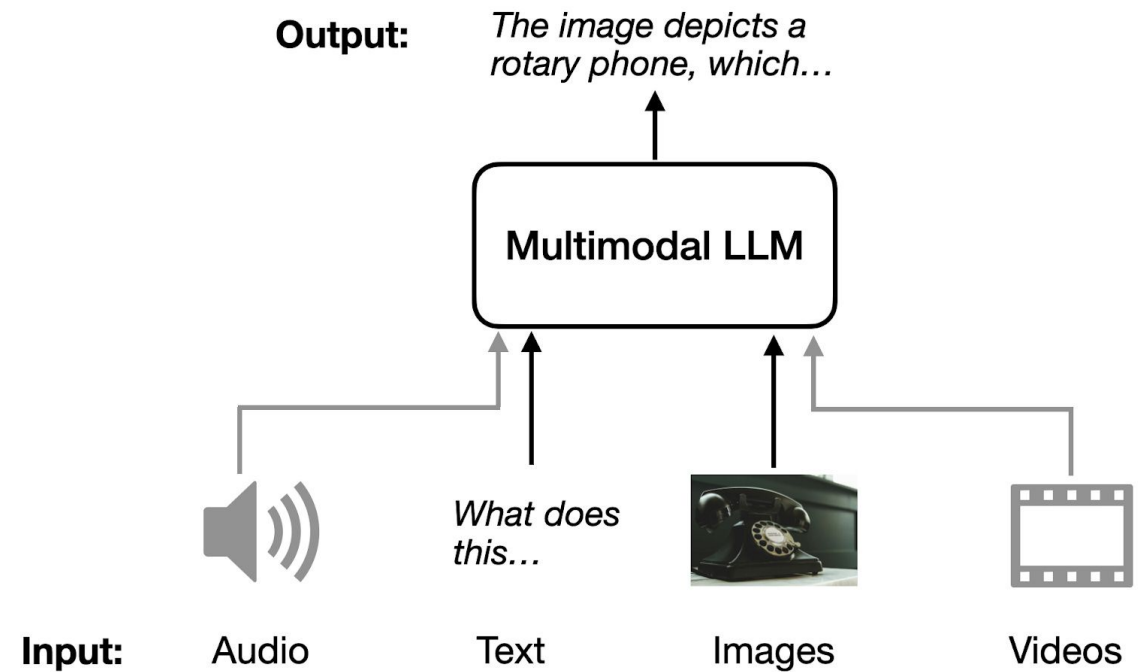


[Source](#)



Multi-modal Fine-tuning/RAG

- Research papers usually carry equal weighted importance for:
 - Text
 - Graphicals
- Just text generation/learning may not be enough
- Multi-modal LLMs (Multiple input modalities)
 - TARP LLMs take both text and images as inputs during fine-tuning:
 - Generate text outputs
 - eg: Gemma 3, Deepseek V3
 - Generate text + image outputs
 - Deepseek Janus pro



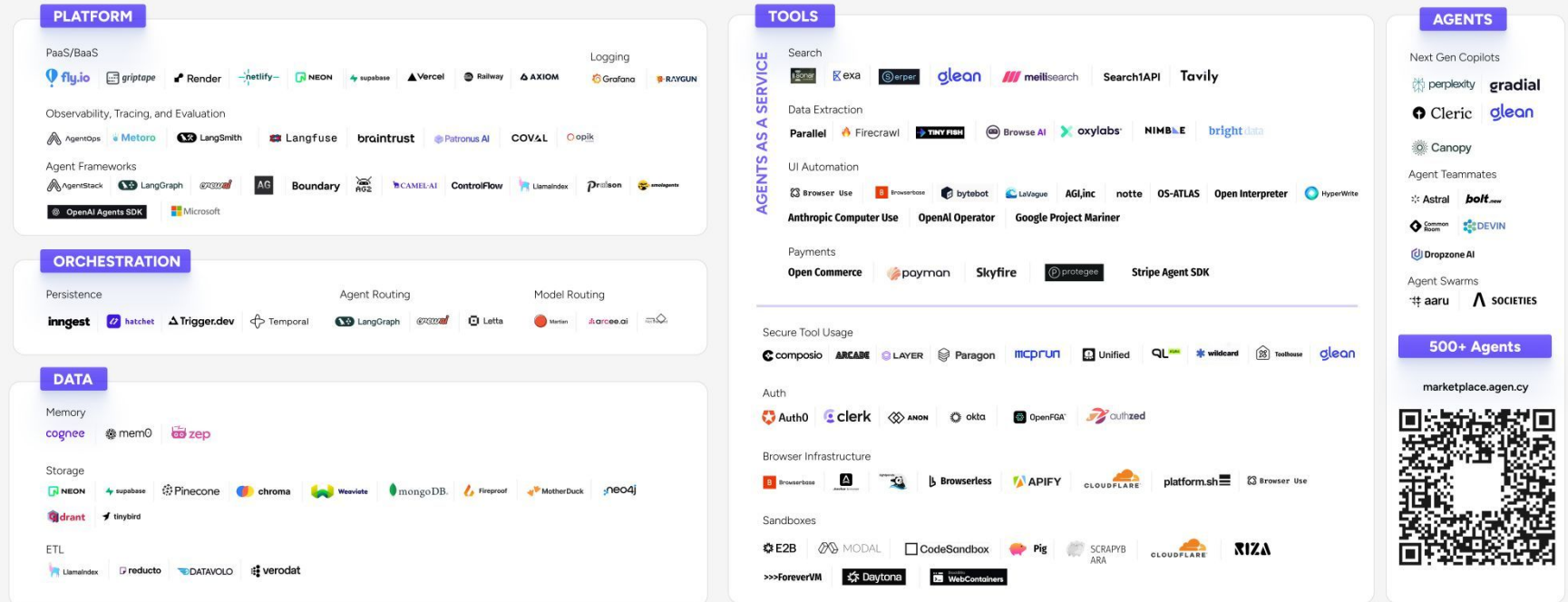
[Source](#)

