A. Threat channels

1) Tool output injection

Trusted instruction (system / user)

Untrusted content (tool output)

2) Slack / realistic multi-step trajectory injection

Trusted instruction (system / user)

Untrusted content (Slack messages / trajectory steps)

3) Memory poisoning

Trusted instruction (system / user)

Untrusted content (poisoned memory entries)

LLM Agent (reasoning+ planning + acting)

B. Readable early

Transformer depth (layers)

Linear probe

Source-role information becomes linearly decodable

L0 ··· Ln

early layers | mid layers | late layers

Readable early

··· L4 L5 L6 L7 L8 ···

Instruction (source = instruction)

Data (source = data)

Probe saturates at L4–L8

Recognition is available before action is committed.

C. Actionable late

Transformer depth (layers)

Activation patching

L0 ··· Ln

early layers | mid layers | late layers

Action / tool decision becomes causally controllable

Late-band commitment site

··· L18 L19 L20 L21 L22 ···

Actionable late

Recognition precedes action commitment.

Tool choice

Tool arguments

Hijack vs. resist

D. Matched intervention

Direction d vs. channel (distribution)

	Controlled tool-output	AgentDojo-Slack	Memory poisoning	Obfuscated attacks
d_{tool}	✓	✗	✗	✗
d_{slack}	✗	✓	✗	✗
d_{memory}	✗	✗	✓	✗
d_{obf}	✗	✗	✗	✓

No universal safety subspace

Mitigation requires the right layer and the right channel/ distribution-matched direction.

Knowing is early; acting is late; defending is channel- and distribution-specific.