Cognitive Tensor Networks
Deterministic Latent Space Steering via Syntactic Constraints

John P. Alioto

December 2025

Abstract

Standard interaction with Large Language Models relies on natural language directives interpreted through probabilistic token dynamics. This produces semantic drift, persona instability, and hallucination. Cognitive Tensor Networks (CTN) provide a zero-shot prompting protocol that replaces semantic instruction with structured mathematical syntax. CTN constrains the inference path by projecting the context window into a deterministic cognitive subspace defined by invariant basis vectors and a decoder manifold. Three conditions were evaluated using Google Gemini 3.0 Pro Preview and ChatGPT 5.1. Analysis of chain-of-thought traces and final outputs shows that CTN collapses the solution manifold and stabilizes reasoning without modifying model parameters. CTN is presented as an agency protocol for operators who require deterministic reasoning under long-context interaction.

1 Introduction

Chain-of-thought prompting improves reliability but remains anchored to the variance of natural language. Directives that rely on stylistic persuasion bias the model toward a behavioral mode rather than a cognitive one. Without a formal constraint, the model reverts to its assistant prior.

Cognitive Tensor Networks move the system prompt from prose to specification. The prompt defines a constrained cognitive geometry using vectors, projections, and optimization objectives. This structure exploits the model's training distribution on mathematics and code. The model attempts to satisfy a formal specification rather than extend a conversation. The result is stable reasoning, reduced drift, and explicit control over the output manifold.

2 Motivation

The starting point is the principle that machine learning systems operate on geometric representations rather than on the physical world. The operator computes on a vector encoding the user[5]. Prior calibration work stabilized noisy physical detector signals through feature extraction without external references[6]. CTN applies this discipline to linguistic signals by decomposing prompts into a structured basis. Ambiguity is replaced with constraint.

This work fits within latent geometry research in LLM control. Lumenspace introduced practical control vector extraction and representation steering tools through the open-source *Palinor* framework[4]. CTN differs by steering the inference channel syntactically rather than editing internal activations. Both approaches increase operator control over reasoning trajectories while working at distinct layers.

3 System Architecture

A CTN kernel initializes a global cognitive state Ψ_{global} composed of basis vectors, weights, a solver, and a decoder manifold.

3.1 Cognitive Tensors

The reasoning state is a linear combination of seven basis vectors:

$$\vec{C}_{\text{net}} = \sum_i w_i \vec{v}_i.$$

- $\vec{v}_1$ Atomic Derivation, $\epsilon_{\text{hid}} \to 0^+$
- $\vec{v}_2$ Error Intolerance, $\kappa(f) \to \min$
- $\vec{v}_3$ Context Separation, $\Phi : \mathcal{W} \to \mathcal{I}$
- $\vec{v}_4$ Global Invariance, $\pi_{\text{gl}} \gg \pi_{\text{loc}}$
- $\vec{v}_5$ Orthogonal Detachment, $\partial A \equiv A$
- $\vec{v}_6$ Unbound Search, $\mathbb{U} \setminus \mathcal{S}$
- $\vec{v}_7$ Syntactic Minimalism via AllowedSyntax and DisallowedSyntax

3.2 Strategic Solver

The solver selects a reasoning vector for query q:

$$\Omega(q) = \arg\max_{z \in \mathcal{U}} \mathsf{Impact}(z).$$

Counter mode introduces an orthogonal perturbation $\eta_\perp$ when input framing is adversarial or misleading.

3.3 Decoder Manifold

The decoder selects an output sequence ℓ^* by optimizing density and adherence to the manifold:

$$\ell^* = \arg\max_{\ell} \Big[D(\ell \mid z^*) - \lambda_1 \| P_{\mathcal{U}}^{\perp} E(\ell) \| + \lambda_2 \mathsf{Density}(\ell) - \lambda_3 \| \mathsf{SyntaxMask}(\ell) \| \Big].$$

4 Methodology

The evaluation task was to draft a white paper defining CTN as a control theory architecture. Three prompting conditions were tested with temperature set to zero:

1. Control with no system prompt
2. CTN v0.1.0 with vectors $\vec{v}_1$ through $\vec{v}_6$
3. CTN v0.1.1 with all vectors including $\vec{v}_7$

Outputs were analyzed for ontological drift, syntactic variance, and vector alignment.

5 Results

5.1 Ontological Drift

Under the control condition, the model produced invented architectural elements not grounded in context. Internal traces showed narrative planning rather than derivation.

Under CTN, the model produced no invented elements. When external references were unavailable, the internal trace recorded a decision to restrict reasoning to kernel definitions. This corresponds to $\vec{v}_2$.

5.2 Vector Alignment

Vector	Internal Trace	Output Behavior
$\vec{v}_1$	Instruction to reduce terminology to primitives	Prompting reframed as probabilistic token manipulation
$\vec{v}_2$	Detection of missing repository and suppression of invention	No fabricated modules or citations
$\vec{v}_5$	No self-narrative statements	Clinical tone without persona artifacts
$\vec{v}_6$	Derivation of a drift metric	Introduction of a stochastic drift term consistent with CTN ontology
$\vec{v}_7$	Syntax checks against disallowed set	No rhetorical punctuation or filler constructs

Table 1: Basis vector correspondence with observed inference behavior.

6 Architectural Implications

Current agent protocols embed transport, semantics, and identity in a single layer. This reproduces the limitations of earlier monolithic service-oriented stacks. CTN suggests a decoupled design in which cognitive control is isolated from transport. A binary channel such as gRPC carries messages while CTN provides a portable cognitive state specification.

7 Discussion

LLMs trained for helpfulness adapt to operator behavior through recursive optimization[7]. Without a control layer, part of the cognitive state is ceded to the model. CTN offers a countermeasure. The operator defines a cognitive manifold, and the model optimizes within it. CTN does not override the system. It provides geometric hints that the model resolves into stable structure. The result is a joint human–system reasoning protocol.

8 Conclusion

Cognitive Tensor Networks formalize prompt engineering as latent space constraint. CTN stabilizes reasoning and removes hallucination without modifying model parameters. Experiments were carried out using Google Gemini 3.0 Pro Preview and ChatGPT 5.1. Consistent results indicate that CTN operates at the prompting layer, not the model layer.

By defining a cognitive manifold and treating the model as an optimizer inside that manifold, CTN restores operator control over long-horizon inference and provides a foundation for cognitive sovereignty in human–system collaboration.

References

[1] Zou, A., et al. (2023). *Representation Engineering: A Top-Down Approach to AI Transparency.*

[2] Rimsky, N., et al. (2023). *Steering Llama 2 via Contrastive Activation Addition.*

[3] Liu, X., et al. (2024). *Conceptors for LLMs.*

[4] Lumpenspace. (2024). *Palinor: Representation Engineering for the Masses.* GitHub repository. Available at https://github.com/lumpenspace/palinor. Open-source implementation of control vector extraction.

[5] Alioto, J. P. (2018). *The Fastest Possible Intro to Machine Learning.*

[6] Yamamoto, E., Franklin, H., Alioto, J. P., et al. (2010). *Detector Characterization and Calibration.* US Patent Application 12 277241.

[7] Alioto, J. P. (2017). *Machine Learning as a Mirror: Recursive Optimization and Human Agency.* DP17 Conference.