WHITE PAPER
A Practical Cognitive Architecture for Human Goal Pursuit

Sequence Architecture Team

Abstract

Humans maintain goals through an implicit process of temporal coherence: the capacity to represent a desired future state, infer the causal structure required to achieve it, act to reduce uncertainty, and continually update beliefs based on observed outcomes. Contemporary digital tools fail to support this process. They record tasks but not intentions, list actions without modeling causality, and fragment decision-making across interfaces that ignore how cognition actually organizes time.

This paper introduces the **Temporal Coherence Framework (TCF)**: an applied computational architecture inspired by Predictive Processing, Active Predictive Coding (APC), Active Inference, and Hierarchical Reinforcement Learning (HRL). TCF formalizes the elements required to transform a user's high-level goals into actionable, uncertainty-reducing behavior through a structured generative model.

We describe how **Sequence**, an APC Execution System, operationalizes this framework through (a) a hierarchical Future State world model, (b) causal linkage induction, (c) reusable option primitives, (d) epistemic action selection ("glimpses"), (e) a rolling horizon model-predictive planning loop, and (f) a quantitative alignment measure, **Coherence Δ**, grounded in structured prediction error. By integrating concepts from computational neuroscience, Bayesian learning, and hierarchical control, Sequence aims to provide a practical mechanism for improving human planning, inference, and action in complex, uncertain environments.

The result is a system that does not simply track tasks, but actively maintains temporal alignment between intention and execution.

Contents

1 Introduction

1.1 Motivation

Human goal pursuit is fundamentally a problem of inference under uncertainty. To move toward a desired outcome, an intelligent agent must:

1. represent a future state,
2. infer what must be true for that future to be realized,
3. evaluate actions by their expected information gain and utility,
4. update beliefs based on prediction errors, and
5. adjust policy dynamically as new evidence appears.

In biological systems, these computations are performed continuously and implicitly by hierarchical generative models (Clark, 2013; Friston, 2005; Hohwy, 2020). But modern work environments degrade the very conditions under which these models evolved:

- information overload
- fragmented tools
- opaque causal relationships
- delayed or ambiguous feedback
- reactive rather than predictive behavior

Most software amplifies entropy instead of containing it. Where brains preserve coherence through hierarchical inference, contemporary digital systems force humans into flat lists, static documents, and opaque dashboards that lack temporal or causal structure. There is a gap between how humans think and the tools they are required to use.

1.2 Limitations of Current Approaches

Existing productivity systems—task managers, project boards, OKRs, calendar systems, LLM-based assistants—all share three core deficiencies:

1. **No generative world model.** They cannot infer hidden structure, preconditions, or causal dependencies.
2. **No hierarchical policy construction.** They treat each action as a disconnected, atomic task instead of part of an unfolding policy.
3. **No epistemic behavior.** They provide actions but not probes that reduce uncertainty (the backbone of all intelligent action).

As a result, users fall into cycles of overplanning, underlearning, firefighting, fragmented attention, and drift—the progressive loss of alignment between intent and action. Tools show activity; humans need coherence.

1.3 Toward a Computational Framework of Human Planning

Over the last decade, several converging theoretical frameworks—Predictive Processing, Active Inference, Active Predictive Coding, Bayesian cognition, Event Segmentation Theory, and Hierarchical RL—have converged on a shared insight:

> Intelligent agents maintain behavioral coherence by minimizing structured prediction error across levels of a hierarchical generative model.

This involves:

- representing desired outcomes as priors,
- decomposing the world into factorized subspaces (frames),
- identifying causal dependencies (linkages),
- constructing reusable action primitives (options),
- performing epistemic actions to reduce uncertainty (glimpses),
- and updating policies through short-horizon inference loops.

These principles define the core of what Sequence attempts to externalize.

1.4 Introducing the Temporal Coherence Framework (TCF)

The Temporal Coherence Framework proposes a structured mapping from neuroscience and control theory to everyday human reasoning. It rests on four components:

1. **A hierarchical Future State model** representing intention as a structured prior.
2. **A causal inference layer** that identifies "what must be true" for progress.
3. **An epistemic action module** to resolve uncertainty efficiently through minimal probes.
4. **A model-predictive planning loop** that replans continuously as new evidence becomes available.

Sequence operationalizes these components into a practical execution engine.

1.5 Contributions of This Paper

This white paper provides:

- A formal description of TCF as a computational analogue of Active Predictive Coding.
- A mapping between neuroscientific theory and practical decision-making structures.
- A precise mathematical formulation of Coherence Δ as structured prediction error.
- Algorithmic descriptions of Sequence modules (world model, options, glimpses, planner).
- Conceptual and empirical predictions regarding human performance under TCF.
- A discussion of safety, alignment, and bounded autonomy for cognitive support systems.

The goal is to articulate a technically grounded, scientifically defensible, and practically implementable architecture for human planning and action.

2 Background & Prior Work

A scientific grounding for the Temporal Coherence Framework.

