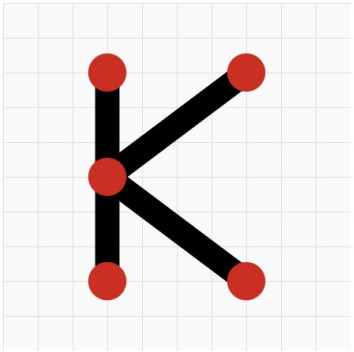
From Deduction Games to Telecom Network Troubleshooting

Simon Lucas
Professor of AI
Queen Mary University of London,
Kwiksim Limited

Simulation-Enhanced Agents for Telco Troubleshooting



Simulation-Based AI

for decisions you can trust

KWIKSIM Limited

HUAWEI

Identify (and repair) faults

Read problem description

Use network tools

Max: accuracy

Min: time, cost, tool-calls

Deduction games and troubleshooting - Agentic AI

Challenge:

- A hidden cause must be inferred from incomplete observations.
- Arises in both games and telecom network troubleshooting
- Approach: use simulation-based reasoning and deduction
- Agentic LLM: process context and make tool calls
- Simulation model + solvers
- Use network tool calls to gather information
- Solver library to reason about it

Does deduction game AI have anything to offer?

- Information Set Entropy Search + LLM + Constraint Satisfaction Problem (CSP)
- See Fandi Meng et al:
  - PhD AI for Deduction Games, QMUL 2026
  - CSP4SDG: Constraint and Information-Theory Based Role Identification in Social Deduction Games with LLM-Enhanced Inference, K Xu, F Meng, C Verbrugge, SM Lucas, Proceedings of the AAAI Conference on Artificial Intelligence 2026
  - Deduction game framework and information set entropy search, F Meng, S Lucas, 2024 IEEE Conference on Games 2024
- Competitive results: Cluedo, Mastermind, Fake coin game etc

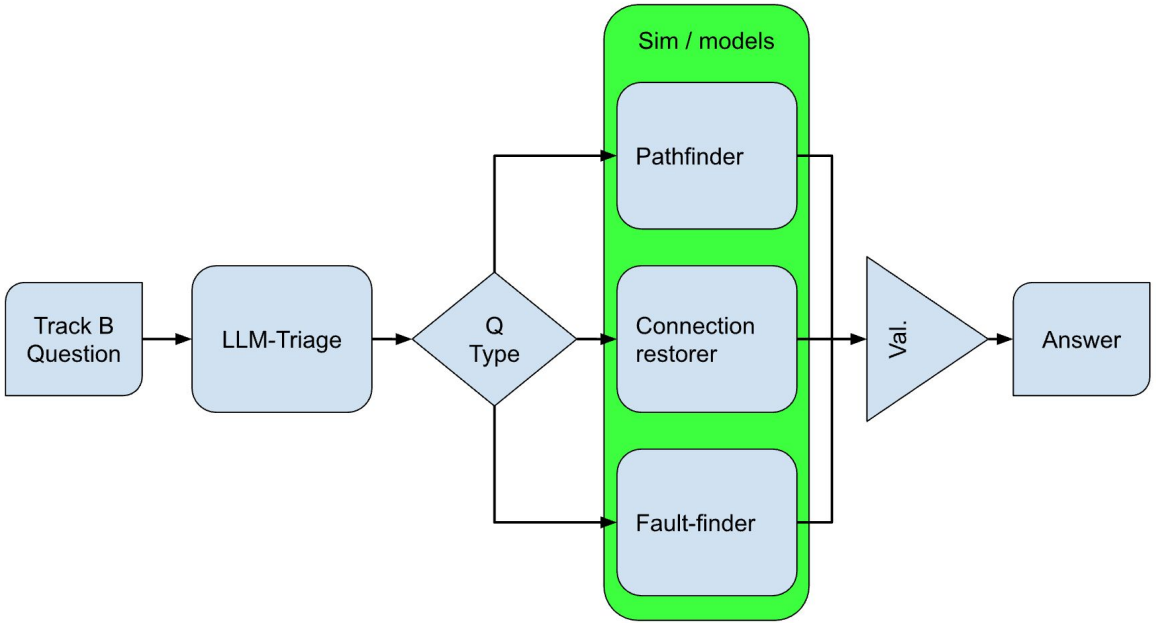
Question 1 🔗 Connection Discovery

Scenario ID: `535afb0d-fa81-419b-9bcc-b456d032df5d`

Question: The link planning data of Gamma-Aegis-01 within the Big Data Zone has been accidentally deleted, and it is necessary to restore the link connections of the interfaces whose status is UP on the Gamma-Aegis-01 node. The link status of interfaces can be queried through several methods: 1. Query the link status directly. 2. Query the interface description (which contains the remote node and remote port), and then confirm the link status through the port and IP correspondence information in the ARP tables of the local node and the remote node. If the interface description and the ARP table port information are different, the ARP table port information shall prevail. Link status query is the preferred query method. When the status cannot be queried through that method, use the second method to confirm the port connections. Based on the actual network topology, supplement the link information for the interfaces that are UP on the Gamma-Aegis-01 node in the plan. Format requirement: local node name(local port number)->remote node name(remote port number). Each line represents one link, with no extra blank lines in the middle or at the end, and no extra whitespace characters before, after, or within each line.

