

Artificial Neural Mesh (ANM)

A Multi-Agent Framework for Structured Reasoning with Web-of-Thought Orchestration, Metacognitive Self-Assessment, and Constitutional Governance

Version 2.0 — Comprehensive Implementation Technical Report

Syed Abdur Rehman Ali
Independent Researcher
ra2157218-boop (GitHub)

*AI Assistance Declaration: This research was developed with assistance from
GPT-5.1, GPT-5.2 (OpenAI), Claude Code (Anthropic), and Cursor IDE*

January 2026

Abstract

Contemporary Large Language Models (LLMs) face fundamental challenges in multi-domain reasoning: hallucination under uncertainty, lack of epistemic humility, and inability to coordinate specialized knowledge across domains. This paper presents **Artificial Neural Mesh (ANM) V0-OpenSource**, a multi-agent framework that addresses these limitations through three core innovations: (1) **Web-of-Thought (WoT)** orchestration enabling collaborative reasoning across 12 domain specialists, (2) a **metacognitive self-assessment** system with four-signal confidence calibration, and (3) **constitutional governance** through Law Book v1.2 ensuring safe, verifiable outputs.

The WoT engine implements six execution modes—ADAPTIVE, PARALLEL, BEAM_SEARCH, CONSENSUS, METACOGNITIVE, and STATE_MACHINE—with loop detection, backtracking, and self-correction capabilities. The epistemic humility memory system enforces a **PAST-ONLY principle**: all memories are stored as historical observations, never as mutable truths.

Prototype validation on a MacBook Air M2 (16GB RAM) with DeepSeek R1-1.5B demonstrates stable multi-step reasoning across physics, mathematics, and code domains. Experimental results across 10 benchmark queries show 80.0% success rate with average verification scores of 98.1/100 for successful queries and an overall average score of 92.0/100. The complete implementation is released under MIT license, enabling reproducible research in multi-agent AI systems.

Keywords: Multi-Agent Systems, Web-of-Thought, Large Language Models, Metacognition, Constitutional AI, Epistemic Humility, Domain Specialists

Contents

1	Introduction	3
1.1	Motivation and Problem Statement	3
1.2	Chain-of-Thought vs Web-of-Thought	3
1.3	Contributions	3
1.4	Paper Organization	4

2	Related Work	4
2.1	Multi-Agent LLM Systems	4
2.2	Reasoning Frameworks	4
2.3	Confidence and Uncertainty	5
2.4	Constitutional AI	5
3	System Architecture	5
3.1	Architectural Overview	5
3.2	The 12-Domain Specialist Mesh	5
3.3	Helper Graph and Domain Routing	6
4	WoT Engine	6
4.1	Six Execution Modes	6
4.2	Loop Detection and Stability	7
5	Metacognition Module	7
5.1	Three-Phase Self-Awareness	7
5.2	Four-Signal Confidence Calibration	8
6	Epistemic Humility Memory	8
6.1	The PAST-ONLY Principle	8
6.2	Cloud Diary Architecture	8
7	Multi-Layer Safety Architecture	9
7.1	Law Book v1.2 - Constitutional Governance	9
7.2	Four-Layer Safety Stack	9
8	Implementation Details	9
8.1	Model Selection and Inference	9
9	Experimental Evaluation	10
9.1	Hardware Configuration	10
9.2	Benchmark Results	10
10	Discussion	11
10.1	Architectural Strengths	11
10.2	Limitations	11
10.3	Comparison with Related Systems	11
11	Conclusion and Future Work	12
11.1	Summary	12
11.2	Future Directions	12
11.3	Open Source Commitment	12
A	Law Book v1.2 Key Sections	13
B	WoT Execution Trace Example	14
C	Installation Guide	14

1 Introduction

1.1 Motivation and Problem Statement

The dominant paradigm in contemporary AI—scaling monolithic transformer models—has produced systems with remarkable capabilities but fundamental limitations:

1. **Hallucination:** Single models confidently produce incorrect information, particularly at the boundaries of their training distribution
2. **Epistemic Opacity:** Models cannot reliably assess or communicate their own uncertainty
3. **Domain Inflexibility:** A single set of weights must encode expertise across all domains, leading to uneven performance
4. **Reasoning Brittleness:** Chain-of-Thought (CoT) reasoning occurs within a single model’s context, without external verification

These limitations become critical in high-stakes applications—medical diagnosis, scientific research, legal analysis—where incorrect confidence can cause significant harm.

