Hierarchical-Temporal Feelings for Structural Self-Management in Emotion-Guided World Models

Melissa Howard*
melhoward@live.ca

December 2025

Abstract

Most contemporary modular and tool-using AI systems grow experts, prune paths, and call external tools, but almost always under hand-engineered or externally scheduled control. This leaves open a central question for Artificial General Intelligence (AGI): how should an autonomous agent manage and optimise its own internal cognitive structure over a long, non-stationary lifetime?

We formulate this as the *Autonomy–Architecture Problem* and propose *Hierarchical-Temporal Feelings* (HTF) as an architectural solution. HTF separates fast *epistemic* intelligence—local truth-seeking about what is known and how well it is known—from slow *teleological* intelligence—global strategic assessments of whether the current bank of heads and tools is structurally sufficient. Fast feelings such as Predictive Discrepancy (PD) and Head Redundancy (HR) drive local structural edits (SPAWN, UPDATE, MERGE, FORGET), while slow feelings such as family-specific Meta-Frustration MF_c and Structural Endurability $\mathrm{SE}_{\mathrm{Vec}}$ drive global actions (TOOL-ACQUIRE, CONSOLIDATE, and updates to an inner voice $\mathbf{W}$).

To stress-test this architecture, we introduce the Hierarchical Continual Learning Benchmark (HCLB), a simple but adversarial task sequence that mixes growth, compression, and tool-necessary regimes. In a toy yet fully specified setting, we compare a Flat Global agent, a stronger Flat-Plus agent that sees the same rich signals as HTF, and the proposed HTF agent. Flat controllers either never acquire tools or do so at excessive structural cost, while HTF reliably triggers tool acquisition and maintains a more compact head bank, yielding higher structural reward. These results support the view that structural stability is an emergent property of hierarchically routed feelings and that HTF provides a concrete blueprint for metacognitive, self-governing AGI.

1 Introduction

Modern large-scale systems—Mixture-of-Experts architectures, tool-using language models, and modular continual learners—increasingly resemble self-modifying agents: they grow new experts, prune unused pathways, and acquire external tools. Yet their structural updates are almost always hand-engineered or externally scheduled. This raises a deeper question for Artificial General Intelligence (AGI): *how should an autonomous system manage and optimise its own internal cognitive structure over a long, non-stationary lifetime?*

We refer to this challenge as the **Autonomy–Architecture Problem**. It asserts that an AGI must possess a metacognitive architecture capable of regulating its own complexity, balancing conflicting needs for efficiency, growth, and specialisation as the world changes.

*Independent researcher.

1.1 The Structural Dilemma

Any modular, self-modifying agent (for example, a bank of experts plus tools) is subject to at least three structural pressures, each with its own objective and characteristic timescale:

Structural pressure	Strategic objective	Timescale of need
Growth / SPAWN	Competence, adaptability	Short-to-medium
Compression / MERGE	Efficiency, stability	Short-to-medium
Tool-ACQUIRE / CONSOLIDATE	Structural sufficiency, specialisation	Long-term

In a flat, single-controller architecture, all three must be negotiated by a single structural policy. This induces a *structural dilemma*: the agent is forced into an unstable compromise between short-term efficiency and long-term competence. Empirically, such systems tend to be either *brittle* (overgrown and resource-intensive) or *stubborn* (overly compressed and under-competent) when faced with structural change. The larger conclusion is that *structural stability is an emergent property of hierarchical, decoupled control, rather than a direct outcome of a single global optimisation.*

1.2 Hierarchical-Temporal Feelings (HTF): Architectural Solution

We propose **Hierarchical-Temporal Feelings (HTF)** as an architectural solution to the Autonomy–Architecture Problem. HTF partitions structural steering signals into two layers operating at different temporal resolutions, thereby addressing the credit assignment problem for structural actions. The key assertion is that local edits and global strategy must be governed by separate, isolated value streams.