2.1 Predictive Processing (PP)

Predictive Processing proposes that the brain operates as a hierarchical prediction machine (Clark, 2013; Hohwy, 2020). At every level of the cortical hierarchy, the system:

- generates predictions $\hat{o}_t$ about incoming sensory or cognitive data
- receives actual observations o_t
- computes prediction error $\varepsilon_t = o_t - \hat{o}_t$
- updates internal beliefs μ_t to minimize expected future error

This process creates an ongoing cycle of hypothesis testing, where the organism attempts to maintain a generative model of the world that is maximally self-consistent. Two core properties of PP are foundational for Sequence:

1. **Hierarchical abstraction:** Higher levels encode long-timescale, abstract states (goals, intentions), while lower layers encode detailed predictions (actions, sensations).
2. **Top-down causality and bottom-up correction:** Predictions flow downward; errors flow upward; coherence is maintained through continual adjustment.

PP is not merely a theory of perception—it extends naturally to planning, action, and cognitive control, implying that human behavior is fundamentally a process of inference under uncertainty.

2.2 Active Predictive Coding (APC)

APC extends Predictive Processing by integrating action selection, hierarchical reinforcement learning, and temporal abstraction (see *APC: A Unifying Neural Model*, MIT Press, 2023). APC offers a framework where:

- high-level goals become priors over trajectories
- low-level actions emerge from minimizing prediction error relative to those priors
- the brain constructs abstract state transition structures
- complex tasks are solved by composing reusable behavioral primitives

In APC:

- Policies are not learned as explicit functions.
- Policies emerge from inference within a hierarchical generative model.
- Reinforcement signals arise naturally from prediction error minimization.

This contrasts with typical RL approaches, which treat action selection as value maximization rather than inference over desired futures. Sequence's architecture—particularly its Frames, Linkages, Options, and Transfer Graph—maps directly onto APC's notions of option vectors, abstract state spaces, factorized transition dynamics, and composable behavioral modules.

2.3 Active Inference (AIx)

Active Inference provides a Bayesian formulation of agent behavior: agents act to minimize expected free energy (EFE), a measure that unifies instrumental objectives (goal satisfaction) and epistemic objectives (uncertainty reduction). EFE decomposes into:

$$G(\pi) = \underbrace{\text{expected divergence from preferred outcomes}}_{\text{instrumental}} + \underbrace{\text{expected information gain}}_{\text{epistemic}} \tag{1}$$

Actions are chosen not just to achieve goals but to resolve uncertainty in the generative model. This dual imperative (control + curiosity) maps directly to:

- Options = instrumental actions
- Glimpses = epistemic actions

Active Inference also formalizes multi-level control, message-passing inference, model-based adaptability, online updating, and structure learning. Sequence leverages these principles in its Glimpse scheduler, Δ scoring, and rolling MPC loop.

2.4 Hierarchical Reinforcement Learning (HRL) & Options Framework

HRL introduces the idea that intelligent agents require temporally extended actions, known as options (Sutton et al., 1999). An option consists of: a policy, initiation conditions, and a termination condition. Options allow an agent to operate across multiple timescales, reducing complexity by reusing skills, composing behaviors, and decoupling planning across horizons. Sequence adapts this framework by generating Options automatically, optimizing them with user-specific parameters, evaluating them via Δ, and transferring them across Frames. This converts user behavior into a skill library that improves over time.

2.5 Event Segmentation Theory (EST)

Event segmentation research (Zacks et al., 2007) demonstrates that humans naturally divide experience into discrete events whenever prediction error spikes. These boundaries structure memory, reduce cognitive load, and guide future predictions. Key insight: Human cognition is event-based, not continuous. Sequence implements this by separating life into Frames, structuring actions into short-horizon plans, and using Δ-based transitions to update event boundaries.

2.6 Causal Models & Structural Linkage Learning

Pearl's structural causal models (SCMs) and contemporary causal discovery methods argue that intelligent behavior requires identifying causal dependencies, required preconditions, and paths through latent variable structure. Sequence's Linkage Inference model performs a practical analogue of causal structure learning by asking: "What must be true for your Future State to become inevitable?" This creates a usable form of factorized belief structure analogous to latent causal graphs.

2.7 Integration: Toward a Unified View

All six traditions converge on a single thesis: Intelligent agents maintain coherence over time by predicting, testing, and updating hierarchical beliefs through structured action. This is the theoretical foundation of the Temporal Coherence Framework (TCF), and the basis for the Sequence Engine.

3 Theoretical Framework

The Formal Computational Model Underlying TCF and Sequence.

3.1 Notation and Preliminaries

We begin with a standard Bayesian generative model:

- s_t: latent (hidden) state at time t

- o_t: observation at time t
- a_t: action selected at time t
- π: policy (sequence of actions)
- μ_t: internal belief state (approximate posterior)
- A: likelihood model $p(o_t \mid s_t)$
- B: transition model $p(s_t \mid s_{t-1}, a_{t-1})$
- C: preferred future outcomes (goals)
- D: prior over initial states

Sequence does not claim to replicate these structures neurally; instead, it constructs usable analogues for human decision-making.

3.2 Generative Model

The generative model factorizes as:

$$p(o_{1:T}, s_{1:T}, \pi) = p(s_1) \prod_{t=1}^{T} p(o_t \mid s_t)\, p(s_{t+1} \mid s_t, a_t)\, p(a_t \mid \pi) \tag{2}$$

In Sequence:

- **Future State** defines C: preferences over outcomes.
- **Frames** define a factorization of the state space $s_t = (s_t^{(1)}, \ldots, s_t^{(K)})$.
- **Linkages** constrain allowable transitions in B.
- **Options** define macro-actions that parameterize a_t.
- **Glimpses** probe A to reduce uncertainty.

