ORACLE

Building the Brain and Backbone of Enterprise AI Agents

Advanced Reasoning & Infrastructure Strategies

Nacho Martinez

jasperan

AI enjoyer

Edit profile

Sponsors dashboard

326 followers · 128 following

@oracle

28:05:FF:58:31:05

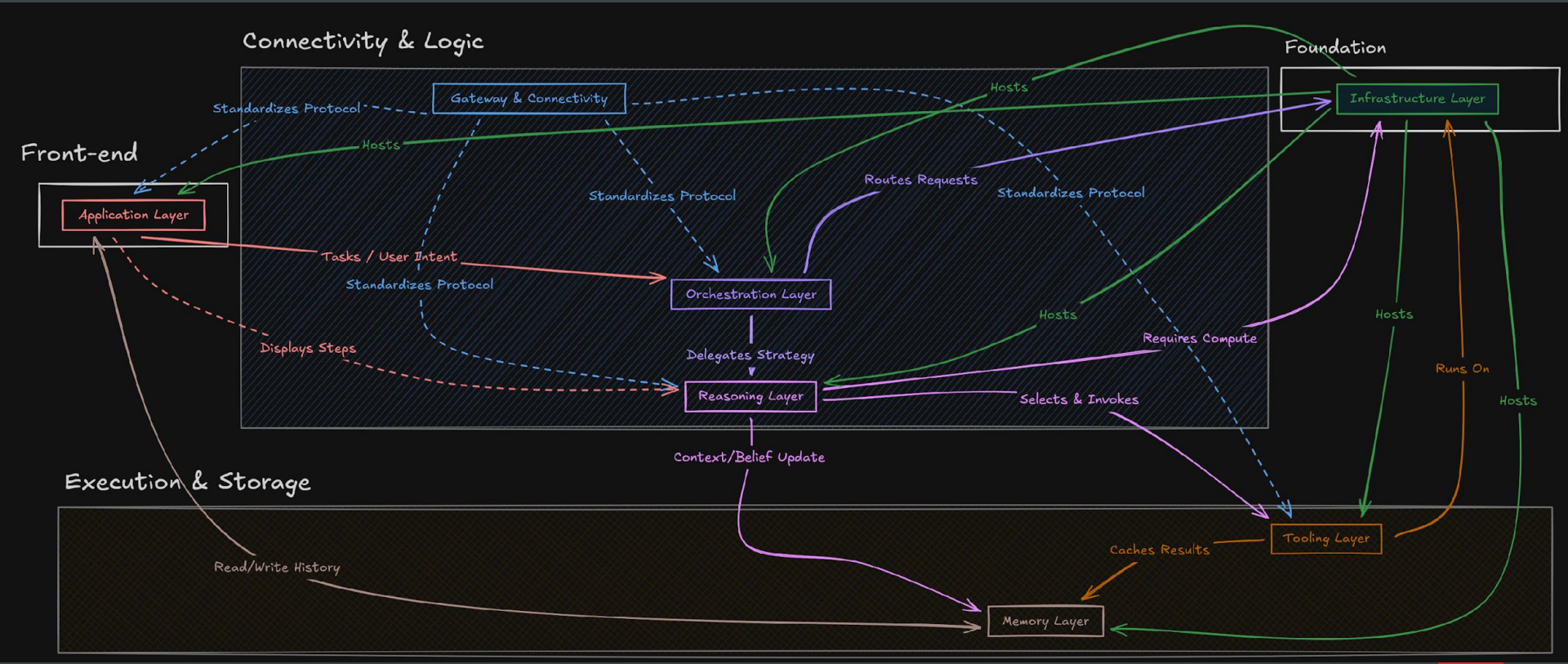
in/jasperan

Achievements

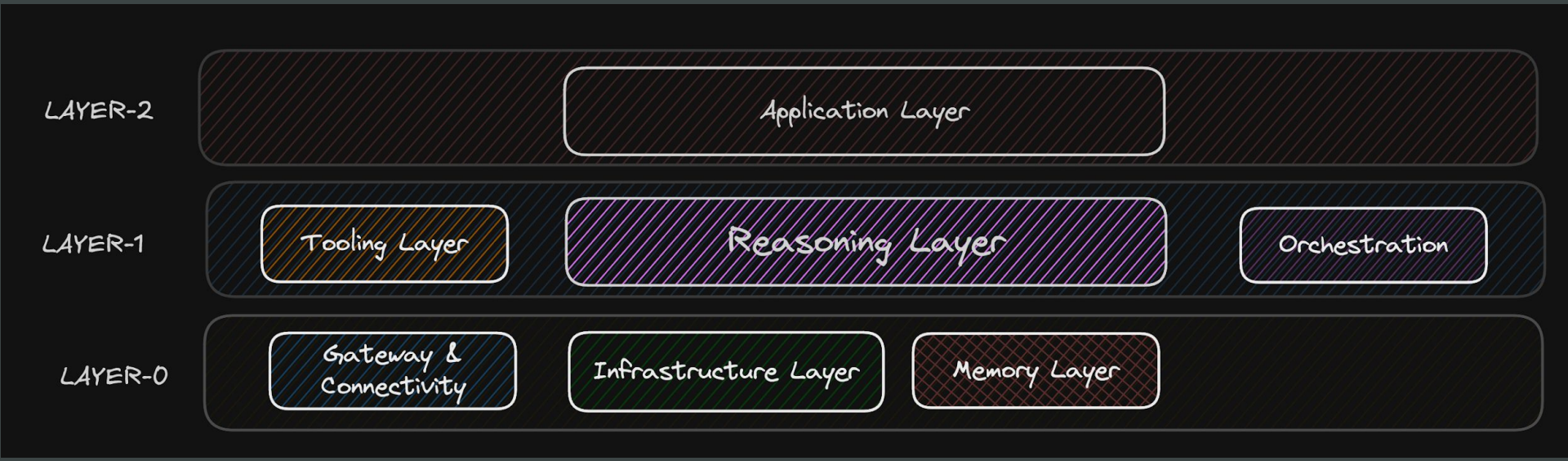
Data Scientist Advocate

Developer Relations @ Oracle

The Agent Stack in 2026



The Agent Stack in 2026



Layer-0