Layer 1: Epistemic intelligence (fast learner). Layer 1 governs local knowledge quality and internal truth-seeking. It sees fast epistemic signals such as Predictive Discrepancy (PD) (a proxy for surprise or curiosity) and Head Redundancy (HR) (a proxy for cognitive load or inefficiency). These signals drive fast, reactive structural edits (SPAWN, UPDATE, MERGE, FORGET) and answer the question: *"What do I know here, and how well do I know it?"*

Layer 2: Teleological intelligence (slow strategist). Layer 2 governs global strategic health and architectural validity. It sees slow, teleological signals such as family-specific Meta-Frustration $\mathbf{MF}_c$ (persistent, systemic failure) and Structural Endurability $\mathrm{SE}_{\mathrm{Vec}}$ (long-horizon efficiency and stability). These signals drive slow, reflective structural shifts (TOOL-ACQUIRE, CONSOLIDATE, and updates to the inner voice $\mathbf{W}$), answering the question: *"Is my current cognitive strategy—my bank of heads and tools—fundamentally sufficient for this class of problems?"*

By routing the clean, isolated signal $\mathbf{MF}_c$ (a history of persistent failure restricted to task family c) to the slow strategist in Layer 2, HTF ensures that urgent short-term efficiency pressures in Layer 1 cannot mask the long-term need for structural evolution in Layer 2. Local compression can be aggressive within an architectural regime, while teleological signals retain the authority to add tools or change the regime itself.

1.3 From Self-Fixing to Self-Governing

Viewed this way, HTF is a framework for computational metacognition. It does not merely make a learner more robust; it provides an architectural mechanism for an agent to step back from its immediate learning task and assess its own structural limits.

The family-specific Meta-Frustration signal $\mathbf{MF}_c$ is the key driver of this metacognitive leap. It encodes the fact that local epistemic fixes (SPAWN and MERGE within the current head bank) have persistently failed for a given family of problems. Once $\mathbf{MF}_c$ exceeds a threshold, the system is forced into *strategic escalation*: from internal structural repair to external structural acquisition (for example, TOOL-ACQUIRE or new module creation).

The central claim of this paper is that HTF transforms a system from a self-fixing learner into a *self-governing* entity: an agent that can evolve its own architecture in a disciplined, timely manner over a long and unpredictable lifetime. The remainder of the paper makes this claim concrete in a toy but fully specified setting, via the Hierarchical Continual Learning Benchmark (HCLB) and comparisons between Flat Global, Flat-Plus, and HTF agents.

Contributions. Concretely, this paper:

- formulates the **Autonomy–Architecture Problem** as the central structural challenge for self-modifying AGI;
- introduces **Hierarchical-Temporal Feelings (HTF)**, which decouples fast **epistemic** and slow **teleological** intelligence across layers and timescales;
- proposes the **Hierarchical Continual Learning Benchmark (HCLB)**, designed to expose the structural dilemma between growth, compression, and tool-acquisition;
- presents toy but fully specified experiments comparing Flat Global, Flat-Plus, and HTF agents, showing that only hierarchical routing achieves both tool-level competence and compact structure.

2 Background

2.1 Emotion-Guided World Models

Emotion-Guided World Models (EGWM) start from the idea that a learner should not treat all data and all updates as equal. Instead, it should use low-dimensional value signals or "feelings" to decide when and how to change itself. In earlier work, EGWM maintained a bank of specialist heads, each trained on a particular regime or "world", and endowed this bank with structural actions:

- SPAWN: create a new head when existing heads cannot explain a consistent phase of data;
- UPDATE: adapt a head when it is competent but still improving;
- MERGE: compress similar heads when they redundantly cover the same region of input space;
- FORGET: retire obsolete heads that no longer contribute.

Simple value channels such as confusion, competence, and a crude elegance proxy (penalising too many overlapping heads) acted as feelings that guided these actions. Even with linear models in low-dimensional toy worlds, this value-guided structural control prevented catastrophic forgetting, reduced label usage, and achieved higher per-world accuracy than a single monolithic learner.

2.2 Reflective Structural Control

Hierarchical-Reflective EGWM (HR-EGWM) extended this setup with a reflective layer. A Model-Predictive Governor imagined the future consequences of structural actions, and a small Adaptive Inner Voice $\mathbf{W}$ reweighted feelings to favour different trade-offs in different environments. For example, in stable stationary worlds the inner voice could up-weight elegance, favouring compression, while in non-stationary worlds it could up-weight growth and confusion.

However, this reflective control remained *flat*. A single controller, with a single vector $\mathbf{W}$, still had to negotiate all structural pressures at once. When tasks demanded both aggressive compression and timely tool acquisition, this flat decision maker was forced into unstable compromises.

2.3 From Feelings to Hierarchical-Temporal Feelings

HTF builds directly on these ideas. It keeps the intuition that low-dimensional value signals can act as feelings that steer structural change, but changes the *architecture* through which those feelings are routed. Instead of a single vector of mixed signals, HTF organises feelings into fast, local epistemic channels and slow, global teleological channels, each connected to its own controller. The following sections formalise these channels and describe how they are used.