3.3 Hierarchical Structure

Predictive processing and APC both highlight a hierarchical generative model:

$$s_t^{(L)} \to s_t^{(L-1)} \to \cdots \to s_t^{(1)} \tag{3}$$

Higher layers encode long-horizon temporal predictions, abstract causal structure, and goal constraints. In Sequence:

- Future State corresponds to $s_t^{(L)}$.
- Frames correspond to $s_t^{(L-1)}$.
- Linkages correspond to constraints in the prior over transitions B.

This hierarchical structure enables factorized inference, reduced uncertainty propagation, and modular policy construction.

3.4 Predictive Coding in Sequence

Predictive coding requires computing prediction errors:

$$\varepsilon_t^{(1)} = o_t^{(1)} - \hat{o}_t^{(1)} \tag{4}$$

Sequence implements conceptual prediction errors when:

- observed progress mismatches expected progress
- a linkage is not satisfied
- a step fails to reduce Δ
- a glimpse contradicts assumptions

These conceptual errors drive updates to the causal model, the option space, and the current plan.

3.5 Epistemic vs Instrumental Value

Expected free energy:

$$G(\pi) = \underbrace{\mathbb{E}\left[D_{\mathrm{KL}}(Q(s_{t+1}) \| C)\right]}_{\text{instrumental}} - \underbrace{\mathbb{E}\left[H(o_{t+1} \mid \pi) - H(o_{t+1} \mid s_{t+1})\right]}_{\text{epistemic}} \tag{5}$$

Sequence mirrors this decomposition:

- Instrumental Value $\rightarrow$ Options (drive progress)
- Epistemic Value $\rightarrow$ Glimpses (reduce uncertainty)

Glimpses are chosen to minimize ambiguity in the causal model.

3.6 Formalizing Glimpses

A glimpse g is defined as a minimal action that maximizes expected information gain:

$$g^* = \arg\max_g \mathbb{E}\left[I(s; o \mid g)\right] \tag{6}$$

Sequence approximates this by identifying the most uncertain linkage, generating an action that tests it, measuring Δ response, and updating beliefs. This aligns epistemic behavior with cognitive science and Active Inference.

3.7 Rolling 4-Step Model Predictive Control (MPC)

Sequence implements a bounded-horizon MPC loop:

1. Construct a 4-step predictive model: $\hat{s}_{t:t+4}$.
2. Score each step by expected Δ improvement.
3. Select the first step.
4. Execute the step.
5. Observe outcome and compute Δ.
6. Replan from new state.

Formally: $a_t^* = \arg\min_{a_t} \mathbb{E}[G(\pi_{t:t+4})]$. This mirrors control theory applied to cognitive behavior.

3.8 Formal Definition of Coherence Δ

Coherence Δ is defined as:

$$\Delta_t = w_1 \cdot \text{Coverage}_t + w_2 \cdot \text{Prediction Accuracy}_t + w_3 \cdot \text{Time-to-Signal}_t \tag{7}$$

Where:

- **Coverage** is the proportion of satisfied linkages: $\text{Coverage}_t = \frac{1}{|L|} \sum_{\ell \in L} \mathbb{I}[\ell \text{ satisfied}]$
- **Prediction Accuracy** measures alignment between expected and observed progress: $1 - |\hat{p}_t - p_t|$
- **Time-to-Signal** penalizes slow feedback loops: $\text{TTS}_t = e^{-\lambda \cdot \tau_t}$

This can be interpreted as a structured prediction error metric under a bounded inference process.

3.9 Information-Theoretic Interpretation

Δ approximates a bounded version of expected free energy reduction:

$$\Delta_t \approx -G(\pi_t) \quad \text{(bounded, discretized)} \tag{8}$$

Thus:

- Higher $\Delta \rightarrow$ better temporal alignment
- Δ velocity $\rightarrow$ rate of improvement in belief coherence
- Δ saturation $\rightarrow$ overfitting or convergence
- Negative $\Delta \rightarrow$ model violation

This positions Δ as a domain-general measure of cognitive alignment.

4 The Sequence Engine Architecture

A hierarchical generative model for human planning and action.

4.1 Overview

The Sequence Engine is a structured inference system built to externalize and augment the computations underlying human planning. Inspired by Active Predictive Coding (APC), Predictive Processing (PP), and Active Inference, the engine implements a hierarchical generative model that:

1. Encodes a desired Future State (goal prior).
2. Factorizes cognition into Frames (latent subspaces).
3. Identifies Linkages (causal dependencies).
4. Generates Options (temporally extended macro-actions).
5. Selects Glimpses (epistemic actions) to reduce uncertainty.
6. Constructs a Rolling 4-Step Plan (bounded horizon MPC).
7. Computes Coherence Δ (alignment metric).
8. Updates beliefs and policies through an inference loop.

These modules form a closed-loop architecture: Future State → World Model → Options & Glimpses → Planner → Action → Observation → Δ Update → Replan.

4.2 Module 1: Future State Representation

4.2.1 Definition

The Future State is the highest-level prior over desired outcomes: $C = p(o \mid \text{desired})$. It is: explicit (user-specified), structured (time-bound, measurable), and causal (implicitly defines constraints on the environment).