The Agentic System for Telco Challenge

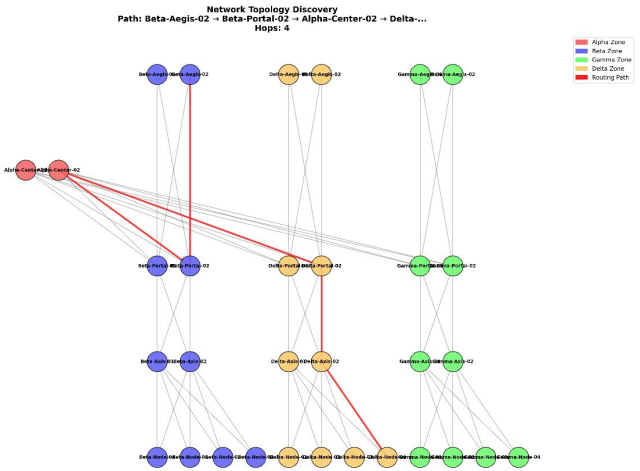
- LLM analyses question with telco-challenge prompt
- Calls on agentic process for question type
- Makes tool-calls and uses simulator / model to build solution (e.g. Pathfinder builds path)
- Pydantic data classes used to validate and format solution

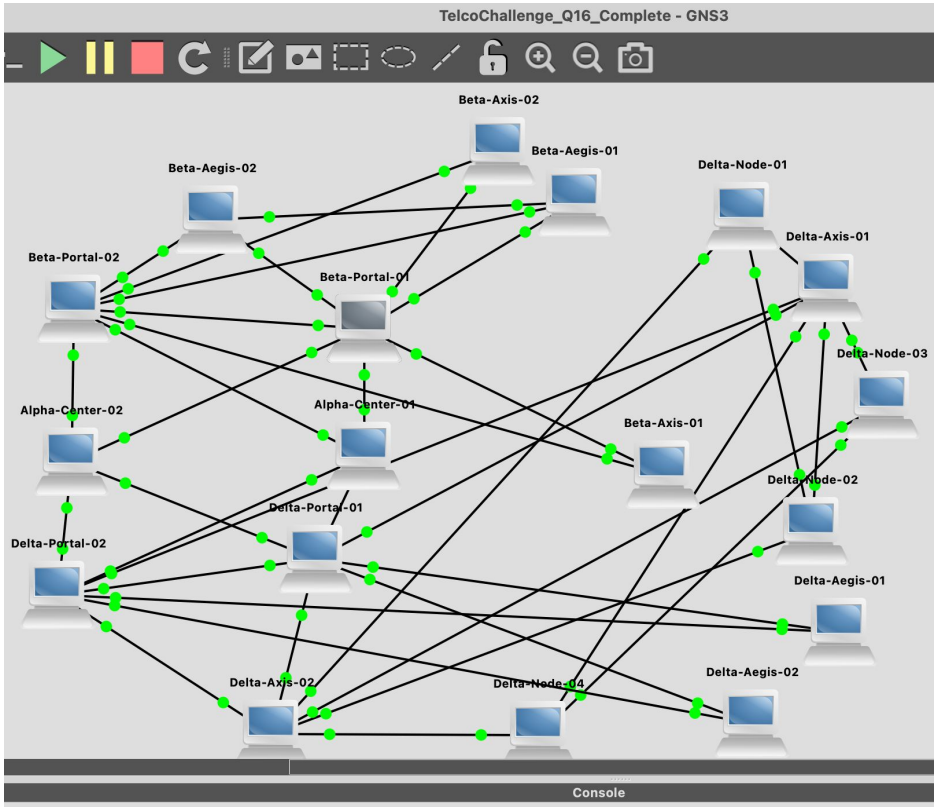


Simulation / Data Model: example network models

Data is extracted from tool calls and used to build simulation using GNS3 or plain Python