All about infrastructure, networking & databases: storage (***memory***), routing (***gateway*** & ***connectivity***) and raw compute (***infrastructure***)

Layer-1

Reasoning & ***Tooling*** Layer

Layer-2

Front-end: ***application*** layer

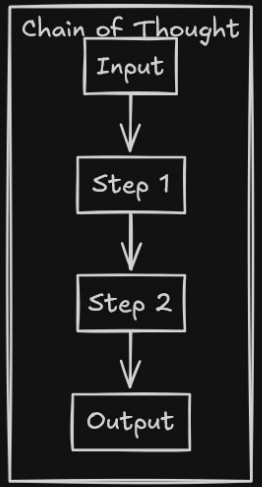
It's not about standard LLM prompts anymore

Context & Belief updates

meaningful

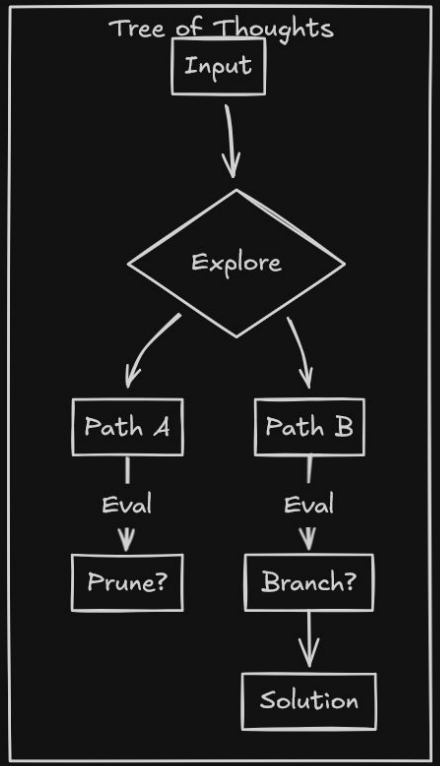
MULTIPLE TRAJECTORIES IN PARALLEL

OPTIMAL FOR REASONING, INTENSIVE TASKS



+58.3% (symbolic reasoning)

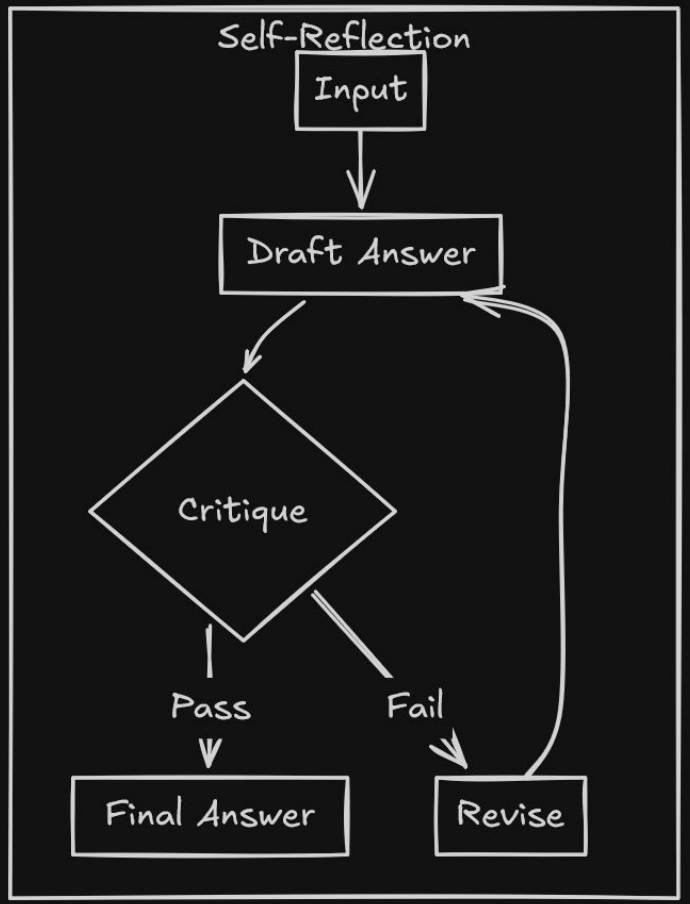
+42.1% (math)



GPT-4: 4% acc

+1850%!