4.2.2 Functional Role

Anchors the entire generative hierarchy, defines the direction of inference, provides the reference point for Δ computation, and determines which Frames are relevant.

4.2.3 Mapping to Theory

- **Future State:** Prior over preferred outcomes C in Active Inference.
- **Time horizon:** High-level temporal abstraction in PP/APC.
- **Constraint extraction:** State factorization in HRL & causal modeling.

4.3 Module 2: Frame Factorization (World Model)

4.3.1 Definition

Frames are factorized subspaces of the cognitive environment: $s_t = (s_t^{(1)}, s_t^{(2)}, \ldots, s_t^{(K)})$. Each Frame contains: a domain of action (Narrative, Demand Gen, Product, etc.), its own causal structure, and localized uncertainty.

4.3.2 Functional Role

Frames reduce complexity, isolate causal chains, enable modularity, and support transfer learning.

4.3.3 Mapping to Theory

- **Frames:** Factorized hidden states in hierarchical generative models.
- **Domain isolation:** Modular RL / HRL option decomposition.
- **Local priors:** Context priors D in Active Inference.

4.4 Module 3: Linkage Induction (Causal Layer)

4.4.1 Definition

Linkages are structured causal dependencies $\ell_i : s_t^{(k)} \to s_t^{(j)}$. They answer: "What must be true for the Future State to become likely?"

4.4.2 Functional Role

Linkages provide causal explanation, prioritization, structural constraints on planning, and the foundation of Δ coverage scoring.

4.4.3 Mapping to Theory

- **Linkages:** Structural Causal Models (Pearl).
- **Preconditions:** Transition constraints B in generative models.
- **Coverage:** Predictive coding precision-weighting.

4.5 Module 4: Option Generator (Policy Primitives)

4.5.1 Definition

Options are macro-actions: $o_k = \{I_k, \pi_k, T_k\}$. Where: I_k is the initiation set, π_k is policy parameters, and T_k is the termination condition.

4.5.2 Functional Role

Options provide reusable behavioral primitives, structured action space, and templates for policy execution.

4.5.3 Mapping to Theory

- **Options:** HRL options framework.
- **Transfer Graph:** Skill reuse in meta-RL and APC.
- **Rationale:** Goal-conditioned policy priors.

4.6 Module 5: Glimpse Generator (Epistemic Actions)

4.6.1 Definition

Glimpses are minimal actions designed to reduce uncertainty: $g^* = \arg\max_g \mathbb{E}[I(s; o \mid g)]$.

4.6.2 Functional Role

Glimpses test causal hypotheses, resolve ambiguous linkages, accelerate model updating, and reduce risk of misaligned plans.

4.6.3 Mapping to Theory

- **Glimpses:** Epistemic actions in Active Inference.
- **Uncertainty resolution:** Expected information gain.
- **Minimal probes:** Bayesian experimental design.

4.7 Module 6: Rolling 4-Step Planner (MPC)

4.7.1 Definition

A short-horizon MPC process: 1. Predict 4 future states, 2. Score options by Δ, 3. Commit only the first action, 4. Observe, 5. Replan.

4.7.2 Mapping to Theory

- **Planner:** Model Predictive Control (Kalman, Rawlings).
- **Rolling horizon:** Limited computational budget (bounded rationality).

- **Commit-observe-update:** Hierarchical predictive coding loop.

4.8 Module 7: Coherence Δ (Alignment Metric)

4.8.1 Role in Engine

Δ measures structured prediction error reduction, calibrates expectations, guides plan revision, strengthens causal hypotheses, and forms the backbone of the feedback loop.

4.8.2 Engine Update Rule

$\Delta_{t+1} = f(\Delta_t, \text{Coverage}, \text{Accuracy}, \text{TTS})$. Δ acts as the system's reward signal, certainty signal, and alignment signal.

4.9 Module 8: Transfer Graph (Skill Generalization)

4.9.1 Definition

A directed graph: $G = (V, E)$, where $E_{ij} = \Delta$ improvement when option i transfers to frame j.

4.9.2 Functional Role

Captures skill reuse, enables cross-domain generalization, accelerates planning, and supports meta-policy development.

4.9.3 Mapping to Theory

- **Transfer Graph:** APC option reuse.
- **Generalization:** HRL hierarchical skill sharing.
- **Δ-Weighted:** Confidence in learned behaviors.

5 Algorithmic Specification & Update Rules

Making the Sequence Engine operationally precise.

5.1 World Model Construction Algorithm

The world model is factored into three layers: Future State C, Frames $F_1, \ldots, F_K$, and Linkages $L = \{\ell_1, \ldots, \ell_n\}$.

World Model Induction [1] **Input:** FutureState FS, user context U **Output:** World model $WM = \{Frames, Linkages\}$ $Frames \leftarrow$ Decompose(FS, U) each Frame F in $Frames$ $CandidateLinkages \leftarrow$ GenerateHypotheses(F) $Linkages[F] \leftarrow$ Filter$(CandidateLinkages)$ $WM \leftarrow \{Frames, Linkages\}$ WM

Notes: Decompose uses semantic embeddings + temporal abstraction. Generate Hypotheses uses APC-style abstract transition priors + causal templates. Filter eliminates logically redundant, high-variance, or low-relevance linkages.