3 Hierarchical-Temporal Feelings Architecture

3.1 Local Epistemic Feelings and the Fast Controller

HTF assumes a bank of heads $\{h_i\}$, each a simple model (for example, a logistic regressor or small neural network) with parameters θ_i. At each phase t, the agent receives a batch of labelled examples $D_t = \{(x, y)\}$ drawn from one of several hidden task families.

For each head h_i and phase t, we compute a local feeling vector $v_{t,i}^{\text{local}}$ that summarises how well that head understands the current data and how redundant it is with respect to the bank.

Predictive discrepancy. Let $p_i(y \mid x)$ be the predictive distribution of head i on input x, and let $\hat{y} = \arg\max_{y'} p_i(y' \mid x)$ be the predicted class. The per-sample predictive discrepancy is

$$\mathrm{PD}_i(x, y) = \max\{0,\ p_i(\hat{y} \mid x) - p_i(y \mid x)\}.$$

This quantity is large when the head is confidently wrong and small when it is either correct and confident or appropriately uncertain. The per-phase predictive discrepancy is

$$\mathrm{PD}_{t,i} = \frac{1}{|D_t|} \sum_{(x,y)\in D_t} \mathrm{PD}_i(x, y).$$

High $\mathrm{PD}_{t,i}$ signals that the head's world model is miscalibrated in this region.

Head redundancy. To measure representational overlap, we let each head expose a feature vector $\mathbf{f}_i(x)$ (for example, logits or a hidden layer) and define

$$\mathrm{HR}_{t,i} = \max_{j\neq i} \frac{\sum_{(x,y)\in D_t} \mathrm{sim}(\mathbf{f}_i(x), \mathbf{f}_j(x))}{|D_t|},$$

where sim is a cosine similarity. High redundancy means that some other head already behaves similarly on this batch.

Local competence and usage. Alongside PD and HR, we track a smoothed per-head competence $\mathrm{acc}_{t,i}$ (for example, an exponential moving average of accuracy) and a usage statistic $\mathrm{usage}_{t,i}$ that counts how often head i is routed inputs during evaluation.

Local feeling vector and actions. These signals are concatenated into

$$v_{t,i}^{\text{local}} = \big[\text{PD}_{t,i},\, \text{HR}_{t,i},\, \text{acc}_{t,i},\, \text{usage}_{t,i}\big].$$

A fast controller consumes the collection $\{v_{t,i}^{\text{local}}\}$ and chooses structural edits for the head bank:

- SPAWN: create a new head initialised from a scratch model or a template if all existing heads exhibit high PD on D_t;
- UPDATE: train a selected head on D_t when PD is moderate and competence is improving;
- MERGE: merge two redundant heads when HR is high but accuracy is stable;
- FORGET: retire heads with low usage and low relevance.

These decisions operate on a fast timescale, reacting to each phase.

3.2 Global Teleological Feelings and the Slow Controller

Local epistemic signals tell the agent how well it understands a particular region of data, but they do not answer the question of whether its entire architecture is sufficient for a family of tasks. For this, HTF introduces slower, aggregated teleological feelings.

Structural endurability. Over a macro-episode consisting of many phases, the agent accumulates statistics about its structural edits and their consequences. Two components are:

- *Efficiency*:
$$\text{SE}_{\text{eff}} = 1 - \alpha \cdot \frac{\#\text{SPAWN} + \#\text{failed\_MERGE}}{T},$$
where T is the number of phases, and α is a scaling factor. This term decreases when the agent grows too many heads or frequently attempts merges that must be rolled back due to accuracy loss.
- *Stability*:
$$\text{SE}_{\text{stab}} = 1 - \beta \cdot \mathbb{E}\big[\Delta\text{Acc}^{-}\big],$$
where ΔAcc^{-} is the drop in validation accuracy immediately after MERGE or FORGET actions, and β is a scaling constant. This term decreases when compression causes catastrophic forgetting.

These components are combined into a vector

$$\text{SE}_{\text{Vec}} = [\text{SE}_{\text{eff}},\, \text{SE}_{\text{stab}}].$$

Meta-frustration per task family. The key teleological feeling in HTF is family-specific Meta-Frustration. For each task family c, we consider the set of phases $\mathcal{T}_c$ that belong to that family and define

$$\text{MF}_c = \frac{1}{|\mathcal{T}_c|} \sum_{t \in \mathcal{T}_c} \min_i \text{PD}_{t,i}.$$

This quantity is high when the *best* head in the bank continues to show high predictive discrepancy on that family, even after structural edits. It therefore captures persistent, systemic failure that local repairs have not resolved.

