SV-Core: A Teleological Cognitive Architecture for Agentic Large Language Models

Alexandre Vinas (2025)

Abstract

Modern Large Language Models (LLMs) exhibit remarkable reasoning abilities but remain fundamentally ateleological: they do not maintain persistent goals, stable internal states, or long-range coherence. SV-Core introduces a minimal teleological cognitive architecture that adds directionality and structured internal dynamics to LLM-based agents without modifying model weights. It provides goal-oriented memory, structural representation layers, orientational shaping, coherence stabilization, and phase-transition mechanisms. This document presents the conceptual foundations, mathematical operators, implementation, and scientific motivations behind SV-Core.

1 Introduction

LLMs generate high-quality text but lack internal purpose. Existing agent systems rely on external scripting (ReAct), heuristic loops (Reflexion), or environment-driven planners (Voyager), instead of an internal cognitive structure.

SV-Core proposes a teleological layer that wraps around any LLM and provides:

- persistent, goal-aligned memory,
- structural transformation,
- orientational activation,
- coherence stabilization,
- phase-transition dynamics.

SV-Core is not a new model: it is a cognitive operator, fully differentiable, modular, and compatible with any transformer.

1.1 Related Work

Several approaches have attempted to add agency to LLMs:

- **ReAct** (Yao et al., 2023): reasoning + action, but no persistent internal state.
- **Reflexion** (Shinn et al., 2023): self-correction without teleological orientation.
- **Tree-of-Thoughts**: improves reasoning, but lacks directional stability.
- **Voyager**: exploration agent, but externally scripted.

- **SOAR / ACT-R**: symbolic architectures incompatible with neural LLMs.

SV-Core differs by introducing internal teleological operators directly in the inference loop, without weight modification or external scripting.

1.2 Scope of the Present Work

This paper presents SV-Core, a minimal and executable subset of a broader cognitive framework called the *Living System* (SV).

SV-Core includes only the foundational operators:

- μ-TEL — goal-conditioned memory,
- Λ — structural transformation,
- Ω^* — orientational shaping,
- $\circledcirc$ — unitary stabilization,
- $C\Omega$ — coherence field,
- PTOr — phase-transition operator.

Higher-level modules of the full SV architecture—such as hierarchical governance, multi-agent teleology, and ecological embedding—are deliberately out of scope.

The purpose of this document is therefore two-fold:

1. to present a minimal operational prototype that researchers can directly test,
2. to establish a scientific foundation for forthcoming SV extensions.

2 Architecture Overview

2.1 The Teleological Pipeline

The full pipeline is:

$$\Phi^* \to \mu\text{-TEL} \to \Lambda \to \Omega^* \to \circledcirc \to C\Omega \to \text{PTOr}$$

Each operator corresponds to a distinct cognitive function.

- Φ^* — **Presence**: initial orientational encoding.
- **μ-TEL — Teleological Memory**: gated memory updated toward a goal vector.
- Λ — **Structure**: extracts and organizes semantic features.
- Ω^* — **Orientation**: produces directional activation.
- $\circledcirc$ — **Unitary Core**: normalizes the state into a stable manifold.
- $C\Omega$ — **Coherence Field**: smooth self-correction across cycles.
- **PTOr — Phase Transition Operator**: creates a qualitative change producing a decision-ready vector.

2.2 Design Principles

SV-Core follows four principles:

1. teleology precedes action,
2. no retraining required,
3. compositional modularity,
4. compatibility with modern LLMs.

3 Implementation Summary

SV-Core is implemented in approximately 200 lines of PyTorch:

- no custom CUDA kernels,
- no model retraining,
- fully differentiable,
- memory registered via `register_buffer`.

It includes a class per operator and an `SVCore` controller orchestrating the pipeline.

4 Integration with LLMs

4.1 Hidden-State Extraction

Any transformer exposes hidden states. SV-Core uses the last-token embedding as the initial state Φ^*.

4.2 Fusion Strategies

A minimal integration uses:

$$\text{fused} = h_{\text{LLM}} + h_{\text{SV}}$$

Other approaches (attention biasing, planning layers) are possible.

4.3 Agentic Behavior

SV-Core provides:

- persistent direction,
- reduced drift,
- more stable multi-step reasoning,
- improved goal-conditioning.

5 Scientific Motivation

LLMs show local competence without global control. SV-Core implements:

- goal-aligned memory,
- structural filtering,
- manifold stabilization,
- predictable phase transitions.

These ideas draw from dynamical systems, cognitive architectures, and attractor dynamics.

6 Experimental Expectations

Predicted behaviors include:

1. fewer contradictions,
2. more stable reasoning chains,
3. inspectable internal states,
4. reduced noise propagation,
5. identifiable phase-transition clusters.

7 Limitations

SV-Core is a minimal foundation with known constraints:

1. no learned teleology,
2. no large-scale experimental validation,
3. requires hidden-state access (not possible for closed APIs),
4. memory dimension must match LLM hidden size,
5. no meta-learning or reinforcement loop,
6. no multi-agent dynamics.

8 Future Work

SV-Core represents only the foundational layer of the Living System architecture. Advanced components—hierarchical governance, multi-scale coherence, and extended agent dynamics—will be presented in future technical reports.

9 Conclusion

SV-Core introduces a minimal teleological architecture compatible with real LLMs. It is modular, stable, reproducible, and provides a scientific basis for developing future agentic systems.

Citation

Vinas, A. (2025). *SV-Core: A Teleological Cognitive Architecture for Agentic Large Language Models*. Zenodo. https://doi.org/10.5281/zenodo.17889413

Code Availability

GitHub: https://github.com/alexjoseph-creator/sv-core Zenodo: https://doi.org/10.5281/zenodo.17889413