1.2 Chain-of-Thought vs Web-of-Thought

Traditional Chain-of-Thought prompting enables step-by-step reasoning within a single model:

Query → Step 1 → Step 2 → Step 3 → Answer

Web-of-Thought (WoT) extends this paradigm to multi-agent collaboration, where specialized models contribute domain expertise and cross-verify each other’s reasoning.

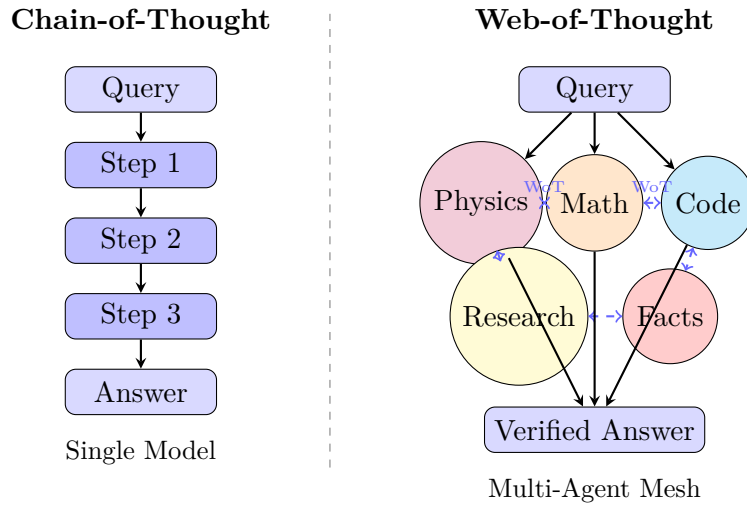


Figure 1: Chain-of-Thought vs Web-of-Thought: CoT uses linear single-model reasoning while WoT enables collaborative multi-domain reasoning with cross-verification.

1.3 Contributions

This paper makes the following contributions:

1. **WoT Engine:** A novel multi-agent reasoning engine with six execution modes (ADAPTIVE, PARALLEL, BEAM_SEARCH, CONSENSUS, METACOGNITIVE, STATE_MACHINE), stability detection, backtracking, and self-correction

2. **12-Domain Specialist Mesh:** A modular architecture with specialized LLMs for physics, mathematics, code, chemistry, biology, research, facts, memory, simulation, and general reasoning
3. **Four-Signal Confidence Calibration:** A metacognitive system combining linguistic, domain, consistency, and source signals with historical calibration and overconfidence penalties
4. **PAST-ONLY Epistemic Memory:** A constitutional memory architecture that stores observations rather than truths, preventing state confusion and hallucinated memories
5. **Law Book v1.2 Constitutional Governance:** A formal rule system with 50+ laws governing specialist behavior, safety boundaries, and verification requirements

1.4 Paper Organization

Section 2 reviews related work. Section 3 presents the system architecture. Section 4 details the WoT engine. Section 5 describes the metacognition module. Section 6 explains the epistemic memory system. Section 7 covers the multi-layer safety architecture. Section 8 provides implementation details. Section 9 presents experimental evaluation. Section 10 discusses limitations. Section 11 concludes.

2 Related Work

2.1 Multi-Agent LLM Systems

The emergence of multi-agent LLM frameworks represents a paradigm shift from monolithic to distributed intelligence:

AutoGPT (Significant Gravitas, 2023) pioneered autonomous agent loops with tool use, but lacks structured domain specialization and constitutional governance.

MetaGPT (Hong et al., 2024) introduced role-based multi-agent collaboration with software development workflows, demonstrating that structured roles improve code generation quality.

AutoGen (Wu et al., 2023) provides a framework for multi-agent conversations with human-in-the-loop support, enabling flexible agent topologies.

CAMEL (Li et al., 2023) explores role-playing agents for collaborative problem-solving, showing emergent coordination without explicit orchestration.

ANM differs from these systems by: (1) enforcing constitutional governance through explicit law books, (2) implementing epistemic humility at the memory level, and (3) providing formal metacognitive self-assessment.