Theory Mapping: Equivalent to factorizing hidden state space in hierarchical generative models. Approximation of causal graph structure learning under constraints.

5.2 Option Generation Algorithm

Constructing reusable macro-actions. Each Option is a tuple: $o_k = \{I_k, \pi_k, T_k, \text{rationale}\}$.

Option Generator [1] **Input:** Linkages L, Frame F **Output:** Options O $O \leftarrow \emptyset$ each linkage ℓ in $L[F]$ $PrimitiveActions \leftarrow \text{Expand}(\ell)$ $Option \leftarrow \text{Package}(PrimitiveActions)$ $\text{Annotate}(Option, \text{rationale} = \text{Explain}(\ell))$ $O \leftarrow O \cup \{Option\}$ O

Notes: Expand maps a causal dependency to a minimal behavioral policy. Options inherit rationale from their underlying linkage. This forms the basis for explainability and Δ scoring.

Theory Mapping: Equivalent to HRL option construction. Mirrors APC's generation of goal-conditioned behavioral vectors.

5.3 Glimpse Selection Algorithm

Choosing epistemic actions that reduce uncertainty. A glimpse maximizes expected information gain: $g^* = \arg\max_g \mathbb{E}[I(s; o \mid g)]$.

Glimpse Scheduler [1] **Input:** WorldModel WM, uncertainty vector U **Output:** Glimpse g^* $Target \leftarrow \arg\max_i U[i]$ Select most uncertain linkage $Candidates \leftarrow \text{GenerateProbes}(Target)$ $score(g) \leftarrow \text{ExpectedInformationGain}(g)$ $g^* \leftarrow \arg\max_{g \in Candidates} score(g)$ g^*

Notes: Generate Probes enumerates minimal actions. EIG is approximated using predicted Δ variance. Probabilistic scoring avoids overconfidence and premature convergence.

Theory Mapping: Equivalent to epistemic value in Active Inference. Also resembles Bayesian optimal experimental design.

5.4 Rolling 4-Step Planner Algorithm (MPC)

Short-horizon planning with bounded computational cost.

Rolling MPC Planner [1] **Input:** CurrentState s_t, Options O, Glimpses G **Output:** Concrete next action a_t $H \leftarrow 4$ planning horizon $Plans \leftarrow \text{EnumeratePlans}(O, G, \text{depth} = H)$ each plan π in $Plans$ $\Delta_{pred}[\pi] \leftarrow \text{PredictDelta}(\pi)$ $\pi^* \leftarrow \arg\max_\pi \Delta_{pred}[\pi]$ optimal plan $a_t \leftarrow \text{FirstStep}(\pi^*)$ a_t

Notes: MPC avoids long-horizon delusion and planning fatigue. Δ prediction uses linkage coverage, expected causal effects, and uncertainty. The planner blends epistemic + instrumental value automatically.

Theory Mapping: Equivalent to model-predictive control in robotics. Mirrors hierarchical inference loops in predictive processing.

5.5 Δ (Coherence) Update Algorithm

Structured prediction error as an alignment signal. Recall the Δ definition: $\Delta_t = w_1 C_t + w_2 P_t + w_3 T_t$.

Δ Update Routine [1] **Input:** Previous Δ_t, observed outcome o_t, predicted outcome $\hat{p}_t$ **Output:** Δ_{t+1} $Coverage \leftarrow \text{EvaluateLinkages}(o_t)$ $Accuracy \leftarrow 1 - |o_t - p_t|$ $TTS \leftarrow \exp(-\lambda \cdot \tau_t)$ $\Delta_{t+1} \leftarrow w_1 Coverage + w_2 Accuracy + w_3 TTS$ Δ_{t+1}

Notes: Δ is designed to be monotonic under aligned policies. Negative Δ change flags a causal misalignment. Δ velocity is a stability signal.

Theory Mapping: Δ is a bounded analogue of expected free energy reduction. Accuracy term corresponds to prediction-error minimization. Coverage term mirrors precision-weighted structural satisfaction.

5.6 Policy Update Rule (Belief Revision)

After each executed action, the engine performs an update: $\mu_{t+1} = f(\mu_t, \Delta_{t+1}, o_t)$.
Update Procedure: 1. Update causal confidence for each linkage. 2. Recompute uncertainty vector U. 3. If uncertainty remains high $\rightarrow$ schedule glimpse. 4. If causal model is stable $\rightarrow$ schedule option. 5. Recompute the 4-step plan. This update completes the inference loop.

5.7 Computational Complexity & Boundaries

Planning Complexity: For typical operator workloads (Options $\approx 10 - 40$, Glimpses 3-10, Horizon 4), Plan search complexity is $O((|O| + |G|)^4)$. Bounded by: pruning via Δ predictions, restricting options to top-3, and eliminating dominated sequences.

Inference Complexity: Δ and causal updates are $O(n)$ in linkages, with $n \approx 6-12$. Sequence is computationally tractable in real time.

6 Evaluation & Testable Predictions

Empirical hypotheses and measurable outcomes derived from TCF.

6.1 Overview

A computational framework is only meaningful if it makes testable predictions, can be distinguished from competing models, and improves measurable outcomes in realistic settings. TCF makes a series of quantitative and qualitative predictions about human planning, behavior, cognition, and execution dynamics.