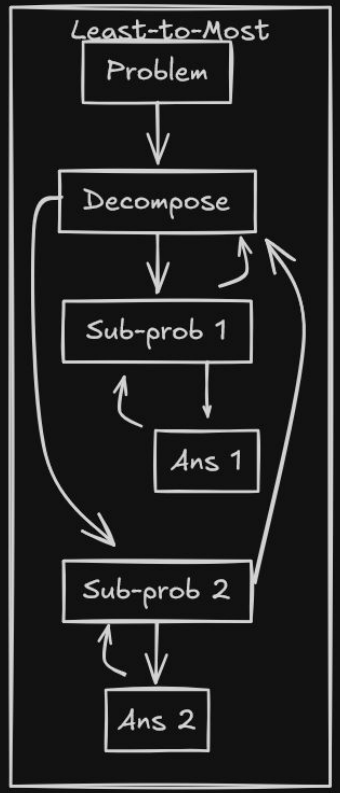
ToT: 74%



33%->76% (analytical reasoning)

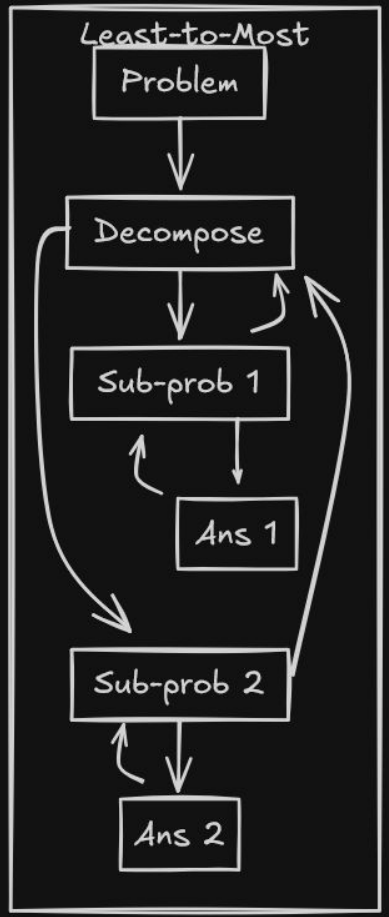
93%->98% (SAT-english)

OPTIMAL FOR HARD REASONING, PROBLEMS

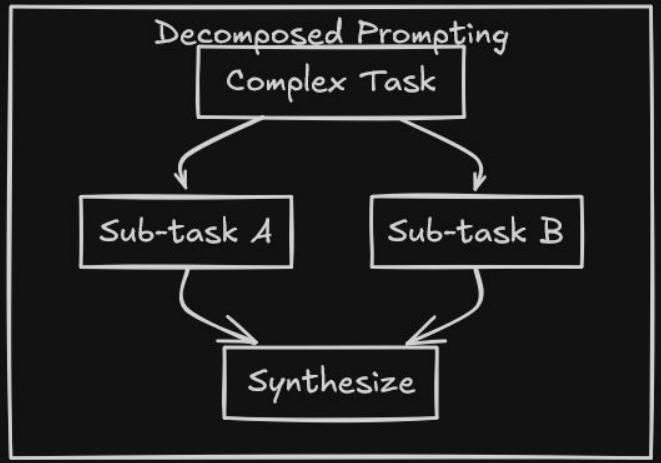


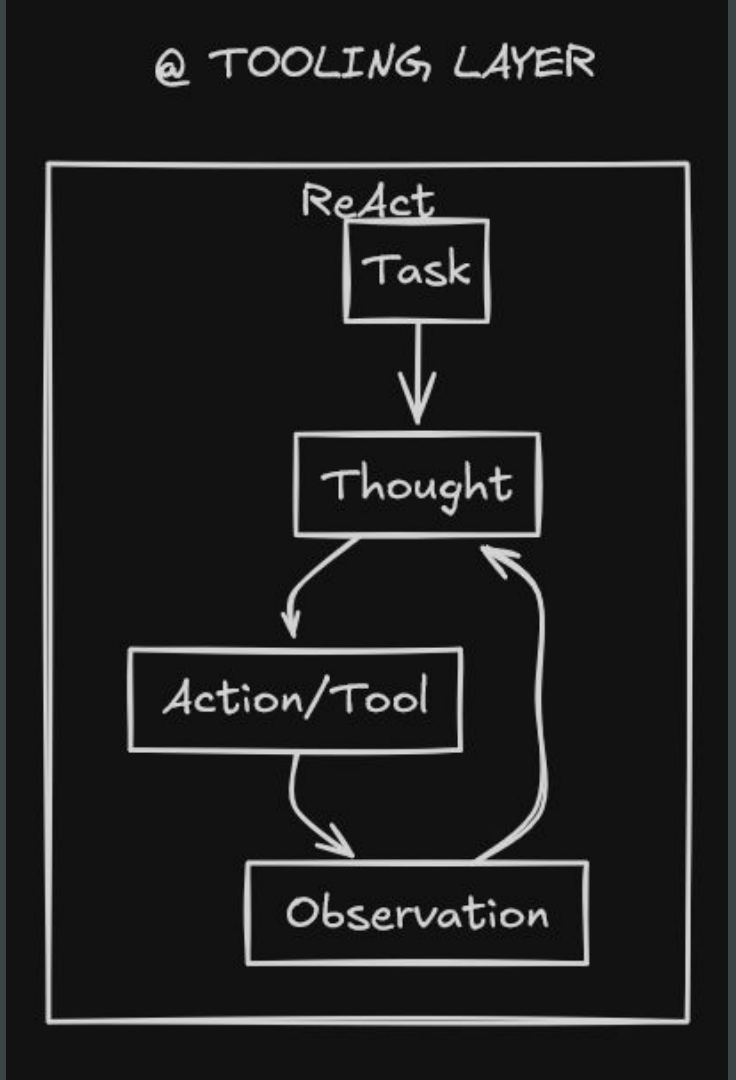
(symbolic reasoning)
(math)

