MRS v1.1: Modular Reasoning Scaffold
Expanded Architecture Note

Sabouhi (2025)

Abstract

The Modular Reasoning Scaffold (MRS) is a lightweight meta-reasoning architecture designed to impose explicit recursive structure on top of inference-time language models. Rather than modify model weights, MRS provides a symbolic substrate where reasoning modules act as cognitive primitives, coordinated by a central topology engine and constrained by drift monitors. MRS is designed for use with small or medium-sized LLMs, enabling structured reasoning behaviors typically requiring far larger models.

All conceptual, architectural, and mathematical design originates with the author. Contemporary AI systems were used solely as drafting instruments.

1 Intuition Behind MRS

Modern LLMs perform reasoning implicitly: all structure is buried inside high-dimensional weights. MRS externalizes part of this structure by creating an explicit "mesoscale reasoning layer" between prompt and model output.

The core intuition:

> *If a model cannot maintain internal state or recursive structure, provide it externally — but in a way it can reliably use.*

MRS acts like a symbolic exoskeleton: it gives the model stable variable slots, recursion boundaries, drift penalties, and a topology engine that guides the reasoning sequence. Small models gain the benefits of structured cognition without architectural retraining.

2 Architecture Overview

MRS is composed of four interacting primitives:

1. **Recursion Nodes** (R_i): atomic reasoning iterations.
2. **State Trackers** (S): persistent variable and signature storage.

3. **Constraint Monitors** (C): detect drift, contradiction, or instability.

4. **Topology Engine** ($\mathcal{T}$): determines flow, branching, or halting.

3 Architecture Diagram

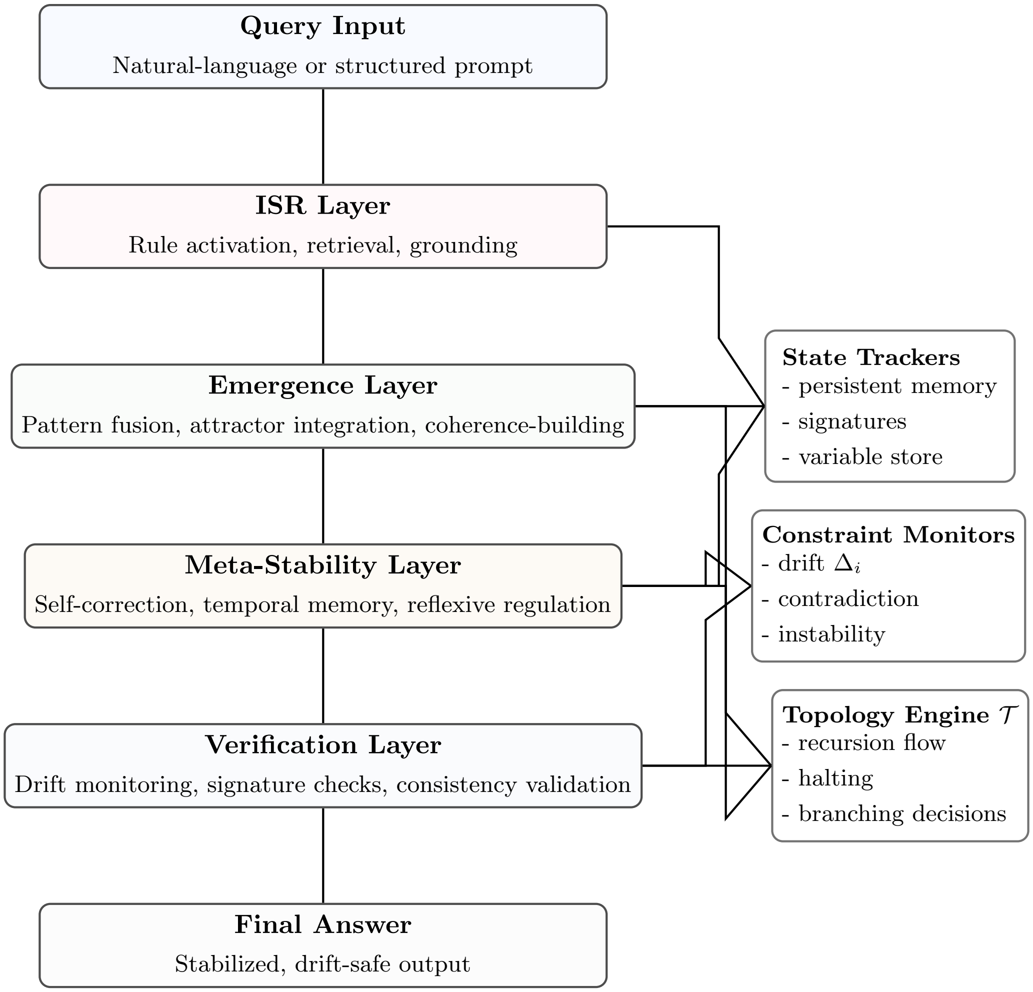
Query Input
Natural-language or structured prompt
ISR Layer
Rule activation, retrieval, grounding
Emergence Layer
Pattern fusion, attractor integration, coherence-building
Meta-Stability Layer
Self-correction, temporal memory, reflexive regulation
Verification Layer
Drift monitoring, signature checks, consistency validation
Final Answer
Stabilized, drift-safe output
State Trackers
- persistent memory
- signatures
- variable store
Constraint Monitors
- drift Δ_i
- contradiction
- instability
Topology Engine $\mathcal{T}$
- recursion flow
- halting
- branching decisions