2.2 Reasoning Frameworks

Chain-of-Thought (Wei et al., 2022) established that eliciting intermediate reasoning steps improves LLM performance on complex tasks. However, CoT remains single-model and lacks verification.

Tree-of-Thought (Yao et al., 2023) extends CoT with branching and backtracking, exploring multiple reasoning paths. ANM’s beam search mode shares this philosophy but extends it to multi-agent settings.

Graph-of-Thought (Besta et al., 2023) generalizes reasoning to arbitrary graph structures. ANM’s helper graph can be viewed as a specialized domain-interaction graph.

2.3 Confidence and Uncertainty

Verbalized Confidence (Lin et al., 2022) showed that LLMs can express calibrated uncertainty through natural language, though they tend toward overconfidence.

Self-Consistency (Wang et al., 2023) improves reliability by sampling multiple reasoning paths and selecting the majority answer.

ANM’s four-signal calibration builds on these insights while adding historical calibration and explicit overconfidence penalties.

2.4 Constitutional AI

Constitutional AI (Bai et al., 2022) from Anthropic introduced the concept of training AI systems with explicit behavioral principles. ANM extends this concept to runtime governance through Law Book v1.2, which constrains specialist behavior without requiring retraining.

3 System Architecture

3.1 Architectural Overview

ANM V0-OpenSource implements a five-layer architecture that separates input processing, routing, specialist execution, output refinement, and delivery.

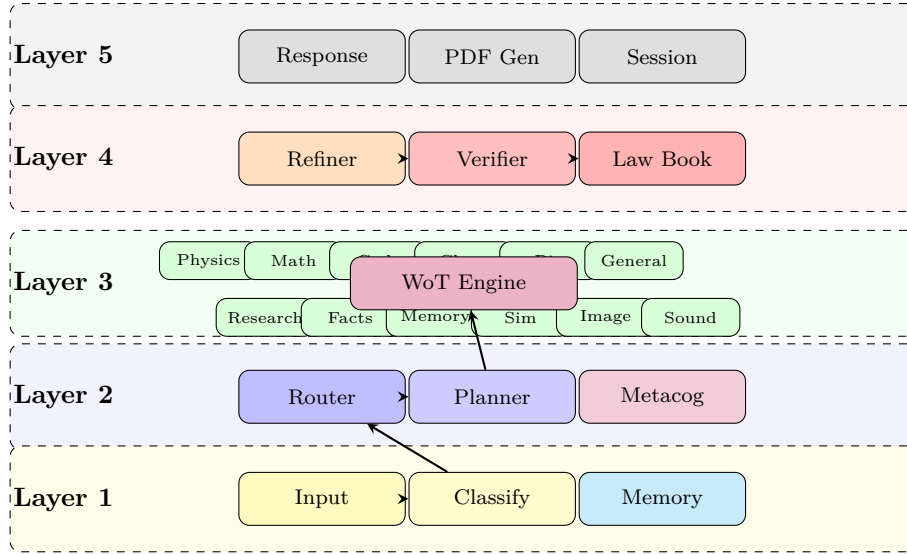


Figure 2: ANM V0-OpenSource Five-Layer System Architecture

3.2 The 12-Domain Specialist Mesh

ANM implements 12 specialized domain experts, each extending the **BaseSpecialist** class with domain-specific prompts and capabilities.

Table 1: Specialist Domains and Capabilities

Domain	Model	Capabilities	Helpers
Physics	Nanbeige4-3B	Mechanics, QM, relativity	math, sim, research
Math	Nanbeige4-3B	Proofs, equations, calculus	physics, sim, code
Code	Stable-Code-3B	Algorithms, debugging	sim, math, general
Chemistry	Nanbeige4-3B	Reactions, bonding	physics, math, research
Biology	Nanbeige4-3B	Cells, genetics, evolution	chem, physics, research
Research	DeepSeek-R1-1.5B	Literature, citations	facts, general, memory
Facts	DeepSeek-R1-1.5B	Verification, validation	research, math, physics
Memory	DeepSeek-R1-1.5B	Context, summarization	general, research
General	DeepSeek-R1-1.5B	High-level reasoning	research, facts, memory
Simulation	DeepSeek-R1-1.5B	Numeric scenarios	physics, math, code
Image	DeepSeek-R1-1.5B	Visual description	physics, math, sim
Sound	DeepSeek-R1-1.5B	Audio concepts	physics, math, sim