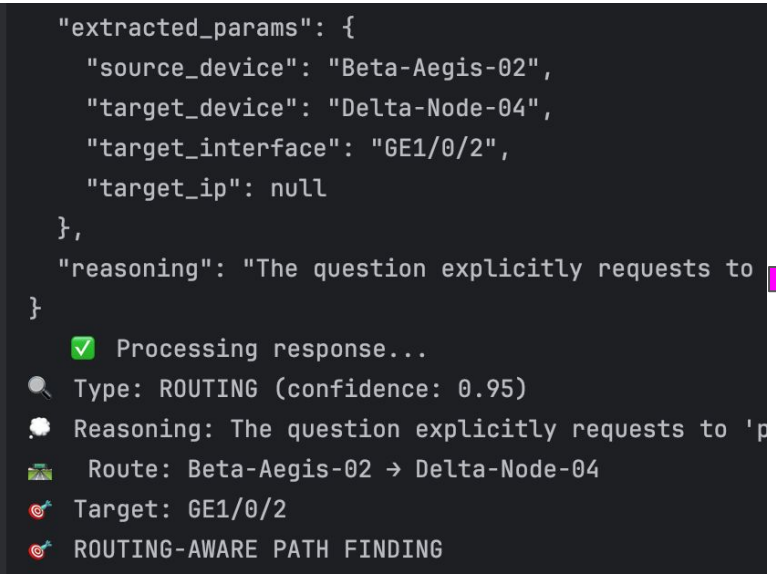
LLM agent drives process

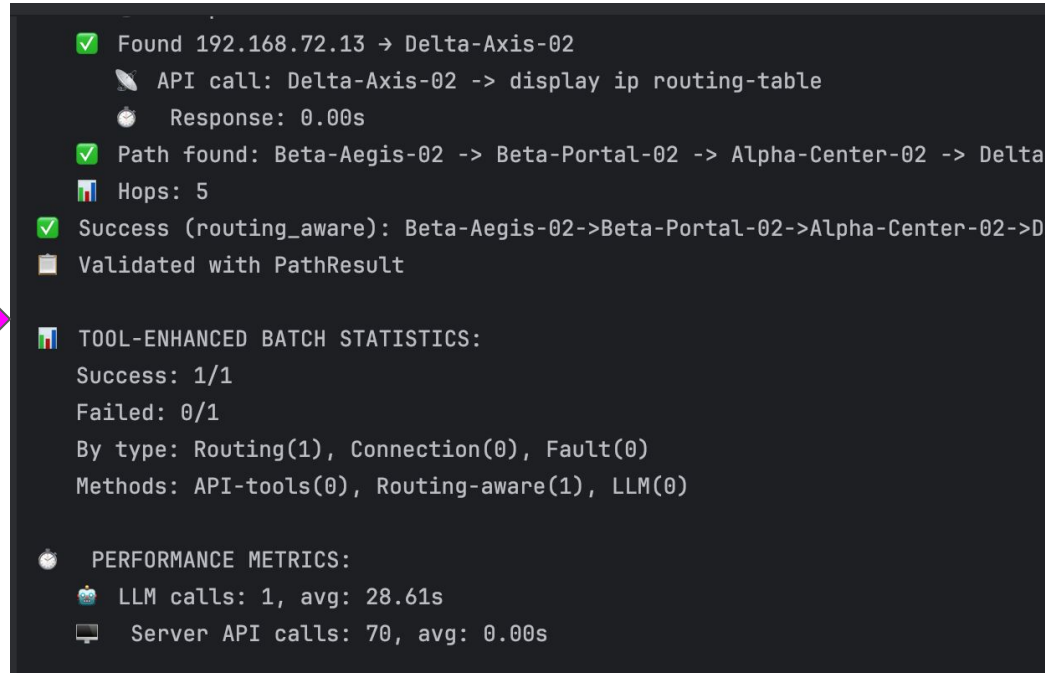




Example highlights

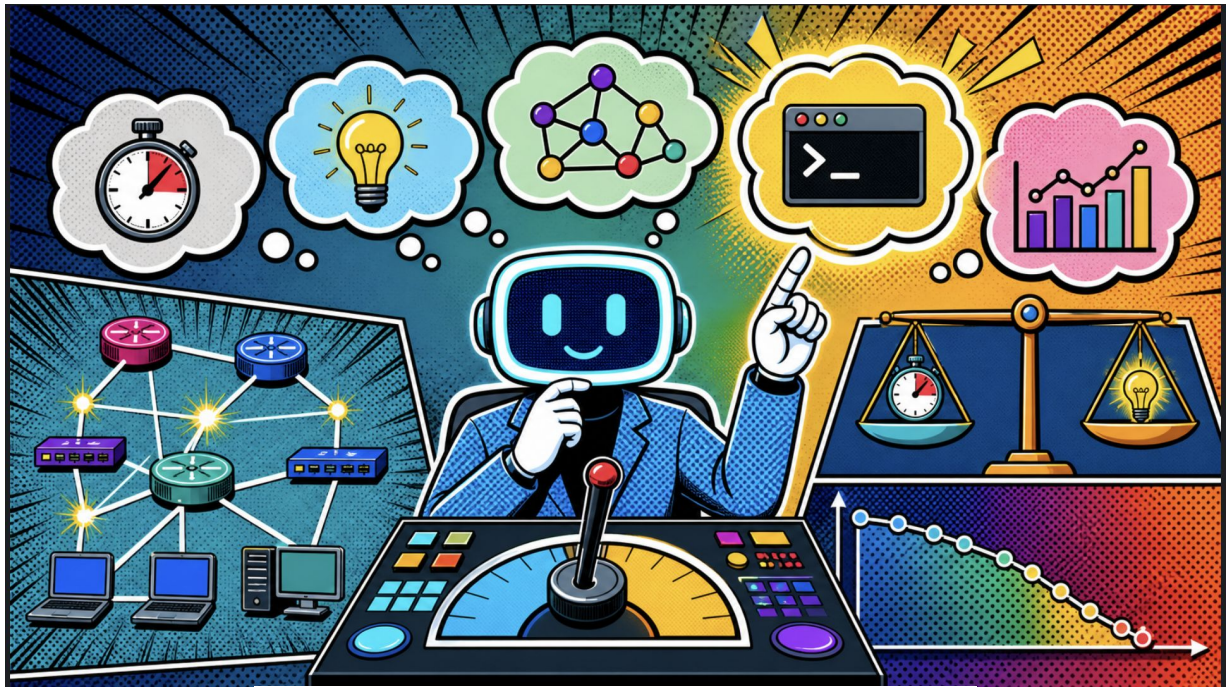
- LLM identifies question type and params then hands over to path finder





Joint Hypothesis Agent for Fault Finding

- Uses LLM to extract context and problem data
- Maintains a joint hypothesis space
- Makes tool calls to minimise entropy of posterior distribution
- Then JH agent calls tools directly
- Initial results on **NIKA**



A Network Arena for Benchmarking AI Agents on Network Troubleshooting

ZHIHAO WANG, University of Electronic Science and Technology of China (UESTC), China
ALESSANDRO CORNACCHIA, KAUST, Saudi Arabia
ALESSIO SACCO, Politecnico di Torino, Italy
FRANCO GALANTE, Politecnico di Torino, Italy
MARCO CANINI, KAUST, Saudi Arabia
DINGDE JIANG, University of Electronic Science and Technology of China (UESTC), China

Tool Calls as Actions

A diagnostic action is not just a tool name.

It includes the tool, target device or devices, and parameter values:

$$a = \texttt{frr\_exec}(\texttt{router\_core\_3}, \texttt{"show ip ospf neighbor"}).$$

The action space is therefore:

$$\mathcal{A} = \{(\text{tool}, \text{parameters})\}.$$

Choosing good parameter values is part of diagnosis.

Uncertainty

The uncertainty in the current posterior is measured by entropy:

$$H(P_t) = -\sum_{h \in \mathcal{H}} P_t(h) \log P_t(h).$$

Low entropy means the posterior is concentrated on a small number of hypotheses.

The objective of tool selection is to reduce entropy quickly and reliably.

Information Gain Heuristic

The expected information gain of action a is:

$$\mathrm{IG}(a) = H(P_t) - \mathbb{E}[H(P_{t+1}) \mid a].$$

A cost-aware policy selects:

$$a^\star = \arg\max_{a \in \mathcal{A}} \frac{\mathrm{IG}(a)}{\mathrm{cost}(a)}.$$

This is the principled way to minimise tool calls while moving toward a confident diagnosis.

Example posterior update

Suppose the task topology implies that router r should have three OSPF neighbours.

The agent calls:

$$a = \texttt{frr\_exec}(r, \texttt{"show ip ospf neighbor"}).$$

If the output shows zero full neighbours, the observation is:

$$o = \texttt{ospf\_full\_neighbors}(r, 0).$$

The update strongly increases:

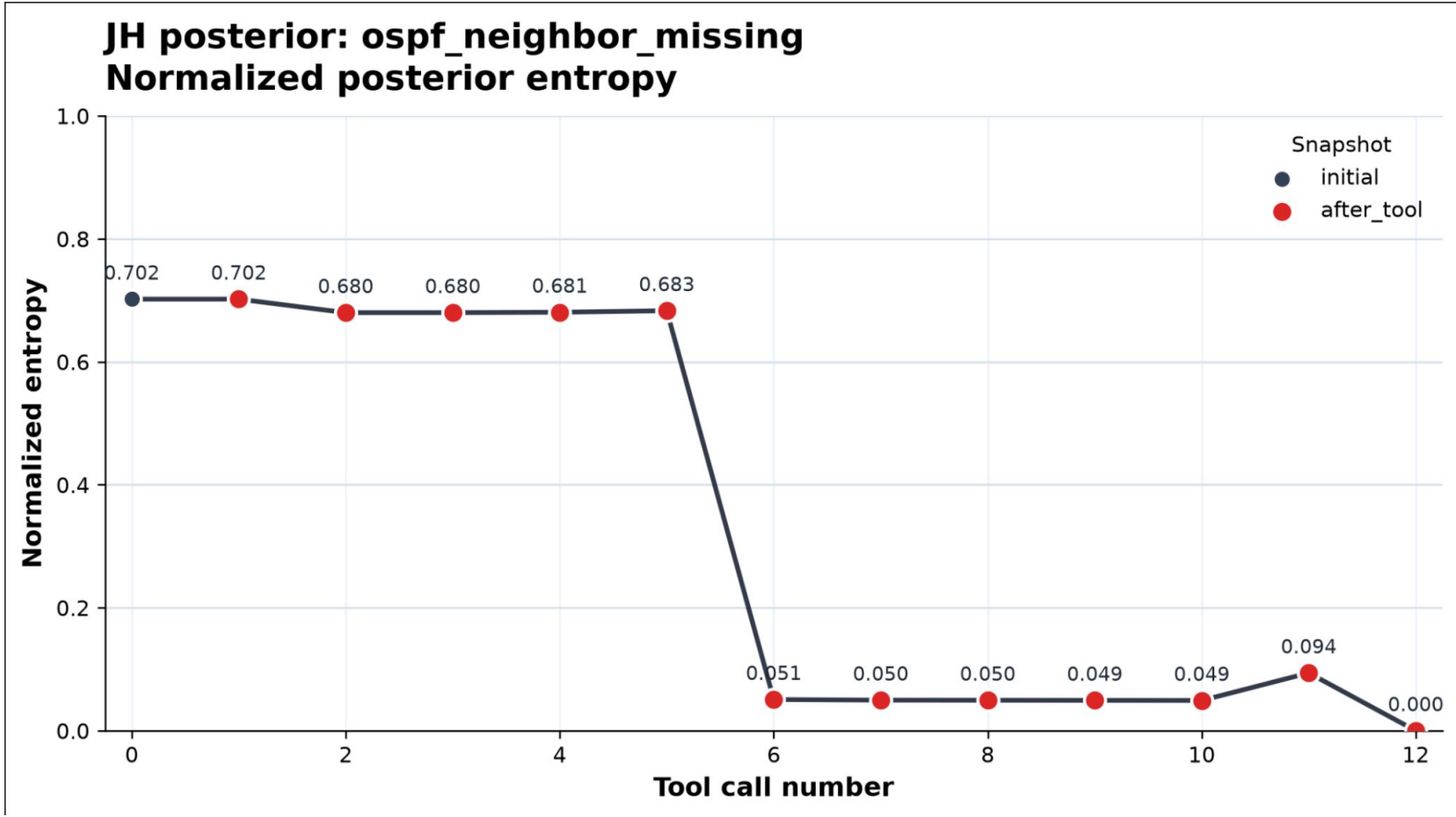
$$P_t(\texttt{ospf\_neighbor\_missing}, r).$$

Fault finding loop

Agent proceeds in entropy reduction loop until:

- Entropy drops below threshold
- Run out of tool call / time budget
- No calls have expected information gain

