OrKA-Reasoning: A Cognitive AI Framework for Transparent, Traceable Thinking

Abstract
OrKa-reasoning is a cognitive AI framework that turns model calls into observable thinking. Instead of a single prompt and a single output, OrKa composes specialized agents, a six-layer memory model, and iterative Loops of Truth that let the system argue, critique, and converge. The result is reasoning that is visible, debuggable, and improvable. Inspired by Minsky's Society of Mind, OrKa treats intelligence as coordination across roles and memories rather than a monolithic model. The framework introduces GraphScout, a beta path-discovery component in v0.9.3 that explores orchestration graphs and suggests alternative routes when flows stall. We include execution traces and early benchmarks to show how multi-agent debate, agreement finding, and memory recall produce more robust outcomes than LLM-only prompting. OrKa-reasoning is research stage and already practical for engineers who want explainability, traceability, and early signs of metacognition without heavy academic overhead. It is a step toward modular, ethical AI where decisions are auditable and alignment is part of the runtime, not an afterthought.

Introduction
Artificial intelligence in 2025 is still dominated by single-model prompting. Most systems wrap an LLM with thin layers of tooling, producing outputs that are fluent but fragile. These pipelines lack transparency, cannot explain their reasoning, and often collapse under real-world complexity. Developers are forced to treat large models as black boxes, while users are left with results they cannot audit or challenge. What is missing is a framework that treats reasoning as a process to be observed, critiqued, and improved, rather than a one-shot output to be consumed.

OrKa-reasoning was created to address this gap. It is not a prompt wrapper but a cognitive framework that orchestrates multiple agents, layered memory, and iterative debate loops into transparent reasoning flows. Inspired by Marvin Minsky's idea of a Society of Mind, OrKa treats intelligence as an emergent property of collaboration and critique among specialized roles. In recent traces, we see agents taking progressive, conservative, realist, and purist positions, debating and refining perspectives before converging on shared ground. This is reasoning that is no longer hidden inside a single model's output. It is reasoning that can be logged, replayed, and studied.

Although still in research stage, OrKa-reasoning has already been stress-tested beyond toy demos. A 1000-run benchmark with a local 1.5B parameter model completed without failures, logging over two thousand agent executions with stable latency and cost efficiency. In practice, this means OrKa can orchestrate debate loops and memory lookups reliably at scale, even on consumer hardware. Traces from version 0.8.0 show agents exploring disagreements, storing short-term and long-term memories in Redis, and calculating agreement scores across rounds. These runs demonstrate that the system is not theoretical. It is real, auditable, and already usable for engineers who want transparent reasoning instead of opaque generations.

The framework continues to evolve. In version 0.9.3, OrKa introduced GraphScout, a beta feature that can explore orchestration graphs and propose new reasoning paths when existing flows stall. This marks an important step toward self-discovery: instead of executing only what the designer prescribes, OrKa can begin to adapt its strategy during runtime. Together with its multi-agent debates, memory presets, and Loops of Truth, GraphScout points toward the possibility of systems that are not only explainable but also capable of reflecting on their own reasoning structures.

Background and Philosophy

Artificial intelligence has long cycled between waves of optimism and disappointment. Symbolic systems promised structured reasoning but struggled with scale and adaptability. Connectionist models such as neural networks offered flexibility but often hid their logic behind opaque layers. Large Language Models (LLMs) have pushed generative AI into the mainstream, yet they bring the same challenge: fluency without transparency. A prompt may yield a convincing answer, but the process that led there remains hidden.

Marvin Minsky's *Society of Mind* argued that intelligence emerges from a collection of simple agents, each with limited scope, collaborating and competing to solve problems. This perspective contrasts with today's monolithic LLMs, which compress reasoning into a single undifferentiated output. OrKa-reasoning draws directly from Minsky's insight. It assumes that cognition should be modular, traceable, and explainable, not collapsed into one opaque step.