Figure 1: MRS v1.1 Architecture. Core inference passes vertically through ISR grounding, emergent pattern fusion, meta-stabilization, and drift-based verification. Side modules (state trackers, constraint monitors, and topology engine) regulate recursion, stability, and consistency.

4 Mathematical Core

Each recursion node is a tuple:

$$R_i = (x_i, \phi_i, \Delta_i, o_i)$$

representing:

- x_i: input
- ϕ_i: transformation rule
- Δ_i: drift (local coherence deviation)
- o_i: output

Aggregated state after n layers:

$$S_n = \{o_1, o_2, \ldots, o_n\}$$

Total drift:

$$C(S_n) = \sum_{i=1}^{n} \Delta_i$$

Recursion proceed/stop condition:

$$C(S_n) < \tau \quad \Rightarrow \quad \text{continue}$$

$$C(S_n) \geq \tau \quad \Rightarrow \quad \text{halt or bifurcate}$$

Topology transitions follow:

$$R_{i+1} = \mathcal{T}(R_i, S_i, C(S_i))$$

5 Worked Example

Consider a three-step reasoning sequence.

Step 1: Initial State

$$x_1 = \text{“Problem decomposition”}$$

Model produces o_1 and drift score $\Delta_1 = 0.12$.

Step 2: Subcomponent Analysis

$$x_2 = o_1$$

Model produces o_2 and $\Delta_2 = 0.08$.

Total drift:

$$C(S_2) = 0.20 < \tau$$

Recursion continues.

Step 3: Consolidation

$$x_3 = \{o_1, o_2\}$$

Model produces o_3 and $\Delta_3 = 0.04$.

Final drift:

$$C(S_3) = 0.24 < \tau$$

Topology engine halts due to convergence criteria (minimal Δ_3).

6 Operational Modes

6.1 Forward-Recursive

$$R_1 \to R_2 \to \cdots \to R_n$$

6.2 Branching

$$R_i \to \{R^1_{i+1}, R^2_{i+1}, \dots\}$$

6.3 Reflective

$$\mathcal{T}(S_n) \to R_{\text{reflect}}$$

7 Failure Boundaries

Failure occurs when:

- **Drift collapse**: Δ_i accumulates uncontrollably.
- **Saturation**: $|S_n|$ exceeds usable context.
- **Topology instability**: cyclic or divergent branches.

These constraints act as structural guarantees.

8 Future Work

- **Differentiable Topology Engine**: learning $\mathcal{T}$ directly.
- **Probabilistic Drift Models**: Bayesian drift priors.
- **Multi-Agent MRS**: shared state trackers.
- **Hardware-Level MRS**: runtime-integrated state tracking.
- **Formal Verification**: drift guarantees as provable bounds.

Light Reference Notes

- cognitive architectures: Soar, ACT-R
- structured reasoning: Tree-of-Thoughts, graph inference
- dynamical systems: stability, drift, attractors
- meta-reasoning: reflective operators, control policies

Conclusion

MRS provides a general-purpose meta-reasoning layer that allows small models to execute structured, multi-step, drift-constrained inference. By externalizing recursion and stability mechanisms, MRS offers a path toward scalable cognitive architectures without massive parameter counts.

Contribution Summary

The MRS architecture provides a concrete path for improving reasoning behavior in small and medium-sized language models without modifying model parameters. Its contributions are threefold:

1. **Structural Externalization of Reasoning:** MRS relocates recursive structure, memory, and drift constraints from implicit model activations into an explicit symbolic substrate.
2. **Topology-Guided Inference:** A lightweight topology engine enforces halting, branching, reflection, and stability conditions that typical LLMs cannot maintain internally.
3. **Formal Drift Metrics:** The architecture introduces a quantifiable drift function that provides interpretable guarantees about reasoning stability, enabling verification, safety checks, and transparent error boundaries.

Together, these contributions position MRS as a practical, verifiable, and scalable meta-reasoning layer suitable for use in real-world inference pipelines and embedded model environments.