OPTIMAL FOR HARD REASONING PROBLEMS



OPTIMAL FOR LONG LISTS OF INSTRUCTIONS



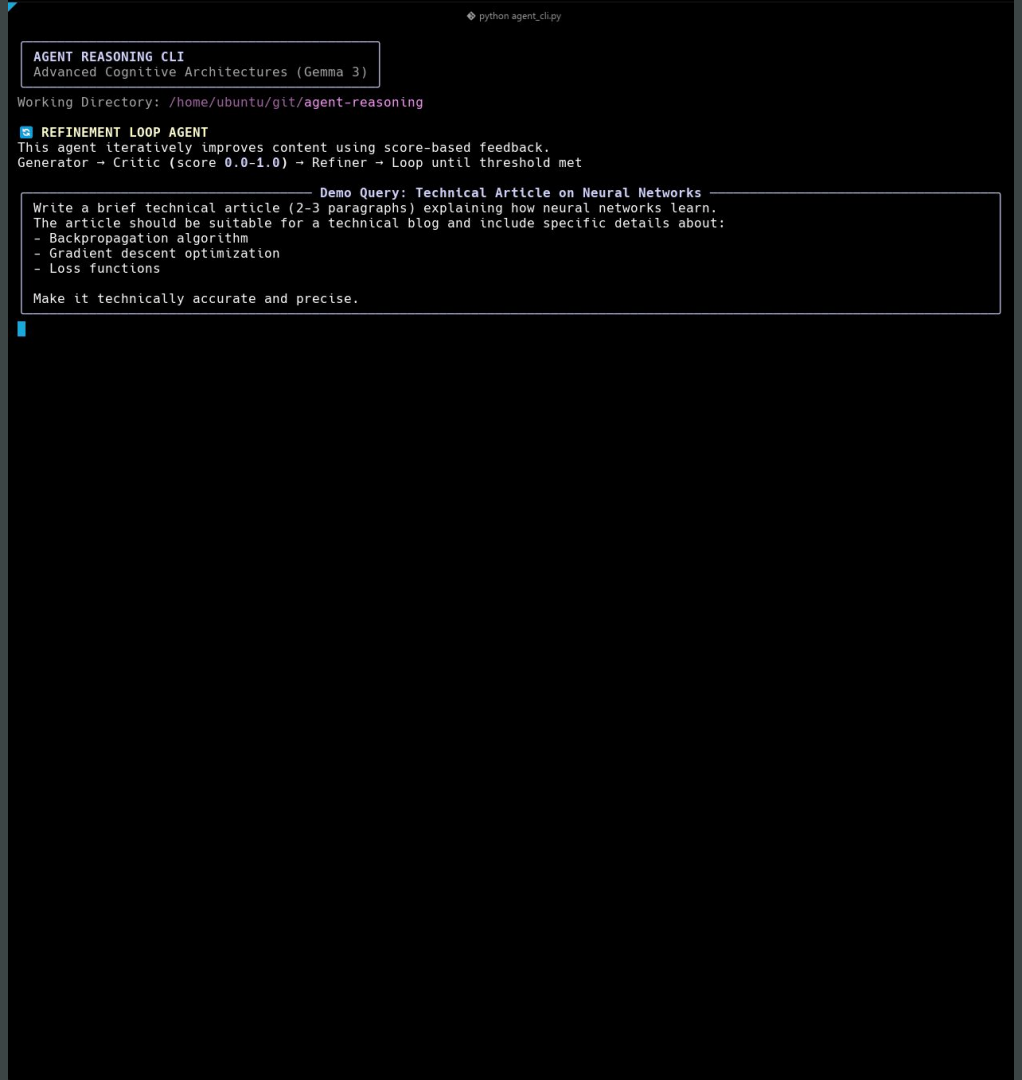


Refinement loops

Automating LLM interactions

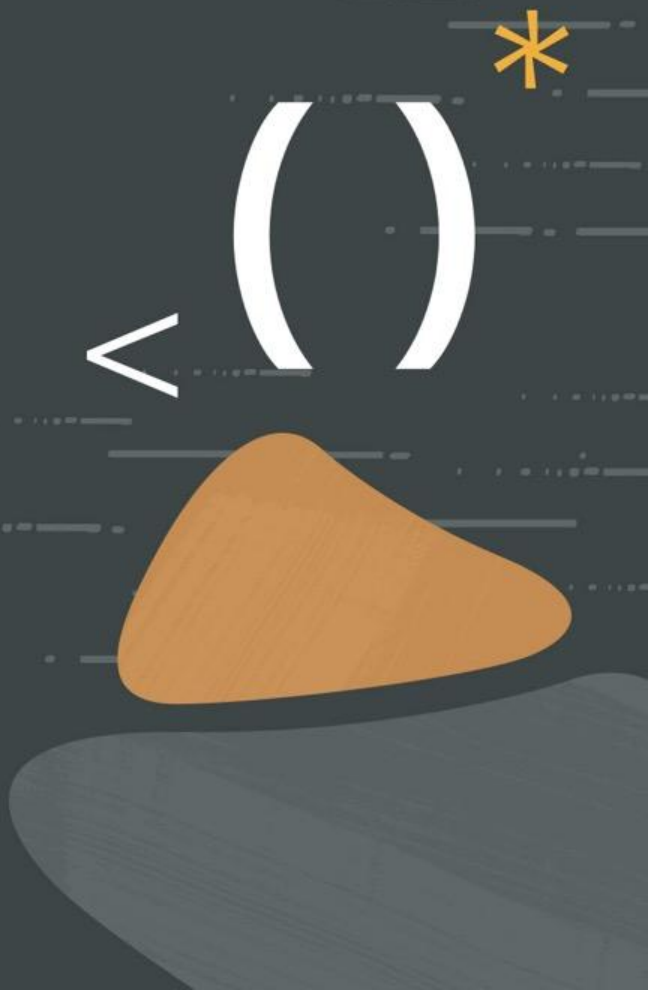
```python
def refinement_loop(query: str, threshold: float = 0.9, max_iterations: int = 5) -> str:
    generator = Generator()
    critic = Critic()
    refiner = Refiner()

    # 1. Initial Draft
    draft = generator.invoke(query)

    # 2. Initial Critique
    critique = critic.invoke(draft)

    iteration = 0
    while critique.score < threshold and iteration < max_iterations:
        iteration += 1

        # 3. Refine
        draft = refiner.invoke(draft, critique.feedback)

        # 4. Critique again
        critique = critic.invoke(draft)

    return draft
```

Source: *github.com/jasperan/agent-reasoning*


python agent_cli.py
AGENT REASONING CLI
