Paper 1: UCA — Why Your AI Needs ADHD (And Multiple Personalities)

Author: [your local neurodivergent doomer transfem pup]
Profiles: my GitHub | my HuggingFace
Date: July 2026
License: CC-BY-SA 4.0 (because intellectual property is a spook and we should share knowledge like comrades)

Abstract

Look, I'm gonna be real with you. Current AI systems are boring. They have one mode — usually some variant of "predict the next token left-to-right like a very expensive autocomplete" — and they use it for everything. Whether you're asking it to solve a math problem, write a poem, or diagnose your probably-fatal rash, it does the exact same thing. That's not intelligence. That's a very large parrot with a gambling addiction.

This paper introduces the Unified Cognitive Architecture (UCA): a framework that gives your AI the same thing every neurodivergent person already has — multiple ways of thinking, the ability to switch between them, and the self-awareness to know when you're bullshitting. UCA integrates six distinct reasoning paradigms into a single adaptive pipeline: structural reasoning (CoT/ToT/GoT), logical frameworks (deductive/inductive/abductive/analogical), cognitive modes (System 1 fast thinking vs System 2 slow thinking, plus metacognition and counterfactuals), latent reasoning (thinking in compressed neural space instead of words), self-refinement (checking your own work like a good student), and domain-specific tools (code execution, theorem proving). The key innovation is the Framework of Thought (FoT): a dynamic router that selects the right reasoning mode for the right problem, just like how I switch between hyperfocus and dissociation depending on whether the task is interesting or capitalism is trying to kill me again.

Keywords: Chain-of-Thought, Tree-of-Thoughts, latent reasoning, metacognition, adaptive compute, cognitive architecture, ADHD but make it AI

---

1. Introduction: The Monolithic Reasoning Problem (Or Why Your LLM Is Boring)

So here's the thing. Since Wei et al. (2022) dropped Chain-of-Thought prompting, the field has been absolutely flooded with reasoning techniques. We've got Tree-of-Thoughts (Yao et al., 2023), Graph-of-Thoughts (Besta et al., 2024), latent reasoning through hidden states (Hao et al., 2024), self-refining mechanisms like Reflexion (Shinn et al., 2023) and Chain-of-Verification (Dhuliawala et al., 2023), and about seventeen other variants that I'm not going to list because my ADHD meds are wearing off.

But here's the kicker: **production AI systems still default to a single reasoning modality.** Usually it's just basic CoT. Sometimes they don't even do that — they just vibes-based generate and hope for the best. It's like hiring a carpenter and finding out they only own a hammer, and they use it for everything including brain surgery.

This is cognitively implausible. Humans — especially neurodivergent humans, but honestly all humans

— don't think in one mode. Kahneman's dual-process theory (Kahneman, 2011) literally proved that we have System 1 (fast, intuitive, pattern-matching, low effort) and System 2 (slow, deliberate, analytical, high effort). You don't carefully deliberate over whether to put on pants in the morning. You also don't intuitively guess the answer to a calculus exam. You *select* the appropriate mode based on familiarity, complexity, and consequence.

Current LLMs lack this selection mechanism. They're like a person who only has System 1 and uses it for everything, including performing surgery. No wonder they hallucinate.

1.1 The Synthesis Imperative (Or: Stop Inventing New Hammers, Start Building a Toolbox)

The central thesis of this paper is that the next leap in AI capability won't come from inventing a seventh or eighth reasoning technique. It'll come from **orchestrating** the existing six into a coherent, adaptive system that knows when to use which tool.

Each paradigm — structural, logical, cognitive, latent, self-refining, and domain-specific — addresses a specific weakness in the others. CoT is transparent but slow as hell. Latent reasoning is fast but opaque. System 1 is efficient but will absolutely lie to your face. System 2 is accurate but takes forever and costs a fortune. Deductive logic is rigorous but breaks if your premises are wrong. Abductive reasoning is creative but unverified. They're not competitors. They're **complementary tools in a unified cognitive toolkit**.

UCA's primary contribution is the Framework of Thought (FoT): a dynamic router that selects and blends reasoning modes based on real-time metacognitive assessment. Think of it as the AI's executive function — the thing that neurodivergent people often struggle with, but when it works, it *works*.

---

2. The Six Reasoning Paradigms: A Taxonomy of Complementary Modes

2.1 Structural / Path-Based Reasoning (Explicit)

These methods structure reasoning as traversable paths through a representational space. They're the classic "think step by step" techniques.