The gap in the current AI ecosystem is clear. Developers rely on increasingly complex prompt engineering to coax reliable outputs from single models. Frameworks like LangChain or AutoGen extend capabilities but still center on LLM calls as the unit of intelligence. What is missing is a system that elevates the orchestration itself into the locus of intelligence. OrKa-reasoning positions the orchestration graph as the true cognitive substrate, where reasoning paths, memory recall, and agent debates define how intelligence unfolds.

In this framing, OrKa-reasoning is less about controlling LLMs and more about building a runtime for cognition. It is designed not only to generate answers but also to **show its work**, producing traces that can be inspected, replayed, and refined. The result is a framework that is aligned with both engineering needs and ethical imperatives: if intelligence is modular and observable, it can also be audited, aligned, and trusted.

1. Defining OrKa-Reasoning

OrKa-reasoning is a cognitive AI framework designed to move beyond the limitations of LLM-only prompting. Instead of relying on a single model to deliver answers, OrKa-reasoning orchestrates a society of agents, layered memory, and iterative loops into a structured reasoning process. At its core, OrKa-reasoning rests on four pillars:

1. **Agentic Multiplicity**
   OrKa coordinates multiple specialized agents that embody distinct perspectives such as realist, progressive, conservative, or purist. This allows the system to simulate pluralistic reasoning, surface conflicts, and iteratively refine outcomes.

2. **Structured Memory**
   OrKa integrates a six-layer cognitive memory model inspired by Marvin Minsky's *Society of Mind*. Memories are distributed across short-term, long-term, semantic, episodic, procedural, and immediate layers, with decay logic and vector retrieval. This enables contextual and historically grounded reasoning.

3. **Loops of Truth (LoT)**
   OrKa introduces an internal mechanism for iterative debate loops. Agents argue, critique, and converge across cycles, with agreement scores and refinement phases logged at each step. This creates a transparent record of reasoning and demonstrates early signs of metacognition. The system does not simply produce answers, it evaluates the quality and alignment of those answers through structured convergence.

4. **GraphScout for Path Discovery**
   Beginning with version 0.9.3, OrKa includes GraphScout as a beta feature. GraphScout explores the orchestration graph itself, probing alternative branches and suggesting new reasoning paths when existing flows fail to converge. This transforms OrKa from a static execution engine into a system that can adaptively discover its own strategies, moving closer to a self-discovering cognitive framework.

Together, these four components establish OrKa-reasoning as a research-stage substrate for transparent and traceable cognition. It is not just an orchestration framework, but an early demonstration of explainable and metacognitive AI.

2. Architecture and Execution

OrKa-reasoning is built around declarative orchestration. Instead of wiring models through ad-hoc scripts, users define reasoning flows in YAML. Each flow specifies the orchestrator, its strategy, and the agents or service nodes involved. The orchestrator then executes these agents with fork-join semantics, routing results according to confidence distributions, agreement scores, or memory lookups.

At the core of OrKa-reasoning is a declarative execution engine. Instead of manually wiring scripts or chaining prompts, users define orchestration flows in YAML. These flows describe how agents, loops, and memory nodes interact. The orchestrator reads the YAML, executes the specified strategy, and logs every step for traceability.

A representative workflow is the **cognitive_society_simple**, shown below. It demonstrates how OrKa structures a debate loop between agents, measures their agreement, and synthesizes a final answer:

This workflow (appendix) captures several architectural principles:

- **Nested orchestration.** The `simple_debate_loop` contains an internal workflow that runs progressive and conservative agents before passing their outputs to an agreement checker. This shows how OrKa treats orchestration itself as a first-class element of cognition.

- **Agreement scoring.** The `score_extraction_config` ensures that each debate round returns a measurable agreement score. This allows the loop to determine whether consensus has been reached or if further rounds are needed.

- **Looped refinement.** The loop can run up to three cycles. Each round introduces the possibility of updated arguments, different phrasing, or new supporting evidence before moving toward convergence.