LLM Agent Tool-Suites

- Status quo:
 - Crowded
 - Dynamic
 - Distributed
- Optimal strat?
 - Patience

2025 AI Agents Infrastructure Stack

Powered by
Agency AI + AgentOps



The infographic illustrates the 2025 AI Agents Infrastructure Stack, organized into five main categories:

- PLATFORM:** Includes PaaS/BaaS (fly.io, griptape, Render, Netlify, NEON, Supabase, Vercel, Railway, AXIOM, Grafana, RYGLN), Observability, Tracing, and Evaluation (AgentOps, Metoro, LangSmith, Langfuse, braintrust, Patronus AI, COVLL, opik), Agent Frameworks (AgentStack, LangGraph, AG, Boundary, CAMEL-AI, ControlFlow, Llamaindex, Pralison, OpenAI Agents SDK, Microsoft).
- ORCHESTRATION:** Includes Persistence (Inngest, hatched, Trigger.dev, Temporal, LangGraph, Letta, Merlin, arcee.ai) and Agent Routing (LangGraph, Letta, Merlin, arcee.ai).
- DATA:** Includes Memory (cognee, mem0, zep), Storage (NEON, Supabase, Pinecone, Chroma, Weaviate, MongoDB, Fireproof, MotherDuck, Neo4j, Drant, Tinybird), and ETL (Llamaindex, Reducto, DATAVOLO, Verodat).
- TOOLS:** Includes Search (Elastic, Exa, Perplexity, Glean, Meltsearch, Search1API, Tavily), Data Extraction (Parallel, Firecrawl, TinyPill, Browse AI, OxyLabs, Nimble, BrightData), UI Automation (Browser Use, Browserbase, Bytebot, LaVague, AGI, Inc, Notte, OS-ATLAS, Open Interpreter, HyperWrite), Anthropic Computer Use, OpenAI Operator, Google Project Mariner, Payments (Open Commerce, Payman, Skyfire, Protegee, Stripe Agent SDK), Secure Tool Usage (Composio, Arc4me, Layer, Paragon, Micpfun, Unified, QL, Wildcard, Youbase, Glean), Auth (Auth0, Clerk, Anon, Okta, OpenFGA, Outfrazed), Browser Infrastructure (Browserbase, Browserless, APIFY, Cloudflare, Platform.sh, Browser Use), Sandboxes (E2B, Modal, CodeSandbox, Pig, ScrapyB ARA, Cloudflare, Riza, ForeverVM, Daytona, WebContainers).
- AGENTS:** Includes Next Gen Copilots (Perplexity, Gradial, Cleric, Glean), Canopy, Agent Teammates (Astral, Bolt, Common Room, Devin), Agent Swarms (Dropzone AI, Aaru, Societies), and a marketplace for 500+ Agents (marketplace.agen.cy).

Powered by
Agency AI + AgentOps

Source
substack.com/@jonturow & marketplace.agen.cy



TARP Tool-Suite

- Datascraping:
 - CDS text:
 - Crawl4AI
 - Firecrawl
 - CDS plots:
 - ColiVara
- Orchestration:
 - LlamaIndex
- LLM serving:
 - Ollama
- Hyper-fast fine-tuning:
 - Sloth
- Tracing and Evals:
 - Comet Opik
- Potential TARP serving:
 - LoRAX
- Rough UI:
 - Streamlit



Firecrawl

tjmlabs/**ColiVara**

Colivara is a suite of services that allows you to store, search, and retrieve documents based on their visual embedding....