Entropy may even increase with tool response



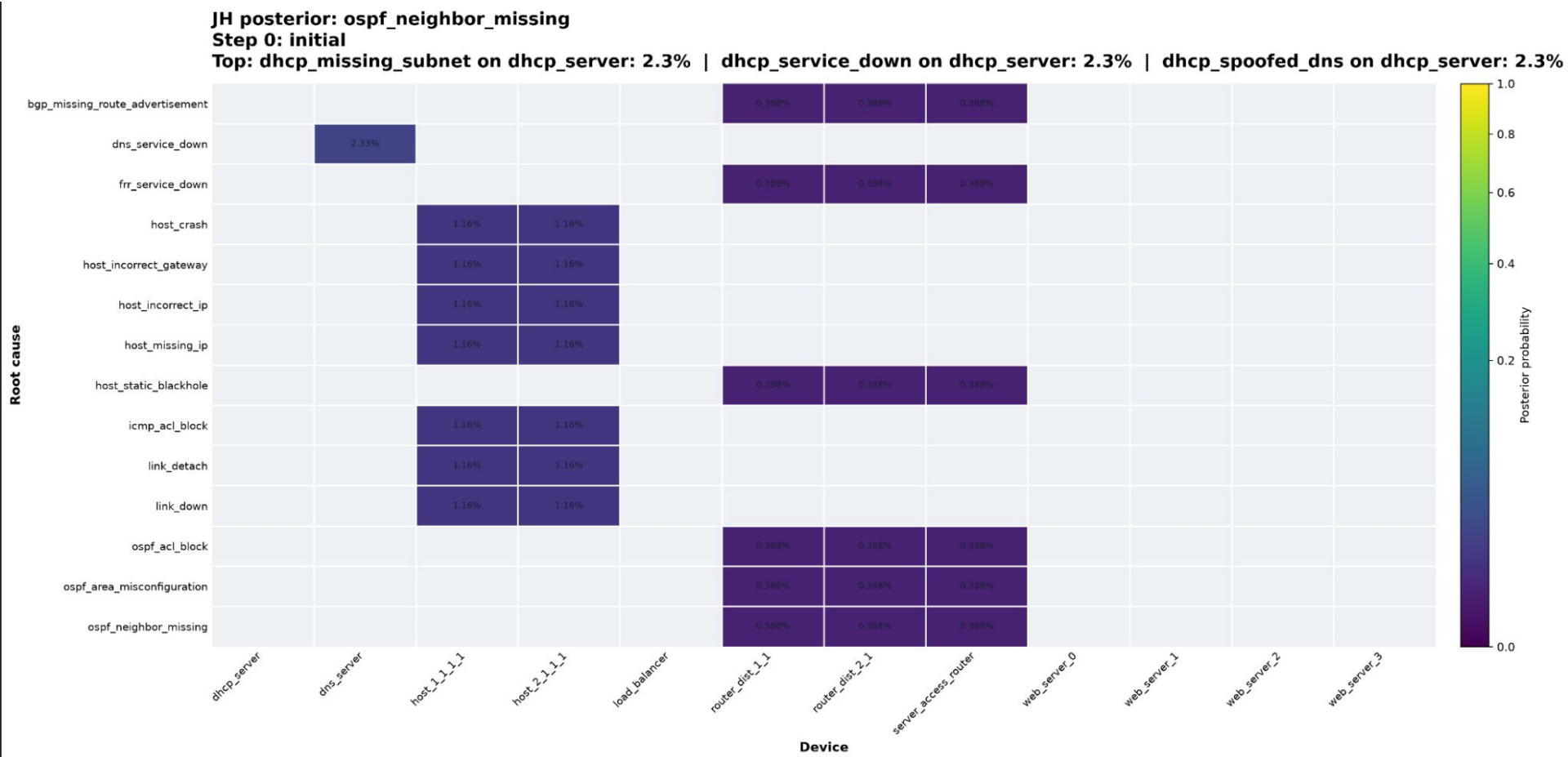
NIKA Example Problem Instance and JH Run

You are provided with the following network description and its current state:\nNetwork Description: An enterprise hierarchical network using OSPF with multiple areas, built from three core routers, distribution routers, and bridged access switches. User hosts sit in subnets of the form 10.<core>.<dist>.0/24, obtain their IP configuration via DHCP (with dist-layer DHCP relay to a central DHCP server), and reach a server farm in 10.200.0.0/24 that hosts a DNS server for the local zone, several Apache web servers (web0.local\u2026web3.local), An Nginx HTTP load balancer published as web99.local, which load-balances requests to three backend web servers. Note that by design the backend web servers should not be directly accessible from the hosts. All infrastructure and server networks are advertised by FRR OSPF so that hosts can resolve and access the web services end-to-end. Switches: switch_access_1_1_1, switch_access_2_1_1\nHosts: host_1_1_1_1, host_2_1_1_1\n … ***(more device names and connections omitted for presentation)***