- **Final synthesis.** The `final_answer` agent does not simply generate text; it has access to loop metadata such as the number of rounds completed and the final score. This lets it explain outcomes transparently, including cases where full agreement was not achieved.

In live traces, similar structures run with additional roles such as pragmatic realist and ethical purist. Their outputs are joined, agreement is measured, and memory nodes log context into Redis for later retrieval. Every decision and score is written into the orchestration log, making the reasoning path replayable and auditable.

In practice, this means OrKa-reasoning executes reasoning as a process, not as a single opaque call. Each run is a cognitive sequence that can be inspected, critiqued, and improved over time.

3. Benchmarks and Evaluation

Evaluation of OrKa-reasoning spans three layers: **task-level performance benchmarks**, **system-level stability benchmarks**, and **trace-based experiments**. Together, they provide a picture of both operational robustness and cognitive transparency.

Math reasoning benchmark: boosting GPT-oss:20b from 77% to 92%
The most recent and significant benchmark focused on mathematical reasoning with the open-source model GPT-oss:20b. As documented in docs/benchmark and summarized in the article *From 77 to 92: How OrKa-reasoning Turns GPT-oss:20b into a Math Reasoning Powerhouse*, the test measured accuracy on structured math problems.

- Baseline GPT-oss:20b performance: **77% accuracy**
- With OrKa-reasoning orchestration (agents + debate loop + agreement checking): **92% accuracy**
- Improvement: **+15 percentage points** in task-level accuracy

This gain was achieved without altering the underlying model weights. Instead, the improvement came from orchestrating multiple agents in Loops of Truth, scoring agreements, and synthesizing answers through structured convergence. This demonstrates OrKa's ability to **elevate raw model performance through cognitive scaffolding**, rather than brute-force scaling.

System benchmark: stability under load
Earlier tests confirmed that OrKa can sustain large-scale execution reliably. A 1000-run benchmark with DeepSeek-R1 1.5B logged over two thousand agent calls with **zero failures**, stable latency (~7.6 seconds average), and minimal resource usage (CPU ~88°C, <5.3 GB RAM). This proved that OrKa's orchestration layer is computationally efficient and reproducible even on consumer hardware.

Trace benchmark: reasoning diversity (SOC-02)
Complementary experiments such as SOC-02 showed OrKa coordinating progressive, conservative, realist, and purist agents. While agreement scores did not always converge, every loop was transparently logged, exposing how perspectives diverged and partially aligned. These experiments highlight the value of OrKa's traceability for qualitative reasoning analysis.

Summary
Taken together, the benchmarks show a system that is both **robust and transformative**. The math reasoning benchmark demonstrates OrKa's ability to boost task accuracy significantly. The stability benchmark proves it can operate at scale without performance collapse. The trace experiments illustrate how OrKa captures the reasoning process itself, even when convergence is incomplete. The combination positions OrKa-reasoning as a framework that not only performs reliably, but also measurably enhances reasoning quality.

4. Discussion

The benchmarks highlight a central insight: **orchestration matters as much as the model itself**. In the math reasoning benchmark, GPT-oss:20b alone produced correct answers 77% of the time. With OrKa-reasoning, accuracy rose to 92%. The model did not change — what changed was the cognitive context in which it operated. Agents debated, agreement scores were tracked, and final synthesis leveraged structured convergence. This is evidence that modular reasoning flows can transform raw model ability into more reliable outcomes. This stands in sharp contrast to the current AI ecosystem, where most frameworks still treat LLMs as monolithic black boxes. Prompt wrappers, even when sophisticated, remain one-shot solutions. They produce outputs but not reasoning traces. They succeed when the model aligns, and fail silently when it drifts. By comparison, OrKa-reasoning makes reasoning **visible, replayable, and debuggable**. The traces from SOC-02 show that even when agreement is not reached, every argument and counterargument is preserved. This shifts the evaluation focus from *answers* to *process*.