Advanced Cognitive Architectures (Gemma 3)
Working Directory: /home/ubuntu/git/agent-reasoning
REFINEMENT LOOP AGENT
This agent iteratively improves content using score-based feedback.
Generator → Critic (score 0.0-1.0) → Refiner → Loop until threshold met
Demo Query: Technical Article on Neural Networks
Write a brief technical article (2-3 paragraphs) explaining how neural networks learn.
The article should be suitable for a technical blog and include specific details about:
- Backpropagation algorithm
- Gradient descent optimization
- Loss functions
Make it technically accurate and precise.

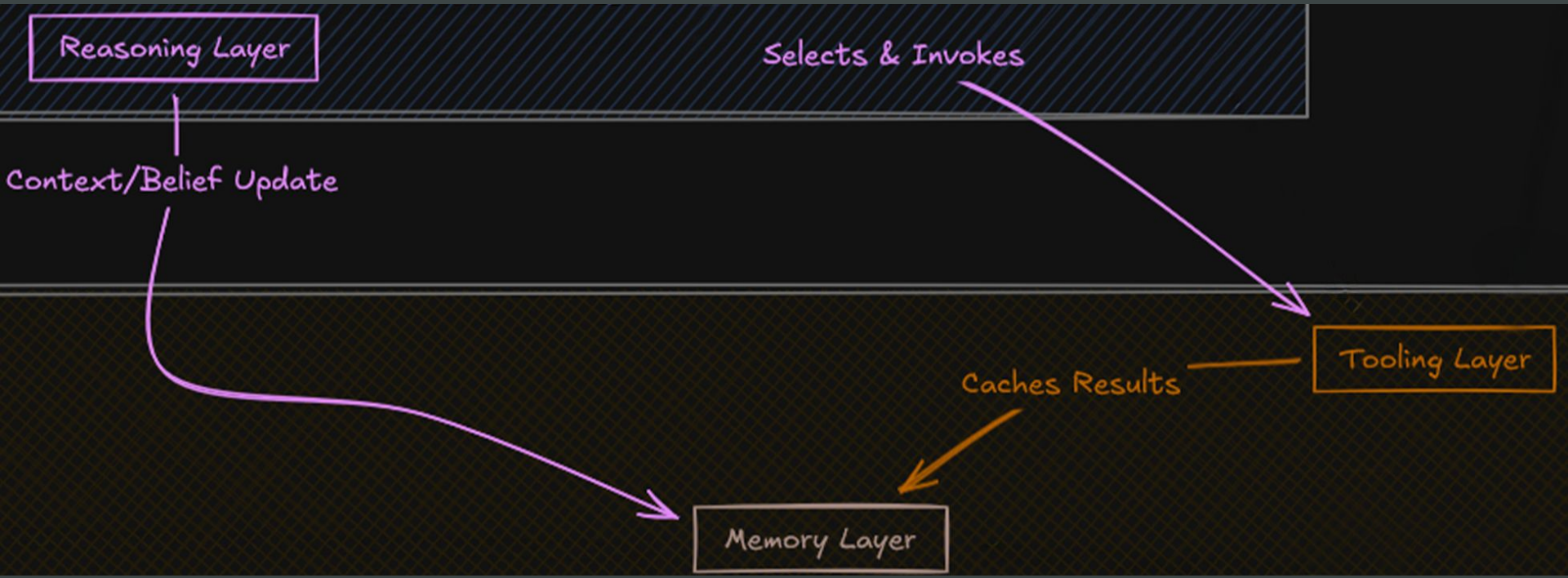
Handling infinite context

Recursive Language Models (RLMs)



Agentic Reasoning + Memory

Context belief & updates



Implementation

langchain-oracledb

This package contains the LangChain integrations with Oracle AI Vector Search.