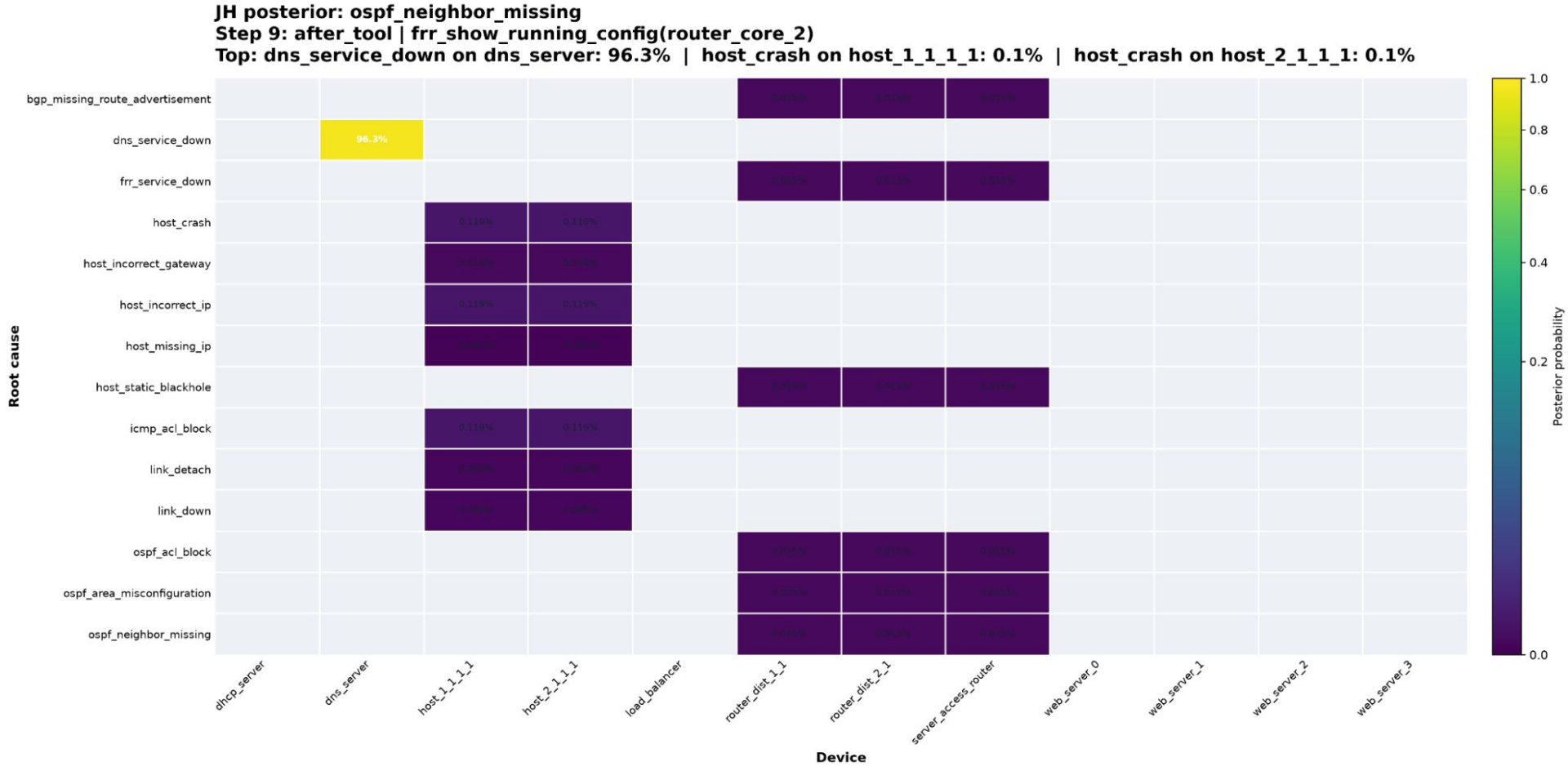
Your goal is to analyze the network condition and, if needed, use the available tools. You need to generate a troubeshooting diagnosis report. The report should reflect your assessment of the network's health, indicate any abnormal behavior you identify, and describe relevant findings based on your analysis.

Focus on producing an informative and coherent diagnostic report derived from the network state.
Do not need to propose any solutions or remediation steps at this stage.