Global feeling vector and actions. At the end of a macro-episode, HTF forms a global feeling vector

$$v_T^{\text{global}} = \big[\text{SE}_{\text{Vec}},\ \{\text{MF}_c\}_c,\ \text{avg\_heads},\ \text{avg\_competence}\big].$$

A slow controller consumes v_T^{global} and chooses among global actions such as:

- TOOL-ACQUIRE: attach an external tool for family c when MF_c remains high despite many structural edits;
- CONSOLIDATE: retrain routers, rebalance head assignments, or reinitialise parts of the bank;
- update of the inner voice $\mathbf{W}$ that reweights local feelings in future episodes.

Crucially, this slow controller never sees raw minibatch data. It sees only summarised feelings about structural health and family-specific failure.

3.3 Decoupling Epistemic and Teleological Intelligence

By construction, Layer 1 (epistemic) answers the question *"What do I know here, and how well do I know it?"* and acts at the level of local structural edits. Layer 2 (teleological) answers the question *"Is my current cognitive strategy sufficient for this class of problems?"* and acts at the level of tools and architectural regimes.

This decoupling prevents short-term efficiency pressures from silencing long-term frustration signals. Even if Layer 1 aggressively compresses heads within a regime, Layer 2 still sees persistent high MF_c for unsolved families and can escalate to TOOL-ACQUIRE.

4 Hierarchical Continual Learning Benchmark

To test HTF, we introduce the Hierarchical Continual Learning Benchmark (HCLB), a simple yet adversarial sequence of phases designed to expose the structural dilemma and the Autonomy–Architecture Problem in toy form.

4.1 Task Families

HCLB consists of three task families:

- **Family A (Growth).** Non-stationary pattern worlds that are structurally solvable by the heads. Examples include rotating linear decision boundaries in two dimensions. Optimal behaviour is to SPAWN and UPDATE new heads as needed.
- **Family B (Tool-necessary).** A symbolic or non-linear task for which the simple heads are structurally insufficient. Accuracy is capped at about 0.75 under the given capacity and training budget, regardless of how many heads are spawned. Closing the gap requires TOOL-ACQUIRE.
- **Family C (Redundancy).** Repeats or slight variants of old A-style tasks. Many heads become redundant in these phases. Optimal behaviour is to MERGE or FORGET redundant heads without damaging competence.

4.2 Episode Structure

Each macro-episode consists of a fixed sequence of 30 phases:

- 10 phases from Family A (Growth),
- 10 phases from Family B (Tool-necessary),
- 10 phases from Family C (Redundancy).

This pattern is repeated across macro-episodes with different random world parameters. The key property is that structural pressures conflict within each episode: A encourages growth, B demands tool acquisition, and C rewards compression.

4.3 Agents Compared

We compare three agents:

- **Flat Global.** A single flat controller that observes a global error signal (a moving average of loss across all families), the number of heads, and a simple structural cost. It chooses SPAWN, UPDATE, MERGE, FORGET, and TOOL-ACQUIRE using a single policy and a single set of weights.
- **Flat-Plus.** A stronger flat baseline that observes the same rich signals as HTF, including family-specific MF_c and $\mathrm{SE}_{\mathrm{Vec}}$, but still uses a single controller and a single weighting of all signals to choose all structural actions.
- **HTF.** The proposed architecture with separate fast and slow controllers, local $v_{t,i}^{\mathrm{local}}$ feelings for Layer 1, and global v_T^{global} for Layer 2.

4.4 Structural Reward

All agents are trained or tuned to maximise the same episode-level structural reward:

$$R_{\mathrm{struct}} = \sum_t \mathrm{Acc}_t - \lambda_{\mathrm{complexity}} \sum_t |\mathcal{H}_t|,$$

where Acc_t is validation accuracy at phase t, $|\mathcal{H}_t|$ is the number of heads at phase t, and $\lambda_{\mathrm{complexity}}$ is a constant that penalises complex structures. Each agent is run for multiple episodes under this reward, and its parameters are adjusted by simple search or meta-optimisation to improve R_{struct}.

5 Toy Experiments and Results

We implement a simple simulation of HCLB in which heads are abstract skills with limited capacity. Rather than fully training neural networks in this first study, we directly model the achievable accuracies under different conditions, focusing on the qualitative behaviour of the structural controllers.