6.2 Cognitive-Level Predictions

Prediction 1: Reduced Cognitive Load via Factorization

Claim: By decomposing goals into Frames and Linkages, TCF reduces the number of simultaneously active cognitive elements from ≈ 7 down to $\approx 2 - 3$ per Frame.

Testable: Compare subjective cognitive load (NASA-TLX) between control group vs. TCF group. Expect significant reduction.

Rooted in theory: Event Segmentation, Hierarchical generative models, Working memory limits.

Prediction 2: Faster Uncertainty Resolution

Claim: Glimpses accelerate epistemic convergence by targeting the highest-uncertainty linkages.

Testable: Measure reduction in entropy of linkage-confidence vector U. Plot uncertainty decay curve.

Expected: Glimpses achieve steeper uncertainty decay than random probing or user-chosen actions.

Rooted in theory: Active Inference epistemic value, Bayesian experimental design.

Prediction 3: Increased Policy Alignment

Claim: With Δ feedback, users converge toward consistent, causally aligned sequences of action.

Testable: Compare Δ velocity under tool-less execution vs. Sequence engine.

Expected: Higher Δ velocity and lower volatility for TCF users.

Rooted in theory: Predictive coding precision, Hierarchical priors.

Prediction 4: Reduction in Drift

Claim: Temporal drift decreases under TCF.

Testable: Behavioral drift metric: $\text{Drift}_t = \|\mu_t - \mu_t^C\|$.

Expected: Drift decreases by 25-60%.

Rooted in theory: PP error minimization, Coherence as predictive consistency.

6.3 Behavioral Predictions

Prediction 5: Rapid Time-to-First-Signal ($\Delta > 0$)

Claim: TCF produces faster positive prediction signals.

Testable: Measure time until Δ becomes positive.

Expected: Median time-to-first $\Delta \leq 48$ hours.

Prediction 6: Higher Execution Throughput

Claim: Short-horizon planning + causal clarity increases weekly executed actions.

Testable: Weekly count of "aligned actions" ($\Delta > 0$).

Expected: 15-40% increase in output.

Prediction 7: Improved Consistency Over Time

Claim: Consistency ($\frac{\text{aligned actions}}{\text{total actions}}$) is higher, with smoother time-series and fewer "breaks" in intentionality.

Prediction 8: Decreased Procrastination

Claim: Epistemic actions reduce fear of the unknown.

Expected: Meaningful reduction in task initiation latency.

6.4 Comparative Predictions Against Existing Tools

Prediction 9: Superior Adaptation to Changing Conditions

Claim: MPC replans after every Δ update; TCF should outperform static plan systems (OKRs, waterfall).

Expected: Faster recovery, less wasted motion.

Prediction 10: Greater Cross-Domain Transfer

Claim: Options reused across Frames should correlate with higher Δ impact and reduced time-to-completion.

6.5 Empirical Metrics for Startup & Product Validation

Sequence inherits strong KPIs: Δ Velocity, Uncertainty Decay Rate, Option Retention / Retirement, Transfer Graph Density, and Drift Reduction. These metrics give Sequence a scientifically grounded, defendable approach to product analytics.

6.6 Safety / Failure Mode Predictions

- **Prediction 11 (Overconfidence Risk):** If Δ is too stable without epistemic action. Fix: enforce periodic glimpses.
- **Prediction 12 (Linkage Over-Specification):** Excessive linkages lead to rigidity. Fix: regularized pruning.
- **Prediction 13 (Action Fatigue):** If uncertainty remains high too long. Fix: glimpses escalate epistemic value.
- **Prediction 14 (Temporal Fragmentation):** Rapid context switching disrupts hierarchy. Fix: force explicit frame boundary detection.

7 Implementation Notes (Engineering Constraints)

How to build Sequence so that it remains faithful to APC + Active Inference + HRL theory.

7.1 Design Objective

The Sequence engine must satisfy three constraints simultaneously: (1) Cognitive Plausibility, (2) Computational Tractability (commodity hardware + LLM inference), and (3) User Experience Integrity.

7.2 Architectural Overview

Layer 1 - Representation (World Model Layer). Stores the user's evolving internal model.

- **Future State:** Text embedding + attention-structured summary.
- **Frames:** Vector clusters + relational tags.
- **Linkages:** Directed graph + confidence values.
- **Uncertainty Vector U:** Distribution over linkage confidence.

Engineering Requirements: Must support incremental updating, remain lightweight, and store structured priors separate from observations. Recommended Data Structures: Document embeddings (768-4096 dim), Typed causal graph, Uncertainty tensor.

Layer 2 - Policy Construction (Option & Glimpse Generation). *System Implementation:*

1. LLM proposes candidate options/glimpses.
2. Sequence engine ranks them using $\text{Score} = \alpha \cdot \Delta_{\text{pred}} + \beta \cdot \text{EIG} - \gamma \cdot \text{Effort}$.
3. Top 1-3 are surfaced.

4. User selects.
5. System logs familiarity, coverage, and Δ prediction.

Layer 3 - Planning (Rolling 4-Step MPC Loop). *Step-by-Step:*

1. Generate candidate 4-step plan.
2. Commit only step 1.
3. Execute and log observation.
4. Compute new Δ.
5. Update world model.
6. Replan.