The architectural choices also point to **early signs of metacognition**. Loops of Truth simulate reflection by iterating until convergence thresholds are met. Memory nodes allow agents to recall and adapt across cycles. GraphScout, introduced in v0.9.3, pushes further by exploring alternative reasoning paths when static flows fail. These features suggest a path toward systems that are not only explainable but also capable of **reflecting on their own reasoning structure**.

Finally, OrKa-reasoning aligns with the ethical imperative for transparency in AI. When decisions are broken into modular steps, logged, and open to inspection, accountability becomes part of the runtime rather than a post-hoc audit. Users and developers can see why an answer emerged, where disagreements surfaced, and how consensus (or lack of it) was established. In an era where opaque AI outputs dominate, this design principle is not only technical but social: it makes alignment observable.

The implications of these results extend beyond narrow benchmarks. They suggest that cognitive scaffolding can **stretch the usefulness of smaller, cheaper models**. Where the dominant narrative has been to scale model size, OrKa-reasoning demonstrates that scaling orchestration can be just as impactful. This lowers the barrier for local and open-source deployments, empowering teams to achieve high-quality reasoning without depending solely on massive proprietary LLMs.

At the same time, the limitations must be acknowledged. Agreement scores are still extracted by pattern recognition and can be brittle. Loops may stall without converging, as seen in SOC-02. GraphScout is in beta and not yet a mature path-finding agent. These are reminders that OrKa-reasoning is not production-proof infrastructure. It is a **research-stage framework** with promising results, requiring further validation at scale and in more diverse domains.

Despite these caveats, the trajectory is clear. Benchmarks show measurable gains in reasoning performance, stability tests show it can run reliably at scale, and traces show reasoning paths that can be studied and improved. Together, they establish OrKa-reasoning not only as a technical toolkit but as a **new substrate for explainable and metacognitive AI**.

5. Future Work

OrKa-reasoning has proven that modular orchestration can elevate model performance, sustain large-scale execution, and expose reasoning as a traceable process. The next stage of development focuses on strengthening its foundations and extending its cognitive capabilities.

Scaling and infrastructure
Current backends rely primarily on Redis for memory and logging, with Kafka support emerging. Future work will expand this dual-mode architecture to handle higher throughput, multi-tenant deployments, and integration with cloud-native infrastructures. This will ensure that OrKa can scale beyond local experiments into production environments without losing traceability.

Richer convergence metrics
Agreement scoring today depends on pattern extraction and thresholds. Upcoming iterations will explore embedding-based consensus metrics, probabilistic scoring, and hybrid evaluation nodes. The goal is to make convergence more reliable, capturing not only surface agreement but also deeper semantic alignment.

Adaptive reasoning strategies
GraphScout introduced the first step toward self-discovery of reasoning paths. Future work will extend this capability so that agents can select, adapt, and even generate orchestration graphs dynamically at runtime. This will move OrKa from static YAML flows toward adaptive cognition, where strategies evolve with the task.

Knowledge integration
Service nodes will be expanded to support direct integration with structured knowledge bases, external APIs, and domain-specific embeddings. This will allow OrKa-reasoning to combine generative debate with factual grounding, bridging the gap between synthetic reasoning and reliable knowledge retrieval.

Toward metacognition
Longer-term work focuses on making OrKa not just explainable but reflective. This means enabling agents to monitor their own reasoning efficiency, compare strategies across runs, and adjust their approaches over time. The foundation is already present in loops, memory layers, and GraphScout. The next step is to turn these features into a coherent metacognitive layer, where reasoning about reasoning becomes a runtime capability.

Ethical and social positioning
Future development will also emphasize auditability and alignment. OrKa's modular design makes it possible to expose every decision, log every disagreement, and explain every convergence. Building on this, the roadmap includes tools for ethical auditing, fairness checks, and policy integration, making transparency not optional but intrinsic to the runtime.

