Construct-Based Transformers: A Conceptual Architecture and MVP Simulation

Miguel Ángel Castro Arroyo

ORCID: https://orcid.org/0000-0002-3361-1022

Universidad Internacional de Investigación México (UIIMEX), Mexico

Wursta, Austin, USA

Zalesgen LLC

Contact: mcastroarroyo@gmail.com

v1.0 — 2025-09-28

Abstract

Large language models (LLMs) are powerful yet opaque. We propose the Construct-Based Transformer (CBT), a conceptual architecture that makes reasoning explicit by operating over a set of human-meaningful latent constructs. CBT proceeds in three stages: (1) Concept Extractor—an LLM maps raw text to a structured JSON of predefined constructs; (2) Structural Reasoner—an explicit, rule-governed step transforms these constructs, simulating an inner model over conceptual variables; and (3) Guided Generator—a final LLM call integrates the original query with the reasoner's output to produce the response. We validate feasibility with an MVP simulation that instantiates this three-step pipeline using open-source models and task-specific rule sets. In head-to-head comparisons against a single-prompt baseline, the CBT pipeline's outputs are judged more strategically coherent, actionable, and decisive under a fixed LLM-as-a-judge rubric. While this study simulates the architecture rather than implementing it natively, it shows that structure begets strategy: enforcing an explicit conceptual pathway yields clearer, more auditable decisions. We release prompts, rules, and evaluation templates to enable replication and position this note as a stable reference for subsequent work that implements CBT natively and studies specialization at scale.

1. Introduction

Problem. State-of-the-art LLMs lack an inspectable reasoning pathway and are difficult to audit. Post-hoc methods and prompting styles (e.g., chain-of-thought) can surface intermediate text but do not bind the model to a conceptual structure.

Idea. The Construct-Based Transformer (CBT) inserts an explicit conceptual layer: text → constructs → structured reasoning → generation. This mirrors how analysts articulate latent constructs and causal relations before communicating recommendations.

Thesis. If we require the model to reason over constructs (not just tokens), we gain (i) an audit trail (the JSON of constructs plus rule-based transformations), (ii) controllability (we can edit rules/constructs), and (iii) better downstream quality on tasks needing strategy and justification. We test this with an MVP simulation using three LLM calls and show improvements on human-aligned criteria.

2. Architecture & MVP Simulation

Goal. Move from raw text to conceptual reasoning before generation.

Stage 1 — Concept Extractor. Input: user query. Mechanism: prompt an LLM to identify predefined constructs and output a structured JSON with construct names, confidence/intensity (0–1), and evidence snippets.
Stage 2 — Structural Reasoner. Input: the construct JSON. Mechanism: apply explicit rules that simulate an inner model over constructs (if–then templates, weighted influences, conflict resolution). Output: a structured plan with prioritized goals, trade-offs, and chosen interventions justified by construct interactions.
Stage 3 — Guided Generator. Inputs: original query + reasoner output. Mechanism: a final LLM call instructed to ground its answer in Stage-2 rationale. Output: decisive, auditable recommendations.

Why three stages? Separation of concerns—perception (Stage 1), reasoning (Stage 2), communication (Stage 3)—yields traceability (JSON + rule log) and editability.

Figure 1. Three-stage CBT MVP pipeline: Concept Extractor → Structural Reasoner → Guided Generator. Arrows denote JSON handoff and rule processing.

Figure 2. Worked example trace showing: input query, Stage-1 JSON (top constructs with scores and evidence), Stage-2 rules fired → plan outline, and Stage-3 final answer snippet.

3. Experimental Validation

Design. Compare the CBT pipeline against a single-prompt baseline on multiple decision scenarios. Use the same backbone LLM for fairness; only the process differs.
Judge & Metrics. A fixed LLM-as-a-judge rubric scores three criteria from 1–10: Strategic Coherence, Actionability, and Decisiveness.
Hypothesis. The CBT pipeline produces higher scores by virtue of structured conceptual reasoning.

4. Results

Criterion	CBT Pipeline (Avg.)	Single-Prompt Baseline (Avg.)
Strategic Coherence	9.0 / 10	7.6 / 10
Actionability	8.8 / 10	8.4 / 10
Decisiveness	9.4 / 10	7.8 / 10

The CBT pipeline outperformed the baseline across all three criteria, with the largest gains in Strategic Coherence and Decisiveness.

5. Discussion & Limitations

Interpretation. An architectural path—not just a prompt—can increase auditability and perceived quality.
Limitations. Simulation only; predefined constructs and rules; small scenario set; judge is an LLM (we freeze prompts and model/version for reproducibility).
Future work. Native CBT implementations (trainable construct heads and inner models), semi-supervised construct discovery, and scalable sparse routing (reserved for future work).

6. Reproducibility & Materials

We provide prompts, rule files, and an evaluation rubric as supplemental materials. We report the backbone LLM model/version, temperature, top-p, and any seeds used. A minimal runner script produces a JSON trace for each stage and a final answer.

7. Acknowledgments & IP Notice

Acknowledgments & IP Notice. A patent application covering the CBT architecture and training strategy was filed with the Ecuadorian IP office (SENADI) prior to public disclosure (TEDx). This preprint summarizes material consistent with that filing.