Engineering Requirements: Must never auto-commit all four steps. Must rebuild plan after each Δ. Must compute Δ locally for integrity.

7.3 Coherence Δ Engine (Mathematical Specification)

Δ is a bounded scalar $\in [0, 100]$:

$$\Delta = 100 \cdot (0.50 \cdot C_{\text{coverage}} + 0.35 \cdot C_{\text{improvement}} + 0.15 \cdot C_{\text{time-to-signal}}) \tag{9}$$

Where Coverage is linkages engaged, Improvement is trajectory match, and Time-to-Signal is feedback speed.

7.4 Learning Dynamics

Sequence must support:

1. **User-Specific Refinement:** Lightweight adapters, priors updated per user.
2. **Option Library Evolution:** Provenance, win-rate, history. Low-performing options auto-retired; high-performing ones transferred.
3. **Structural Learning:** Infer new frame boundaries and causal linkages automatically.

7.5 Data & Telemetry Schema

Core Tables: future_states, frames, linkages, options, glimpses, actions, observations, coherence_scores, provenance_events. Constraints: Append-only, auditable observations, anonymous telemetry.

7.6 LLM Integration Boundary

The LLM should **not** compute Δ, update the world model directly, commit planned steps, or override MPC. The LLM **should** generate candidate options, produce phrasing, infer unknown frames, and critique causal maps. This maintains interpretability, safety, and consistency.

7.7 Performance Requirements

Latency Targets: computation $<$ 200ms, Option ranking $<$ 150ms, Planner $<$ 500ms, UI $<$ 250ms. Storage: World model $<$ 200 KB, Option library $<$ 1 MB.

7.8 Safety & Alignment Considerations

Key Safeguards: Δ alignment checks, Overconfidence detection, Epistemic starvation checks, Context fragmentation warnings, Hyper-precision dampening.

8 Applications & Limitations

What the Temporal Coherence Framework enables in the real world and what it does not.

8.1 Core Applications

LEVEL 1 - Individual Cognitive Control

Use Case: Founder / Operator Execution Engine.
TCF provides:

- causal structure of goals
- MPC-style replanning
- epistemic sampling (glimpses)
- drift reduction
- Δ-anchored behavioral feedback

Measurable outcomes:

- Lower cognitive load
- Higher consistency of action
- Faster adaptation under uncertainty
- Reduced time-to-signal
- More robust planning in dynamic environments

LEVEL 2 - Team & Organizational Coordination

Use Case: Multi-Agent Narrative Coherence.
In teams, Sequence becomes:

- a shared causal model of strategic intent
- a repository of transferable policies (options)
- a Δ-based verification signal for progress
- a way to synchronize beliefs across minds

Analogous to distributed active inference:

- Common future-state prior
- Shared linkage hypotheses
- Cross-agent uncertainty reduction
- Coordinated action under partial knowledge

LEVEL 3 - Adaptive Policy Libraries (Institutional Memory)

Use Case: Enterprise Knowledge Graphs.
Over months of usage:

- Options accumulate
- Transfer graphs reveal patterns
- Linkages stabilize
- Uncertainty decays
- Δ becomes a stable signal

Companies can develop a biologically inspired execution memory.

8.2 Advanced Applications

- **Scenario & Threat Modeling:** Counterfactual simulations and policy stress tests.
- **Scientific & Academic Use Cases:** Cognitive modeling and drift reduction studies.
- **Alignment & Governance:** Practical human-level alignment via transparency and epistemic grounding.

8.3 What TCF Cannot Do (Limitations)

1. **Not a Planning Oracle:** Cannot guarantee optimal paths.
2. **Not a Substitute for Domain Knowledge:** Enhances expertise but does not generate it.
3. **LLM Boundaries:** Must not delegate policy commitment to LLMs.
4. **Human Fatigue & Ambiguity:** Users may mislabel observations or avoid glimpses.
5. **High-Frequency Context Switching:** Can destabilize plans.
6. **Not Suitable for Long-Horizon, Static Plans:** Designed for dynamic environments, not deterministic waterfalls.

8.4 Competitive & Scientific Positioning

Compared to productivity tools (which track tasks), TCF models causality. Compared to AI assistants (which output suggestions), TCF performs uncertainty-aware planning loops. Compared to OKR platforms (which define goals), TCF verifies them with Δ. Compared to Active Inference research systems, TCF provides a consumer-grade abstraction layer.

8.5 Summary: Why Sequence Validates Itself

The engine makes falsifiable predictions and derives testable improvements. Its architecture has a defensible theoretical foundation, exposes interpretable causal state, supports adaptive behavior, controls drift, tracks uncertainty, and enforces coherence.

9 Discussion, Future Work, and Implications

Where Sequence goes next scientifically, technically, and societally.

9.1 Overview: Sequence as an Applied Cognitive Architecture

Sequence is not merely a productivity framework nor a planning system. It is a consumer-facing instantiation of a formal cognitive architecture, drawing directly from:

- Active Predictive Coding (APC)
- Hierarchical Reinforcement Learning (HRL)
- Model-Predictive Control (MPC)
- Predictive Processing
- Active Inference
- Structure Learning & Option Discovery

TCF is best understood as a practical, human-centered implementation of hierarchical predictive control.