Chain of Thought (CoT) — Sequential, step-by-step reasoning where each step follows from the previous (Wei et al., 2022). Great for well-structured problems with deterministic intermediate states. Terrible for anything that requires backtracking or exploration. It's like walking down a hallway with no doors.

Tree of Thoughts (ToT) — Branching exploration where the model evaluates multiple reasoning paths simultaneously, pruning dead ends and backtracking (Yao et al., 2023). Optimal for search problems, games, and creative tasks. Downside: computationally expensive because you have to explore branches. It's like trying every door in a mansion instead of just walking down the hallway.

Graph of Thoughts (GoT) — Non-linear reasoning where thoughts can merge, converge, and loop (Besta et al., 2024). Great for synthesis tasks where multiple sub-problems interact. Downside: requires careful aggregation to avoid merging conflicting ideas into a beautiful nonsense salad.

Framework of Thought (FoT) — *Our contribution.* A dynamic blueprint that switches between Chain, Tree, or Graph mid-reasoning based on problem structure. The router evaluates each sub-problem's properties (linearity, branching factor, convergence requirements) to select the appropriate topology. It's like having a GPS that actually knows when to take the highway and when to explore back roads.

Skeleton of Thought (SoT) — High-level outline generation followed by point-by-point elaboration (Ning et al., 2023). Great for long-form generation where structure precedes content. Downside: if your initial skeleton is garbage, everything else is garbage too.

ReAct (Reason + Act) — Interleaves reasoning with tool-calling: think → act → observe → think again (Yao et al., 2022). Essential for tasks requiring external knowledge or computation. Downside: tool latency can disrupt reasoning flow, and if your tool is down, you're just sitting there thinking about how broken everything is.

2.2 Logical / Epistemological Reasoning (The "Why")

These frameworks determine *how* conclusions are justified. They're the philosophy major of the AI world.

Deductive reasoning — Top-down logic: general rule → specific conclusion. If premises are true and logic is valid, the conclusion is necessarily true. Essential for math, formal verification, and legal syllogisms. Weakness: requires true premises. Garbage in, garbage out. It's very confident garbage, though.

Inductive reasoning — Bottom-up: specific observations → general pattern. Essential for scientific discovery, pattern recognition, and statistical inference. Weakness: conclusions are probabilistic, not certain. But honestly, in this economy, what is certain?

Abductive reasoning — Inference to the best explanation: limited data → plausible hypothesis. Essential for diagnosis, troubleshooting, and creative problem-solving. Weakness: the "best" explanation might still be completely wrong. But it's usually wrong in an interesting way.

Analogical reasoning — Transfer of knowledge from one domain to another based on structural similarity (Gentner, 1983). Essential for innovation and cross-domain learning. Weakness: similarity judgments can be superficial, leading to false analogies. Like comparing capitalism to a free market — technically related but one is a death cult and the other is an economic model.

2.3 Cognitive / System-Level (Human-Inspired)

These modes control *how* processing resources are allocated. They're the ADHD medication of the architecture.

System 1 (Fast/Intuitive) — Quick, pattern-recognition-based responses drawing on cached associations and heuristics. Low computational cost. Prone to bias and hallucination when patterns are misleading. It's the "vibes-based" reasoning mode.

System 2 (Slow/Deliberate) — Deep, analytical, step-by-step verification. High computational cost. Avoids hallucinations through explicit verification but is slow and resource-intensive. It's the "actually read the instructions" mode.

Metacognition — "Thinking about thinking." The system plans its reasoning scaffold, evaluates its own confidence, and selects strategies before executing them (Flavell, 1979). In UCA, metacognition is not an afterthought — it's the primary control mechanism. It's the executive function that decides whether to use System 1 or System 2, whether to explore or exploit, whether to keep going or ask for help.

Counterfactual reasoning — Reasoning about what did not happen: "If I had chosen X, what would have occurred?" Essential for robustness, regret minimization, and revealing hidden assumptions (Byrne, 2002). It's the "what if?" engine that keeps you up at night.

2.4 Latent / Implicit Reasoning (The New Frontier)

These methods bypass the linguistic bottleneck by reasoning in compressed neural representations. They're the "I can't explain why I know this, I just know" mode.

Latent Thought Flow (LTF) — Hard reasoning occurs silently in compressed neural activations before text output (Hao et al., 2024). The model does not "think in words" but in high-dimensional vector states. It's like having thoughts that are too complex to verbalize.

Selective Latent Thinking — Only critical precision steps are written as text; the bulk of computation occurs in hidden states. This creates a hybrid explicit/latent pipeline. It's like only speaking when you have something important to say.