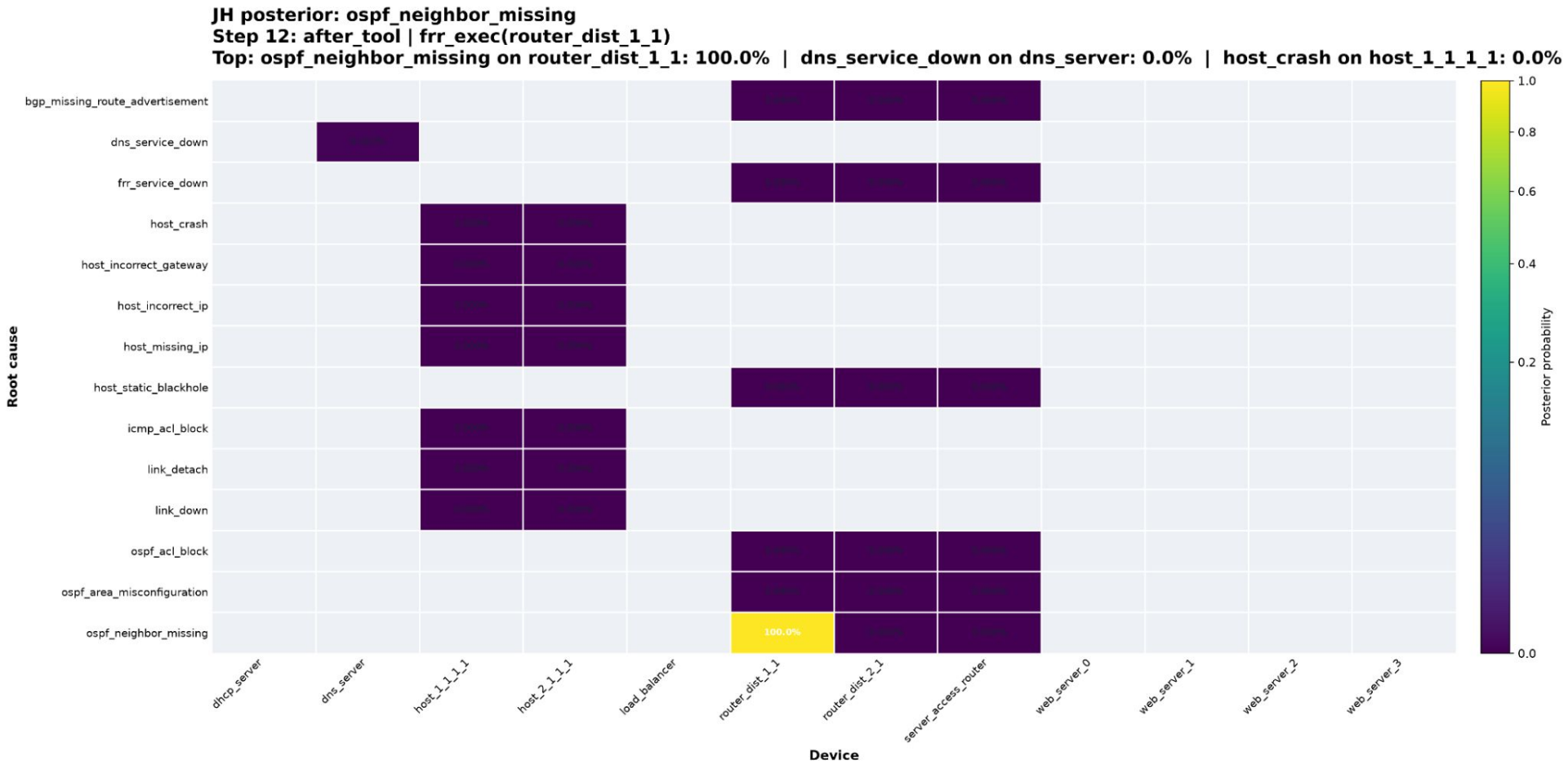
Initial


JH posterior: ospf_neighbor_missing
Step 0: initial
Top: dhcp_missing_subnet on dhcp_server: 2.3% | dhcp_service_down on dhcp_server: 2.3% | dhcp_spoofed_dns on dhcp_server: 2.3%
Root cause
bgp_missing_route_advertisement
dns_service_down
frr_service_down
host_crash
host_incorrect_gateway
host_incorrect_ip
host_missing_ip
host_static_blackhole
icmp_acl_block
link_detach
link_down
ospf_acl_block
ospf_area_misconfiguration
ospf_neighbor_missing
2.33%
1.16%
0.388%
dhcp_server
dns_server
host_1_1_1_1
host_2_1_1_1
load_balancer
router_dist_1_1
router_dist_2_1
server_access_router
web_server_0
web_server_1
web_server_2
web_server_3
Device
Posterior probability
1.0
0.8
0.6
0.4
0.2
0.0

False dawn


JH posterior: ospf_neighbor_missing
Step 9: after_tool | frr_show_running_config(router_core_2)
Top: dns_service_down on dns_server: 96.3% | host_crash on host_1_1_1_1: 0.1% | host_crash on host_2_1_1_1: 0.1%
Root cause
bgp_missing_route_advertisement
dns_service_down
frr_service_down
host_crash
host_incorrect_gateway
host_incorrect_ip
host_missing_ip
host_static_blackhole
icmp_acl_block
link_detach
link_down
ospf_acl_block
ospf_area_misconfiguration
ospf_neighbor_missing
96.3%
0.119%
dhcp_server
dns_server
host_1_1_1_1
host_2_1_1_1
load_balancer
router_dist_1_1
router_dist_2_1
server_access_router
web_server_0
web_server_1
web_server_2
web_server_3
Device
Posterior probability
1.0
0.8
0.6
0.4
0.2
0.0

Converges on correct answer



Proof of Concept Results - NIKA Benchmark (Smoke Test)

Agent / model	Correct	Tools	LLM	Input tok.	Output tok.
Joint hypothesis / GPT-5-mini	5/5	29	5	14,062	2,822
ReAct / GPT-5-mini	4/5	75	50	407,034	21,372
Joint hypothesis / GPT-4o-mini	5/5	30	5	14,276	493
ReAct / GPT-4o-mini	2/5	25	27	55,950	4,065

- Save on tokens
- More accurate, especially for simpler models

Conclusions and future directions

- We significantly improve Agentic AI using:
  - Triage, Deduction, Model-Building / Simulation, Search
- Solver Library Approach:
  - Develop Python utilities to build model of problem and solve it
  - Works well for known class of problems with fixed faults
- Open problems:
  - Intermittent or dynamically injected faults
  - Automated solver development using coding agents / evolutionary program search