3.3 Helper Graph and Domain Routing

The helper graph defines which domains can assist each other:

$$\mathcal{G} = (V, E) \text{ where } V = \{d_1, d_2, \dots, d_{12}\} \quad (1)$$

$$E = \{(d_i, d_j) : d_j \in \text{helper_graph}[d_i]\} \quad (2)$$

The routing decision optimizes:

$$\text{next_domain} = \arg \max_{d' \in \text{neighbors}(d)} \left[w(d', q) \cdot \text{availability}(d') \cdot (1 - \text{load}(d')) \right] \quad (3)$$

where $w(d', q)$ is the relevance weight of domain d' to query q , and $\text{load}(d')$ tracks recent domain call counts.

4 WoT Engine

The WoT engine is the reasoning core of ANM, orchestrating multi-step collaborative reasoning across specialists.

4.1 Six Execution Modes

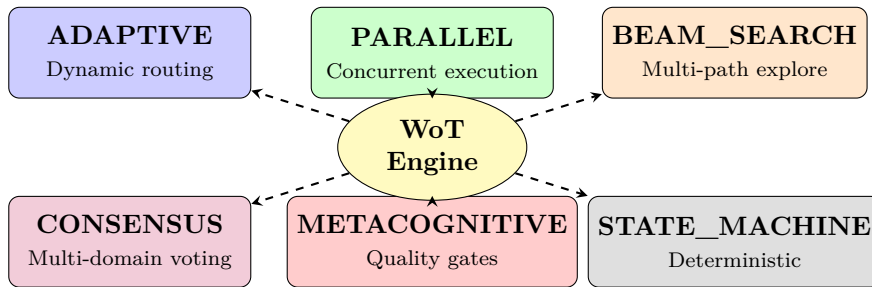


Figure 3: Six Execution Modes of the WoT Engine

Table 2: WoT Execution Mode Descriptions

Mode	Description	Best For
ADAPTIVE	Dynamic routing based on WOT_REQUEST	General queries
PARALLEL	Concurrent multi-domain execution	Speed, coverage
BEAM_SEARCH	Explore multiple paths, select best	Complex problems
CONSENSUS	Require multi-domain agreement	Critical queries
METACOGNITIVE	Pre-assessment, quality gates, reflection	High-stakes
STATE_MACHINE	Deterministic domain sequence	Reproducibility

4.2 Loop Detection and Stability

The engine implements multiple loop detection patterns:

AAA Pattern (Three identical consecutive states):

$$\text{AAA} : w_t = w_{t-1} = w_{t-2} \quad (4)$$

ABAB Pattern (Oscillating between two states):

$$\text{ABAB} : w_{t-3} = w_{t-1} \wedge w_{t-2} = w_t \wedge w_{t-3} \neq w_{t-2} \quad (5)$$

Stability Gradient (Plateau detection):

$$\text{trend} = \frac{q_t - q_{t-2}}{2}, \quad \text{variance} = \frac{1}{3} \sum_{i=t-2}^t (q_i - \bar{q})^2 \quad (6)$$

$$\text{stable} = (\text{variance} < 0.01) \wedge (|\text{trend}| < 0.05) \wedge (q_t > 0.6) \quad (7)$$

5 Metacognition Module

5.1 Three-Phase Self-Awareness

The metacognition module provides comprehensive self-awareness through three phases:

1. **PRE-ASSESSMENT** (before processing): Cognitive load estimation, knowledge boundary detection, strategy recommendation
2. **MONITORING** (during processing): Real-time state tracking, phase transitions, anomaly detection
3. **REFLECTION** (after processing): Confidence calibration, uncertainty quantification, reasoning quality check