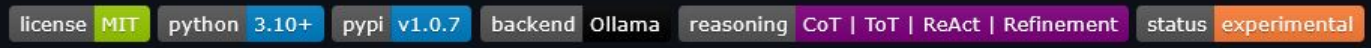
Installation

```
python -m pip install -U langchain-oracledb
```

Documentation

- Oracle AI Vector Search: Vector Store
- Oracle AI Vector Search: Generate Summary
- Oracle AI Vector Search: Document Processing
- Oracle AI Vector Search: Generate Embeddings

Agent Reasoning: The Thinking Layer




```
        AGENT REASONING CLI
Advanced Cognitive Architectures (Gemma 3)

                Reasoning Layer
Working Directory: /home/ubuntu/git/agent-reasoning

? Select an Activity: (Use arrow keys)
   Chat with Standard Agent
   Chain of Thought (CoT)
   Tree of Thoughts (ToT)
   ReAct (Tools + Web)
   Recursive (RLM) [NEW]
   Self-Reflection
   Decomposed Prompting
   Least-to-Most
   Self-Consistency
   ---------------
 » ⚔️ ARENA: Run All Compare
   Select AI Model (Current: gemma3:latest)
   Run Logic Benchmark
   ---------------
   Exit
```

AGENT REASONING CLI
Advanced Cognitive Architectures (Gemma 3)

Working Directory: /home/ubuntu/git/agent-reasoning

ℹ **REASONING AGENTS GUIDE**

Available Reasoning Strategies

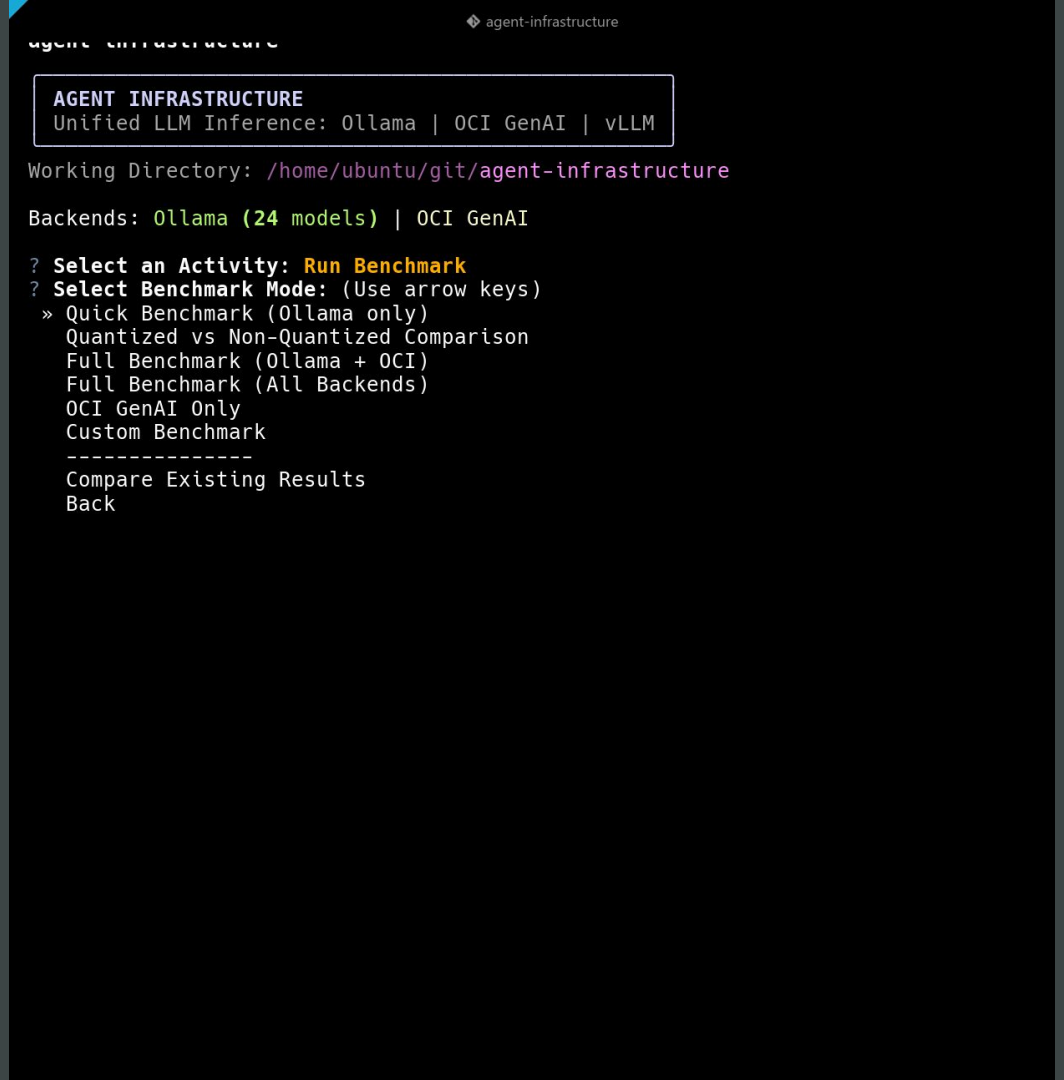
Strategy	Full Name	How It Works	Best For	Reference
standard	**Standard**	Direct generation	Baseline responses	N/A
cot	**Chain of Thought**	Step-by-step reasoning	Math, logic, analysis	Wei et al. 2022
tot	**Tree of Thoughts**	Branching exploration with pruning	Complex puzzles, riddles	Yao et al. 2023
react	**ReAct**	Reasoning + tool actions	Fact-checking, calculations	Yao et al. 2022
recursive	**Recursive LM**	Code REPL with sub_llm()	Data processing, long-context	Author et al. 2025
reflection	**Self-Reflection**	Draft → critique → refine loop	Creative writing, code	Shinn et al. 2023
decomposed	**Decomposed**	Break into sub-tasks, solve each	Planning, complex queries	Khot et al. 2022
least_to_most	**Least-to-Most**	Easiest to hardest sub-questions	Multi-step reasoning	Zhou et al. 2022
consistency	**Self-Consistency**	k samples + majority vote	Diverse problems	Wang et al. 2022
refinement	**Refinement Loop**	Score-based iterative improvement	Technical writing	Iterative Refinement
complex_refine…	**Complex Pipeline**	5-stage optimization pipeline	High-quality content	Multi-Stage Refinement