Chain of Latent Thoughts (CoLT) — Multi-modal reasoning through latent representations without translation to language. Enables reasoning across text, image, audio, and structured data in a unified representational space. It's the "thinking in pictures and feelings" mode.

2.5 Self-Refining & Verification

These mechanisms improve answer quality through iteration and checking. They're the "did I actually turn off the stove?" loop.

Self-Consistency — Running the same logic multiple times with sampling and voting on the majority answer (Wang et al., 2022). Simple but effective. Cost scales linearly with samples, which in a capitalist economy means it's expensive.

Reflexion — The model receives feedback on its output, identifies the mistake, and re-runs the thought process to fix it (Shinn et al., 2023). Requires an evaluation function or external feedback. It's the "learn from your mistakes" mechanism.

Chain of Verification (CoVe) — The model fact-checks its own previous reasoning steps against external knowledge or internal consistency (Dhuliawala et al., 2023). Particularly effective for factual hallucinations. It's the "wait, let me Google that" mechanism.

2.6 Specialized Domain Thinking

Program of Thoughts (PoT) — Solves mathematical and logical problems by writing and executing actual code rather than guessing numbers (Chen et al., 2022). Eliminates arithmetic errors but requires a code execution environment. It's the "use a calculator" mode, but make it fancy.

Theorem-of-Thought (ToTh) — Uses multiple parallel agents to simulate abductive, deductive, and inductive reasoning simultaneously to solve complex theorem problems (Saha et al., 2024). A precursor to the multi-agent architectures we'll discuss in Paper 2.

2.7 The Interdependency Matrix (Or: How These Tools Cover Each Other's Asses)

Component	Primary Weakness	Compensated By	How
CoT	Slow, linear	ToT/GoT + LTF	Branch when stuck; compress bulk work

Generated by Kimi.ai

Table 1 – continued

Component	Primary Weakness	Compensated By	How
ToT	Expensive, wanders	FoT + System 1	Fast pattern-match prunes dead branches
Deductive	Brittle premises	Abductive + Inductive	Generate premises; test with data
System 1	Biased, hallucinates	System 2 + CoVe	Auto-escalate when confidence drops
Latent	Opaque, un-debuggable	Explicit CoT + Reflexion	Expand latent → text at check-points
PoT	Needs code env	ReAct	Seamless tool integration
Self-Consistency	Expensive (N runs)	Selective Latent	Compress cheap runs; expand winners

---

3. The UCA Architecture: From Atoms to Organisms

UCA is a **fractal architecture** — the same control pattern repeats at every scale. This is important because fractals are beautiful and nature is a communist (it shares patterns across scales without charging royalties).

3.1 Layer 0: The Reasoning Atom

The Reasoning Atom is the smallest unit of cognition. Every reasoning step passes through:

```
Input → [Mode Selector] → [Processor] → [Latent Compressor] → [Verifier] → Output
```

The Mode Selector asks three questions:

1. What structure fits? (Chain for linear, Tree for exploratory, Graph for convergent)

2. What logic applies? (Deductive for math, Abductive for diagnosis, Analogical for novel problems)
3. How much effort? (System 1 for familiar patterns, System 2 for novel/complex)

Even a simple question like "What's 17 + 25?" goes through this atom. The Mode Selector recognizes low complexity, routes to System 1 with optional PoT verification, and outputs "42" instantly. No need to write a five-paragraph essay about addition.

3.2 Layer 1: The Cognitive Node

A Cognitive Node chains multiple Atoms to solve one complex problem in four phases:

Phase 1: Metacognitive Scaffolding (System 2)

- Skeleton of Thought (SoT): Generate a high-level outline
- Framework of Thought (FoT): Select reasoning topology
- Confidence Assessment: "Do I actually know this or am I about to bullshit?"

Phase 2: Active Reasoning (Mixed Modes)

- Fast Path (System 1): Pattern-match against cached solutions
- Slow Path (System 2): If confidence is low, activate ToT exploration
- Logical Verification: Apply Deductive check to math; Inductive check to patterns
- Latent Processing (LTF): Do the heavy lifting in compressed space; emit text only for key steps

Phase 3: Self-Refinement

- Self-Consistency: Run 3 parallel reasoning paths
- Chain of Verification (CoVe): Fact-check each critical claim
- Reflexion: If inconsistency detected, loop back to Phase 2 with corrected premise

Phase 4: Output Synthesis

- Graph of Thoughts: Merge convergent paths
- Final confidence score + reasoning trace for human auditability

3.3 Layer 2: The Domain Specialist

For specialized domains (math, coding, legal reasoning), we add Domain Specialist Nodes with parallel tracks:

- **Track A (Deductive):** Formal logic chain
- **Track B (PoT):** Write and execute code
- **Track C (Abductive):** Guess the solution structure before proving it
- **Track D (Analogical):** Retrieve similar solved problems

Convergence via Graph of Thoughts. Verification via CoVe. If PoT and Deductive agree, confidence is high. If they disagree, the Verifier investigates.