6. Conclusion

OrKa-reasoning demonstrates that intelligence does not have to emerge from a single black-box model. By orchestrating multiple agents, layering memory, and running iterative Loops of Truth, the framework turns reasoning into a process that is transparent, auditable, and improvable. Benchmarks confirm its promise: accuracy gains from 77% to 92% in math reasoning, stable performance across 1000 orchestrated runs, and qualitative traces that expose how different perspectives interact.

This positions OrKa-reasoning as more than an orchestration toolkit. It is a **cognitive substrate** where reasoning is not hidden but lived out through explicit steps, agreements, and disagreements. Inspired by Minsky's Society of Mind, it shows early signs of metacognition, from iterative self-correction to exploratory path discovery with GraphScout.

The framework remains research-stage. Agreement scores are brittle, convergence is not guaranteed, and adaptive reasoning is still emerging. Yet even in this state, OrKa-reasoning provides something scarce in today's AI landscape: **a system that shows its work**. In a field dominated by opaque outputs, OrKa's approach makes cognition observable and alignment a runtime property rather than an afterthought.

Looking ahead, OrKa-reasoning offers a path toward AI systems that are not only more capable but also more trustworthy. By combining robustness with explainability and performance with transparency, it sketches the outlines of a future where artificial cognition can be studied, guided, and trusted in ways single-model prompting cannot deliver.

APPENDIX

```
orchestrator:
  id: cognitive_society_simple
  strategy: sequential
  agents:
    - simple_debate_loop
    - final_answer

agents:
  - id: simple_debate_loop
    type: loop
    max_loops: 3
    score_threshold: 0.8
    score_extraction_config:
      strategies:
        - type: pattern
          patterns:
            - 'AGREEMENT_SCORE[":]?\s*([0-9.]+)'
            - '"AGREEMENT_SCORE":\s*([0-9.]+)'
    internal_workflow:
      orchestrator:
        id: simple_debate_internal
        strategy: sequential
        agents:
          - progressive_agent
          - conservative_agent
          - agreement_check
      agents:
        - id: progressive_agent
          type: local_llm
          model: gpt-oss:20b
          model_url: http://localhost:11434/api/generate
          provider: ollama
          temperature: 0.7
          prompt: |
            ROLE: Progressive
            TASK: Argue from a progressive stance
            INPUT: {{ get_input() }}
        - id: conservative_agent
          type: local_llm
          model: gpt-oss:20b
          model_url: http://localhost:11434/api/generate
          provider: ollama
          temperature: 0.7
          prompt: |
            ROLE: Conservative
            TASK: Argue from a conservative stance
            INPUT: {{ get_input() }}
        - id: agreement_check
          type: local_llm
          model: gpt-oss:20b
          model_url: http://localhost:11434/api/generate
          provider: ollama
          temperature: 0.2
          prompt: |
            Evaluate agreement between these positions.
            USER_ASKED: {{ get_input() }}
            PROGRESSIVE: {{ get_agent_response('progressive_agent') }}
            CONSERVATIVE: {{ get_agent_response('conservative_agent') }}
            Return AGREEMENT_SCORE: <0.0-1.0>
  - id: final_answer
    type: local_llm
    model: gpt-oss:20b
    model_url: http://localhost:11434/api/generate
    provider: ollama
    temperature: 0.3
    prompt: |
      USER_ASKED: {{ get_input() }}
      Debate summary
      - loops_completed: {{ get_agent_response('simple_debate_loop')['loops_completed'] if get_agent_response('simple_debate_loop')
else 'Unknown' }}
      - final_score: {{ get_agent_response('simple_debate_loop')['final_score'] if get_agent_response('simple_debate_loop') else
'Unknown' }}
      - agreement_reached: {{ (get_agent_response('simple_debate_loop')['final_score'] >= 0.8) if
get_agent_response('simple_debate_loop') and get_agent_response('simple_debate_loop')['final_score'] else False }} Produce a concise
answer that reflects the debate outcomes. If agreement_reached is False, highlight the strongest arguments from each side and explain
what evidence would close the gap.
```