Raw compute vs. Paid models

We offer you both options: OCI Generative AI Service and computing power

Model Family	Model Name	Input Price (per 1M Tokens)	Output Price (per 1M Tokens)
Cohere	Cohere Command R (08-2024)	~$0.36 (Blended)	~$0.36 (Blended)
Cohere	Cohere Command R+ (08-2024)	~$6.24 (Blended)	~$6.24 (Blended)
Google	Google Gemini 2.5 Pro (< 200k context)	$1.25	$10.00
Google	Google Gemini 2.5 Pro (> 200k context)	$2.50	$15.00
Google	Google Gemini 2.5 Flash	$0.30	$2.50
Google	Google Gemini 2.5 Flash-Lite	$0.10	$0.40
Meta	Meta Llama 4 Maverick	~$0.72 (Blended)	~$0.72 (Blended)
Meta	Meta Llama 4 Scout	~$0.72 (Blended)	~$0.72 (Blended)
Meta	Meta Llama 3.3 (70B)	~$0.72 (Blended)	~$0.72 (Blended)
Meta	Meta Llama 3.2 90B Vision	~$2.00 (Blended)	~$2.00 (Blended)
Meta	Meta Llama 3.2 11B Vision	~$0.72 (Blended)	~$0.72 (Blended)
Meta	Meta Llama 3.1 (405B)	~$10.68 (Blended)	~$10.68 (Blended)
Meta	Meta Llama 3 (70B)	~$0.72 (Blended)	~$0.72 (Blended)
xAI	xAI Grok Code Fast 1	$0.20	$1.50
xAI	xAI Grok 4.1 Fast	$5.00	$25.00
xAI	xAI Grok 4 Fast	$5.00	$25.00
xAI	xAI Grok 4	$3.00	$15.00
xAI	xAI Grok 3	$3.00	$15.00
xAI	xAI Grok 3 Mini	$0.30	$0.50
xAI	xAI Grok 3 Fast	$5.00	$25.00
xAI	xAI Grok 3 Mini Fast	$0.60	$4.00

Model Family	Model Name	Estimated Price (per 1M Tokens)
Cohere	Cohere Embed 4	~$0.40
Cohere	Cohere Embed English Image 3	~$0.40
Cohere	Cohere Embed English Light Image 3	~$0.40
Cohere	Cohere Embed Multilingual Image 3	~$0.40
Cohere	Cohere Embed Multilingual Light Image 3	~$0.40
Cohere	Cohere Embed English 3	~$0.40
Cohere	Cohere Embed English Light 3	~$0.40
Cohere	Cohere Embed Multilingual 3	~$0.40
Cohere	Cohere Embed Multilingual Light 3	~$0.40


agent-infrastructure
AGENT INFRASTRUCTURE
Unified LLM Inference: Ollama | OCI GenAI | vLLM
Working Directory: /home/ubuntu/git/agent-infrastructure
Backends: Ollama (24 models) | OCI GenAI
? Select an Activity: Run Benchmark
? Select Benchmark Mode: (Use arrow keys)
» Quick Benchmark (Ollama only)
Quantized vs Non-Quantized Comparison
Full Benchmark (Ollama + OCI)
Full Benchmark (All Backends)
OCI GenAI Only
Custom Benchmark

Compare Existing Results
Back

Live Results

gemma3 (7B)

Prompt	Lat	TTFT	Tok	TPS
Why is the sky ...	9774	373	505	51.7
Write a Python ...	19367	8140	839	43.3

gemma3 (270M)

Prompt	Lat	TTFT	Tok	TPS
Why is the sky ...	1085	198	112	103.2
Write a Python ...	5750	183	687	119.5
Explain quantum...	1987	173	206	103.7