LlamaIndex



Ollama



unslot

comet-ml/**opik**

Debug, evaluate, and monitor your LLM applications, RAG systems, and agentic workflows with comprehensive tracing, automated evaluations, and production-ready dashboards.



LoRAX

Multi-LoRA inference server that scales to 1000s of fine-tuned LLMs



Streamlit

My Takeaways:

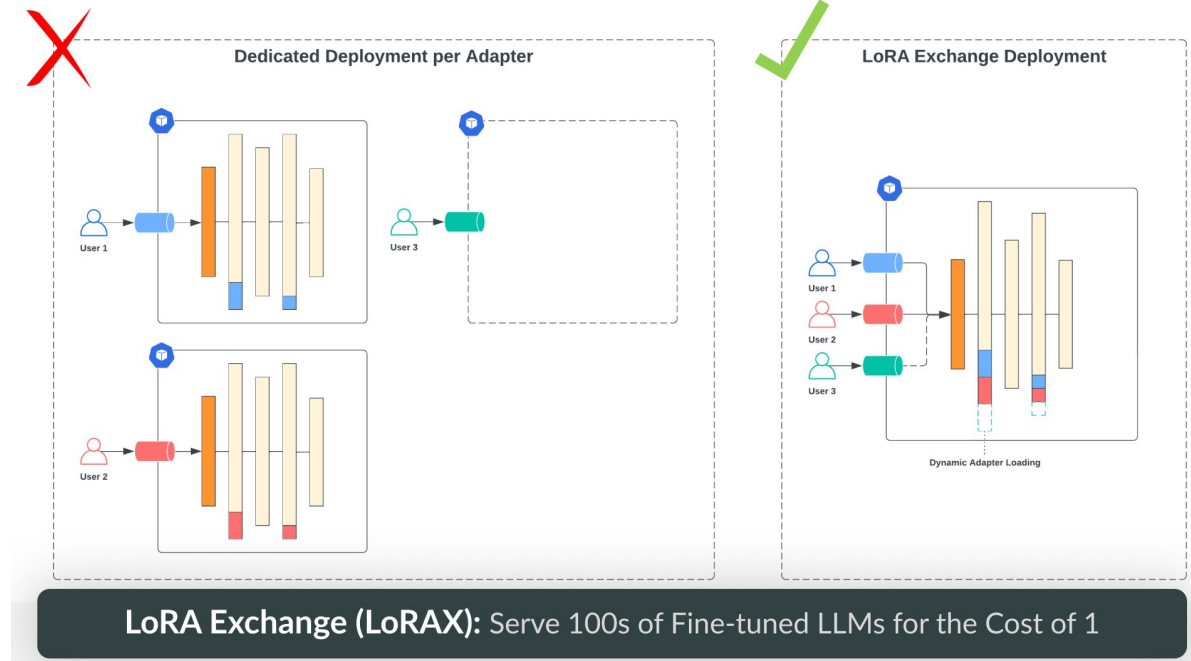
- Can HEP benefit from AI tools?
 - Yes
- Are there any good off-the-shelf options currently?
 - No (for most, specific use cases)
- Developing our own AI tools?
 - Yes, but..
 - Mimicking existing tools on top of general purpose LLMs doesn't work in short term
 - Way too many new tools/protocols coming out frequently
 - Risk locking in to a suboptimal setup
 - Mimicking existing text-gen, image-gen and code-gen
 - Should(?) work in the long run
 - Constant iteration required
- So what do we do now?
 - Build our own Base models
 - Foundation models for Physics
 - LLMs from scratch tailored to Sciences
 - Asking Shakespeare to sing Feynman is not the way to go!

Backup



SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

- Serving 100s of fine-tuned adapters on a single GPU
 - Dynamic adapter selection



[Source](#)

AI Dev Tools used in TARP

- Analyze tool-suite repos to make choices:
 - Gitingest
 - Gitdiagram
- Deep-research + Plan generation:
 - Gemini 2.5 Pro Beta
 - Perplexity
- Agentic Planning + Skeletons:
 - Manus AI
- Code Gen:
 - Zed AI
 - Gemini 2.5 Pro
 - Claude 3.5 Sonnet



Instruction Fine-tuning

- How do you generate labeled datasets from HEP text datasets?
 - Maybe, like this 