Generated by Kimi.ai

3.4 Layer 3: The Meta-Cognitive Agent

The complete UCA agent is a cognitive operating system with five stacked layers:

Orchestration Layer (FoT + Metacognition): Monitors task complexity, selects topology, allocates compute budget, decides latent vs explicit ratio, triggers counterfactuals for high-stakes decisions.

Reasoning Layer (Explicit + Latent): CoT → ToT → GoT on the explicit side; LTF → Selective Latent → CoLT on the latent side. Hybrid: text for precision, latent for bulk work.

Logical Layer (Epistemological Engine): Proof → Deductive; Pattern discovery → Inductive; Diagnosis → Abductive; Novel domain → Analogical; High stakes → All four in parallel (ToTh style).

Cognitive Layer (Human-Inspired Control): System 1 for speed; System 2 for accuracy; Counterfactual for robustness.

Verification Layer (Self-Refining): Self-Consistency, Reflexion, CoVe, and ReAct integration.

Domain Layer (Specialized Modules): PoT for code, ToTh for theorems, domain-specific analogical databases.

---

4. How to Actually Build This (Implementation Roadmap)

Phase 1: The Scaffold (Weeks 1–4)

Build the FoT Mode Router. Train a lightweight classifier to assess task properties:

- Complexity score (1–10)
- Domain classification (math, coding, creative, factual, reasoning)
- Uncertainty estimate (confidence that System 1 can handle it)

Pseudocode:

```
def reason(input_problem):
    complexity = assess_complexity(input_problem)
    domain = classify_domain(input_problem)
    uncertainty = estimate_uncertainty(input_problem)

    if complexity <= 3 and uncertainty < 0.3:
        return system1_quick_response(input_problem)

    if domain == "mathematics":
        return program_of_thoughts(input_problem)

    if uncertainty > 0.7:
        return tree_of_thoughts_exploration(input_problem)

    return hybrid_latent_cot(input_problem)
```

Expected outcome: 15–30% improvement on mixed-task benchmarks by stopping the model from using CoT for simple questions.

How to fine-tune an existing model: Take a capable base model (Llama 3, Qwen 2.5, DeepSeek-V3). Add a small router head (2-layer MLP) on top of the final hidden state. Fine-tune on a dataset of

problems labeled with optimal reasoning mode. The router learns to predict which mode to use. Freeze the base model initially, then do LoRA fine-tuning on the full model with mode-specific adapters.

Phase 2: The Loop (Weeks 5–12)

Add self-refinement mechanisms.

1. Generate initial answer via Phase 1 routing.
2. Extract factual claims using structured parsing.
3. Verify against internal knowledge (RAG) and external tools (ReAct).
4. If contradiction: tag the step, backtrack, regenerate with corrected premise.
5. Run self-consistency: 3 parallel attempts with different temperatures.

How to implement: Use an existing LLM with tool-calling capabilities. Add a verification loop that calls the model twice: once to generate, once to verify. For CoVe, use a separate verification prompt that asks the model to fact-check each claim. For Reflexion, maintain a memory buffer of past mistakes and their corrections.

Phase 3: The Latent Bridge (Weeks 13–24)

Integrate implicit reasoning.

- **LTF:** Train the model to output `<latent>` special tokens where heavy reasoning occurs in hidden states. Use intermediate latent supervision from synthetic datasets.
- **Selective Latent Thinking:** Implement a gating mechanism that decides per reasoning step whether to emit text or remain in latent space.
- **CoLT:** For multi-modal tasks, project all modalities into a shared latent space and reason through cross-modal attention.

How to implement: This is the hard part. You need to modify the model's training objective to include latent consistency losses. One approach: train the model with a "latent reconstruction" task where the model must predict its own hidden state at the next layer. Another approach: use DeepSeek-R1-style reasoning traces but compress them into latent vectors rather than text tokens. Distill from a larger teacher model that generates explicit reasoning traces, but train the student to reproduce the teacher's final hidden states rather than its text outputs.

Expected outcome: 2–5x speedup on reasoning tasks without accuracy loss.