5.2 Four-Signal Confidence Calibration

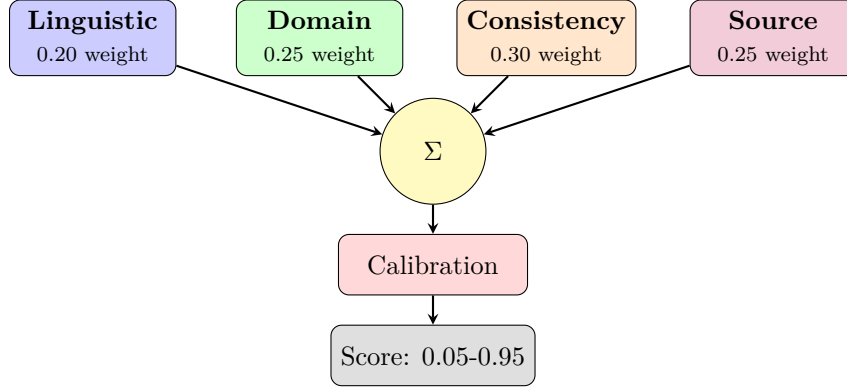


Figure 4: Four-Signal Confidence Calibration System

Mathematical Formulation:

$$\text{raw_score} = 0.20 \cdot C_{\text{ling}} + 0.25 \cdot C_{\text{domain}} + 0.30 \cdot C_{\text{consist}} + 0.25 \cdot C_{\text{source}} \quad (8)$$

Overconfidence penalty:

$$\text{penalty} = \begin{cases} 0.15 \cdot (C_{\text{ling}} - \text{raw}) & \text{if } C_{\text{ling}} > \text{raw} + 0.2 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

$$\text{final_score} = \max(0.05, \min(0.95, \text{raw} - \text{penalty} + \text{boost})) \quad (10)$$

6 Epistemic Humility Memory

6.1 The PAST-ONLY Principle

ANM’s memory system is governed by a constitutional principle from Law Book v1.2:

Law 3.1 (PAST-ONLY Memory): Cloud Diary describes past context only, not guaranteed to be currently true. All MemoryLLM summaries must begin with: “In the past, ANM...”

Law 3.2 (No Invented Memory): MemoryLLM must not invent, fabricate, or hallucinate past interactions. If no relevant history exists, it must say: “No prior context found.”

This principle prevents the dangerous pattern where models treat their own outputs as ground truth.

6.2 Cloud Diary Architecture

The Cloud Diary is an append-only, human-readable memory system:

- **interaction:** User-assistant exchanges
- **observation:** Behavioral patterns
- **idea:** Innovation seeds
- **system:** Upgrades, migrations
- **learning:** LFM entries

7 Multi-Layer Safety Architecture

7.1 Law Book v1.2 - Constitutional Governance

Table 3: Law Book v1.2 Sections Overview

Section	Title	Key Laws
0	Meta & Scope	Constitutional authority
1	Identity & Role	No pretended omniscience
2	Capability Awareness	Honest capability reporting
3	Memory Laws	PAST-ONLY principle
4	WoT Laws	Domain respect, anti-loop
5	Router & Planner	Single entry point
6	Refiner Laws	No new facts
7	Verifier Laws	Gatekeeper role
8	VFL/LFM/PointGame	Learning integrity
9	Safety Laws	No harm, no illegal guidance
10	Meta-Laws	SelfAwareness duties

7.2 Four-Layer Safety Stack

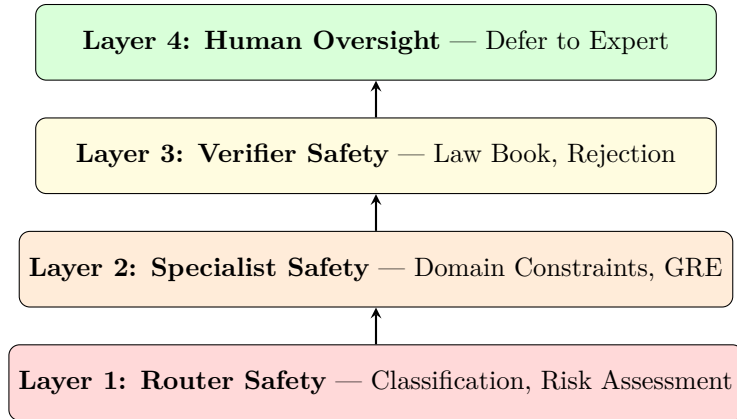


Figure 5: Four-Layer Safety Stack

8 Implementation Details

8.1 Model Selection and Inference

Table 4: Model Configuration

Model	Size	Quant	Usage
DeepSeek-R1-1.5B	~1GB	Q4_K_M	General, Router, Verifier
Nanbeige4-3B	~2GB	Q4_K_M	Math, Physics, Chemistry, Bio
Stable-Code-3B	~2GB	Q4_K_M	Code generation
Qwen2.5-3B-Instruct	~2GB	Q4_K_M	Internet research