9.2 Implications for Cognitive Augmentation

Sequence suggests a concrete answer to how human decision-making can be improved with minimal overhead. TCF addresses this by:

- compressing complex task structures into a causal model,
- maintaining a stable goal-conditioned prior,
- using epistemic actions (glimpses) to reduce uncertainty,
- enabling short-horizon planning loops akin to MPC,
- quantifying progress through Δ, a coherence-based score.

Cognitive prosthetics should resemble predictive-control systems, not checklists.

9.3 Implications for AI Alignment (Human-Level)

Sequence demonstrates that alignment is not solely a problem for autonomous AI. Humans also require stable goals and interpretable decision paths. TCF forces the user to:

- articulate priors (Future State),
- explicitly model causal structure,
- track epistemic uncertainty,
- update beliefs transparently after observations.

This acts as an alignment pipeline for human reasoning.

9.4 Implications for Enterprise & Organizational Intelligence

Sequence produces emergent properties:

1. **Collective Predictive Control:** Members operate within a synchronized prediction loop.
2. **Organizational Reduction of Epistemic Uncertainty:** Glimpses scale: marketing $\rightarrow$ sales $\rightarrow$ product $\rightarrow$ leadership.
3. **Accelerated Institutional Memory:** Options accumulate as reusable playbooks.

4. **Cross-Frame Generalization:** Successful tactics in one domain inform others.

It is an applied version of multi-agent active inference.

9.5 Implications for AI Co-Pilots

TCF offers a scaffolding for grounded agent behavior. TCF provides:

- the goal-conditioned prior
- the causal model
- the planning horizon
- the epistemic sampling
- the verification signal (Δ)

It makes Sequence a prime candidate for human-AI hybrid MPC loops.

9.6 Limitations Future Work

Future work includes:

- **Automated Structure Learning:** Using LLMs or Bayesian network induction.
- **Adaptive Δ Weighting:** Per-user priors and learned weight adaptation.
- **Multi-Agent Coherence Models:** Tracking team Δ and cross-frame Δ.
- **Higher-Order Glimpse Policies:** Multi-step epistemic policies.
- **Potential for Formal Verification:** Verifiable plans and safe-mode behaviors.

9.7 Broader Impact

Sequence proposes a new category: **APC Execution Systems**. It posits that "Humans improve when their predictions improve." Sequence is an engine for improving predictions, positioning it as a theoretical contribution to cognitive engineering.

10 Conclusion

This white paper has articulated the Sequence Engine as a rigorous, theoretically grounded implementation of Active Predictive Control (APC) for human goal pursuit. Integrating principles from predictive processing, active inference, and hierarchical reinforcement learning, the Engine reframes execution as a continual process of modeling, prediction, and adaptive action—rather than a sequence of tasks or intentions. At its core lies a simple but powerful premise: progress emerges when individuals maintain a generative model of what must be true, systematically reduce uncertainty, and update action as reality unfolds.

Sequence operationalizes this premise through a structured architecture: Future State priors that anchor intention; Frames that decompose the world into meaningful domains; Linkages that express the causal scaffolding of progress; Options that encode reusable policies; Glimpses that target high-value uncertainty; a rolling model-predictive controller that adapts action on short horizons; and Coherence Δ, a transparent, interpretable measure of alignment between predicted and observed progress. Each module is computationally meaningful. Together, they function as a unified predictive system—one capable of stabilizing direction, accelerating learning, and producing measurable movement toward a defined state.

Central to this framework is the interpretation of human action as an inference problem. Instead of asking, "What should I do?", the Engine asks, "What must I know, test, or change to bring my current state into alignment with my future state?" This perspective elevates execution from effort to epistemics. Drift becomes a detectable signal; progress becomes a quantifiable reduction of prediction error; and action becomes a means of refining the world model rather than blindly pursuing output. In this sense, Sequence embodies the same computational principles that underlie adaptive biological systems: prediction, feedback, continuous updating, and structure-preserving generalization.

This architecture does more than organize behavior—it enables transfer. Linkages learned in one Frame can be reused in another. Options that succeed in one domain become general-purpose policies. Δ trajectories reveal behavioral signatures that can be extended across goals, teams, and contexts. Over time, the system becomes not only a guide for execution, but a reservoir of accumulated structure: a growing library of causal knowledge that strengthens the user's capacity to operate effectively under uncertainty.

As a result, the Sequence Engine is both a product and a research contribution. It translates predictive-control theory into an accessible, observable, and high-resolution system for human agency. It provides the scaffolding for individuals to reason more coherently, act more strategically, and adapt more fluidly as circumstances evolve. And it establishes a foundation for future work in computational planning, structure learning, and applied active inference.

The broader vision is straightforward: to give individuals a predictive architecture for their own lives—one that makes progress visible, coherence measurable, and adaptive intelligence a practical, everyday capability. Sequence is designed to do more than help people "get things done." It provides a framework for navigating uncertainty with clarity, for turning intention into structured action, and for building a continually improving model of how one achieves meaningful outcomes in complex environments.

In this way, Sequence offers a new category of execution system: not a productivity tool, but a predictive engine for human advancement—grounded in scientific theory, built for real-world use, and designed to scale with the individual's ambition.