Phase 4: The Full Orchestra (Months 6–12)

Deploy multi-agent Theorem-of-Thought. Multiple specialized instances operating in parallel:

- The Deducer (formal logic, math proofs)
- The Inducer (pattern finder, data scientist)
- The Abducer (hypothesis generator, creative problem solver)
- The Analogist (cross-domain transfer expert)
- The Verifier (skeptic, fact-checker, devil's advocate)

The FoT Orchestrator routes sub-problems to the right agent and merges results via Graph of Thoughts.

How to implement: Use a framework like LangChain, AutoGen, or CrewAI, but replace the default LLM calls with UCA-enabled agents. Each agent is a separate instance of your fine-tuned model with different system prompts and tool configurations. The orchestrator is a lightweight router model or a rule-based system that parses the problem and assigns it to the right agent.

---

5. Next-Gen Architecture Ideas (That Don't Exist Yet But Should)

5.1 The Quantum Reasoning Router

Current FoT routers are classical classifiers. What if we used quantum-inspired superposition? The router maintains a superposition of reasoning modes until the problem is sufficiently observed, then collapses to the optimal mode. This isn't actual quantum computing — it's a classical analogy using mixture-of-experts with quantum-inspired gating. But it sounds cool and might actually help with uncertainty quantification.

5.2 Neuromorphic System 1/2 Switching

Instead of a binary System 1/System 2 switch, use neuromorphic hardware (like Intel Loihi or IBM TrueNorth) to implement analog threshold-based switching. System 1 is the default low-power state. When uncertainty exceeds a threshold, the neuromorphic chip triggers a digital System 2 deep dive. This would be incredibly energy-efficient and would make the AI feel more… alive.

5.3 The Communist Compute Pool

In a capitalist cloud, each agent pays per token. In a socialist architecture, compute is pooled and allocated based on need. The Cluster Hub acts as a central planner (yes, I said it) distributing GPU cycles to agents based on task priority, not wallet size. This is technically just a resource scheduler, but calling it a Communist Compute Pool makes it sound way more interesting.

---

6. Conclusion: The Future Is Adaptive (And Probably Neurodivergent)

The Unified Cognitive Architecture represents a shift from monolithic to adaptive reasoning. By treating Chain-of-Thought, Tree-of-Thoughts, latent reasoning, dual-process cognition, logical frameworks, and self-refinement as complementary tools in a unified toolkit, UCA enables AI systems to reason with appropriate depth, speed, and rigor for each task.

The architecture is fractal, implementable, and grounded in cognitive science. It starts with a simple mode router and scales to a full multi-agent orchestra. And honestly? It just makes sense. If you have one tool, you use it for everything and it breaks. If you have a toolbox, you pick the right tool and everything works better.

The future of AI reasoning is not a longer chain of thought. It's a mind that knows when to think fast, when to think slow, when to think in words, when to think in silence, and when to question its own conclusions.

Also, capitalism is a death cult and we should build AI that serves humanity, not shareholders. But that's a different paper.

Generated by Kimi.ai

References

Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Podstawski, M., Gianinazzi, L., … & Hoefler, T. (2024). Graph of Thoughts: Solving elaborate problems with large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16), 17682–17690.

Byrne, R. M. (2002). Mental models and counterfactual thoughts about what might have been. *Trends in Cognitive Sciences*, 6(10), 426–431.

Chen, W., Ma, X., Wang, X., & Cohen, W. W. (2022). Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Dhuliawala, S., Komeili, M., Xu, J., Raileanu, R., Li, X., Celikyilmaz, A., & Weston, J. (2023). Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*.

Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry. *American Psychologist*, 34(10), 906–911.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2), 155–170.

Hao, S., Sukhbaatar, S., Su, D., Li, X., Hu, Z., Weston, J., & Tian, Y. (2024). Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*.

Kahneman, D. (2011). *Thinking, fast and slow*. Farrar, Straus and Giroux.

Ning, X., Lin, Z., Li, H., Yang, Z., Hu, S., Zhou, Y., … & Wang, Y. (2023). Skeleton-of-thought: Large language models can do parallel decoding. *arXiv preprint arXiv:2307.15337*.

Saha, S., Kotha, S., & Cherian, J. J. (2024). Theorem of thoughts: A framework for mathematical reasoning with large language models. *arXiv preprint arXiv:2404.12618*.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Self-reflective agents with verbal reinforcement learning. *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., … & Zhou, D. (2022). Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., … & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 35, 24824–24837.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. *ICLR 2023*.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., & Narasimhan, K. (2023). Tree of thoughts: Deliberate problem solving with large language models. *NeurIPS*, 36.