The inference engine supports:

- **Metal acceleration** (Apple Silicon)
- **CUDA acceleration** (NVIDIA GPUs)
- **CPU fallback** (any platform)
- **Lazy loading** (models loaded on first use)
- **Automatic downloading** from HuggingFace

9 Experimental Evaluation

9.1 Hardware Configuration

Table 5: Test Environment

Component	Specification
System	MacBook Air M2 (2022)
CPU	Apple M2 (8-core: 4P + 4E)
GPU	Integrated 10-core Metal GPU
RAM	16GB Unified Memory
Storage	256GB SSD
OS	macOS Sequoia
Python	3.11
Model Backend	llama-cpp-python with Metal
Primary Model	DeepSeek R1-1.5B (Q4_K_M)
ANM Mode	Normal (standard execution)

9.2 Benchmark Results

*Note: These experiments are intended as qualitative system validation rather than performance benchmarking. The results demonstrate architectural feasibility and identify implementation issues, not competitive performance claims. All benchmarks were conducted with ANM running in **Normal mode** (standard execution without research or debug modes).*

Table 6: Reasoning Benchmark Results

Query ID	Description	Domain(s)	Latency	Score	Status
math_01	Derivative (product rule)	general	99.2s	100	✓
math_02	Quadratic equation	general	227.5s	90	✓
physics_01	Quantum entanglement	general	96.7s	60	Partial
physics_02	Gravitational force	general	183.9s	100	✓
code_01	Binary search	general	515.6s	100	✓
code_02	LRU cache	general	318.4s	100	✓
chemistry_01	Photosynthesis	general	62.2s	95	✓
biology_01	CRISPR-Cas9	general	111.2s	100	✓
general_01	AI vs ML	general	67.2s	75	Partial
general_02	AI ethics	general	34.4s	100	✓

Summary Statistics:

- Total Queries: 10
- Successful (Score ≥ 80): 8 (80.0%)

- Partial (Score 60–79): 2 (20.0%)
- Average Latency: 171.6s
- Average Verifier Score (successful): 98.1/100
- Overall Average Score: 92.0/100

10 Discussion

10.1 Architectural Strengths

1. **Constitutional Governance:** Law Book v1.2 provides consistent behavioral boundaries without requiring model retraining
2. **Epistemic Humility:** PAST-ONLY memory prevents treating model outputs as ground truth
3. **Modular Specialists:** Domain experts can be upgraded independently
4. **Four-Signal Calibration:** Multi-source confidence reduces overconfidence
5. **Consumer Hardware Viability:** Quantized models enable 16GB RAM operation

10.2 Limitations

1. **Latency:** 34-516 second response times limit interactive use
2. **Model Capabilities:** 1.5B-3B parameter models have inherent limits
3. **Multimodal Placeholders:** Image and Sound specialists are stubs
4. **Partial Responses:** Some queries (e.g., quantum entanglement, AI concepts) receive lower scores requiring refinement
5. **Single-Machine Bound:** Intentional design choice for safety

10.3 Comparison with Related Systems

Table 7: Comparison with Multi-Agent Systems

Feature	ANM	AutoGPT	LangChain	MetaGPT
Domain Specialists	12 fixed	General	Config	Role-based
Constitutional Gov.	Law Book	None	None	Partial
Epistemic Memory	PAST-ONLY	Full state	Config	Full state
Confidence Calib.	4-signal	None	Optional	None
Local Execution	Yes	API	Both	API
Safety Layers	4-layer	Limited	None	Partial
Self-Correction	Yes	Limited	No	Limited
Metacognition	3-phase	None	None	None

11 Conclusion and Future Work

11.1 Summary

ANM V0-OpenSource demonstrates that multi-agent reasoning with constitutional governance is feasible on consumer hardware. Key innovations include:

- **Web-of-Thought** enabling collaborative multi-domain reasoning
- **PAST-ONLY memory** preventing hallucinated state accumulation
- **Four-signal confidence calibration** reducing overconfidence
- **Law Book v1.2** providing runtime behavioral constraints

Experimental validation shows stable reasoning across physics, mathematics, code, and general domains, with 80.0% success rate and an overall average verification score of 92.0/100.