OCI GenAI

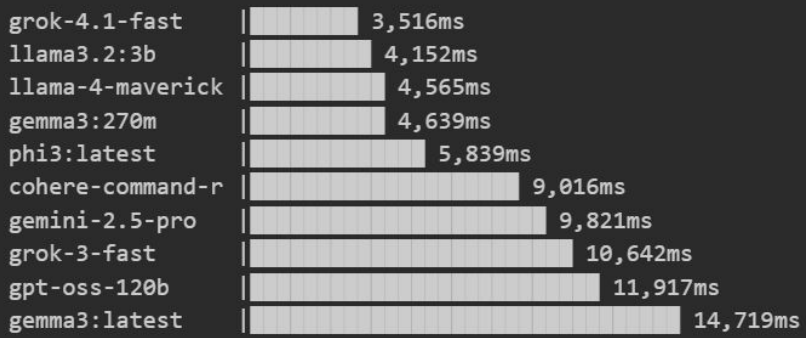
Prompt	Lat	TTFT	Tok	TPS
Write a Python ...	3287	488	200	60.9
Explain quantum...	3579	670	200	55.9
What are the ma...	2627	656	200	76.1
Summarize the p...	1830	367	134	73.2

4. Detailed Results

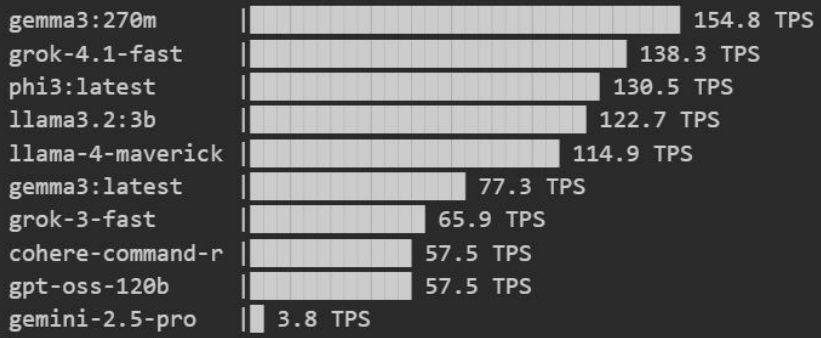
Full Comparison (Sorted by Latency)

Source	Model	Provider	Latency	TTFT	Tokens	TPS	Cost	Success
OCI	grok-4.1-fast	xai	3,516ms	3,516ms	486	138.3	$0.00074	6/6
Ollama	llama3.2:3b	local	4,152ms	1,341ms	415	122.7	GPU	6/6
OCI	llama-4-maverick	meta	4,565ms	4,565ms	525	114.9	$0.00079	6/6
Ollama	gemma3:270m	local	4,639ms	1,274ms	654	154.8	GPU	6/6
Ollama	phi3:latest	local	5,839ms	1,608ms	646	130.5	GPU	6/6
OCI	cohere-command-r	cohere	9,016ms	9,016ms	518	57.5	$0.00078	6/6
OCI	gemini-2.5-pro	google	9,821ms	9,821ms	37	3.8	$0.00006	6/6
OCI	grok-3-fast	xai	10,642ms	10,642ms	702	65.9	$0.00106	6/6
OCI	gpt-oss-120b	openai	11,917ms	11,917ms	686	57.5	$0.00139	6/6
Ollama	gemma3:latest	local	14,719ms	750ms	1,138	77.3	GPU	6/6

Latency Comparison (Lower is Better)



Throughput Comparison (Higher is Better)



About

Technical resources for AI developers to build applications, agents, and systems using Oracle AI Database and OCI services

oracle-devrel.github.io/oracle-ai-develop...

kubernetes ai artificial-intelligence agents oracle-database oraclejet rag kustomize generative-ai ai-developer agentmemory oracleaidatabase

- Readme
- Contributing
- Security policy
- Activity
- Custom properties
- 92 stars
- 12 watching
- 38 forks
- Audit log

Report repository

Notebooks (/notebooks)

Jupyter notebooks and interactive tutorials covering:

- AI/ML model development and experimentation
- Oracle Database AI features and capabilities
- OCI AI services integration patterns
- Data preparation and analysis workflows
- Agent development and orchestration examples

Name	Description	Stack	Link
agentic_rag_langchain_oracledb_demo	Multi-agent RAG with langchain-oracledb: OracleVS, OracleEmbeddings, OracleTextSplitter, and CoT agents	Oracle AI Database, langchain-oracledb, Ollama	
fs_vs_dbs	Compare filesystem vs database agent memory architectures.	LangChain, Oracle AI Database, OpenAI	
memory_context_engineering_agents	Build AI agents with 6 types of persistent memory.	LangChain, Oracle AI Database, OpenAI, Tavily	
oracle_langchain_example	Build a RAG application using Oracle 26ai vector storage and LangChain	Oracle AI Database, langchain-oracledb, HuggingFace	
oracle_rag_agents_zero_to_hero	Learn to build RAG agents from scratch using Oracle AI Database.	Oracle AI Database, OpenAI, OpenAI Agents SDK	
oracle_rag_with_evals	Build RAG systems with comprehensive evaluation metrics	Oracle AI Database, OpenAI, BEIR, Galileo	

oracle-devrel / oracle-ai-developer-hub

Questions?

Thanks for coming to the session!


ORACLE