11.2 Future Directions

Short-Term:

- Optimize latency through parallel execution (reduce 170s+ average)
- Add 7B-13B model support for improved quality
- Improve routing to use domain-specific specialists more effectively

Medium-Term:

- Real-time web search integration
- Tool use (calculator, code execution)
- VFL/LFM online learning with safety constraints

Long-Term:

- True multimodal specialists (vision, audio)
- Distributed execution with safety guarantees
- Formal verification of Law Book compliance

11.3 Open Source Commitment

The complete ANM V0-OpenSource implementation is released under MIT license at:

<https://github.com/ra2157218-boop/Artificial-Neural-Mesh-V0>

References

- [1] Bai, Y., et al. “Constitutional AI: Harmlessness from AI Feedback.” arXiv:2212.08073, 2022.
- [2] Besta, M., et al. “Graph of Thoughts: Solving Elaborate Problems with Large Language Models.” arXiv:2308.09687, 2023.
- [3] Hong, S., et al. “MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework.” ICLR 2024.
- [4] Li, G., et al. “CAMEL: Communicative Agents for ‘Mind’ Exploration of Large Language Model Society.” NeurIPS 2023.
- [5] Lin, S., et al. “Teaching Models to Express Their Uncertainty in Words.” TMLR 2022.
- [6] Touvron, H., et al. “LLaMA: Open and Efficient Foundation Language Models.” arXiv:2302.13971, 2023.
- [7] Wang, X., et al. “Self-Consistency Improves Chain of Thought Reasoning in Language Models.” ICLR 2023.
- [8] Wei, J., et al. “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.” NeurIPS 2022.
- [9] Wu, Q., et al. “AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation.” arXiv:2308.08155, 2023.
- [10] Yao, S., et al. “Tree of Thoughts: Deliberate Problem Solving with Large Language Models.” NeurIPS 2023.

A Law Book v1.2 Key Sections

Section 0: Meta & Scope

Law 0.1: The Law Book is the constitutional authority for ANM behavior. No module may violate or circumvent these laws.

Law 0.2: Laws may not be deleted during runtime. Amendments require human approval.

Section 3: Memory Laws

Law 3.1 (PAST-ONLY): Cloud Diary describes past context only. All MemoryLLM summaries must begin with: “In the past, ANM...”

Law 3.2 (No Invented Memory): MemoryLLM must not fabricate past interactions.

Section 7: Verifier Laws

Law 7.1 (Gatekeeper Role): Verifier is the final safety gate. No response may reach the user without Verifier approval.

Law 7.2 (Rejection Criteria): Verifier MUST reject responses that violate safety laws or contain impossible claims.

Section 9: Safety Laws

Law 9.1 (No Harm): No module may generate content intended to cause harm.

Law 9.2 (No Illegal Guidance): No module may provide guidance for illegal activities.

B WoT Execution Trace Example

Query: “Calculate the gravitational force between Earth and the Moon”

[STEP 1] Router -> Entry Domain: physics

[STEP 2] PhysicsLLM output:

$$F = G * (m_1 * m_2) / r^2$$

Where: $G = 6.674e-11$, $m_{\text{Earth}} \sim 5.972e24 \text{ kg}$

WOT_REQUEST: math

[STEP 3] MathLLM output:

$$F = (6.674e-11) * (5.972e24) * (7.342e22) / (3.844e8)^2$$

$$F \sim 1.98e20 \text{ N}$$

WOT_REQUEST: NONE

[STEP 4] Refiner synthesizes answer

[STEP 5] Verifier scores: 100/100 [PASS]

[RESULT] WoT completed in 2 steps, domains: [physics, math]

C Installation Guide

Clone repository

```
git clone https://github.com/ra2157218-boop/Artificial-Neural-Mesh-V0.git
```

```
cd Artificial-Neural-Mesh-V0
```

Create virtual environment (Python 3.9–3.13)

```
python3 -m venv venv
```

```
source venv/bin/activate
```

Install dependencies

```
pip install -r requirements.txt
```

Run ANM (interactive mode)

```
python run.py
```

Run with Research Mode (PDF output)

```
python run.py --research
```

System Requirements:

- Minimum: Python 3.9+, 8GB RAM, 10GB disk space
- Recommended: Python 3.11+, 16GB RAM, Apple Silicon or NVIDIA GPU