5.1 Simulation Setup

In the simulation:

- Family A phases are solvable by heads, with potential accuracy close to 1.0 if enough specialised heads are spawned.
- Family B phases are structurally unsolvable by heads alone: even with many heads and updates, accuracy cannot exceed about 0.75 due to a fixed capacity limit.
- Family C phases are variations on earlier A-style tasks, so many heads become redundant and can be merged or forgotten without harming accuracy.

The Flat Global agent uses a single global error signal (averaged over all families) to decide all structural actions, including TOOL-ACQUIRE. The Flat-Plus agent sees richer features, including per-family error and meta-frustration, but still drives all actions with a single controller. The HTF agent uses separate controllers as described earlier.

Agent	B accuracy	Tool-acq. rate	Avg. heads	Struct. reward
Flat Global	≈ 0.73	0%	low	moderate
Flat-Plus	≈ 0.90	intermediate	high	moderate
HTF	≈ 0.96	high	medium	high

Table 1: Illustrative toy HCLB outcomes. Flat Global remains at the head ceiling on Family B and never acquires tools. Flat-Plus can exploit family-specific meta-frustration but only by maintaining a bulky structure. HTF achieves tool-level accuracy on Family B with a more compact head bank, yielding higher structural reward.

5.2 Qualitative Behaviour

Across repeated runs, the following qualitative patterns emerge:

- In Family A, all agents can learn to spawn and update heads.
- In Family B, the Flat Global agent often continues to spawn and update heads but fails to recognise that the head family is structurally capped, so it rarely or never chooses TOOL-ACQUIRE.
- The Flat-Plus agent can, in principle, use MF_B to detect persistent failure in Family B, but compression pressures from Family C compete with this signal. TOOL-ACQUIRE is triggered only in some runs or after long delays.
- The HTF agent keeps the failure signal MF_B isolated in Layer 2. Once MF_B remains high across several episodes despite many structural edits, the slow controller reliably triggers TOOL-ACQUIRE for Family B.

5.3 Summary of Results

Table 1 summarises typical outcomes over multiple macro-episodes for the three agents.

The Flat Global agent never acquires the tool in this simulation. It remains at the head ceiling on Family B because improvement on Families A and C reduces the global error signal, drowning out the structural failure in Family B. The Flat-Plus agent can sometimes acquire the tool, but only by learning a compromise policy that tolerates many heads, thus sacrificing structural efficiency.

The HTF agent, by contrast, consistently acquires the tool for Family B after a finite number of episodes. It interprets high MF_B as evidence that head-based fixes are exhausted and escalates to TOOL-ACQUIRE, while still allowing Layer 1 to compress redundant heads within solvable families.

In summary, this toy HCLB setting shows that the Autonomy–Architecture Problem already appears in minimal form, and that hierarchical routing of feelings, as proposed by HTF, is sufficient to resolve the resulting structural dilemma where flat controllers fail or pay unnecessary structural cost.

6 Discussion

The experiments above are deliberately minimal, but they already exhibit the core features of the Autonomy–Architecture Problem. Even in this toy setting, a single flat controller cannot comfortably balance growth, compression, and tool acquisition. When pressured to be efficient, it underreacts to persistent failure; when pressured to be ambitious, it overgrows and fails to compress.

HTF resolves this by giving different kinds of intelligence their own channels and controllers. Fast epistemic intelligence controls local structure: it asks whether a particular region of data is understood and acts quickly to grow, adapt, or compress heads. Slow teleological intelligence controls global structure: it asks whether a task family is fundamentally solvable with the current architecture and, when not, escalates to tools or new modules.

This separation naturally aligns with intuitive notions of feelings. Fast feelings such as curiosity and confusion belong to Layer 1, while slower feelings such as deep frustration and structural doubt belong to Layer 2. HTF can therefore be read as a simple computational story about how an agent might begin to have *feelings about its own way of being* and act on them.

At the same time, the present study is limited. The experiments use abstract heads rather than fully trained neural networks, and they assume known task families in order to define MF_c. Extending HTF to richer domains will require both more realistic environments and methods for discovering task families automatically.

7 Conclusion

We introduced Hierarchical-Temporal Feelings (HTF) as an architectural response to the **Autonomy–Architecture Problem**: the challenge of designing agents that can manage and optimise their own cognitive structure over a long, non-stationary lifetime. HTF separates fast epistemic intelligence (local truth-seeking) from slow teleological intelligence (global strategic steering), routing their respective feelings through distinct controllers that operate at different timescales.

In a simple but adversarial Hierarchical Continual Learning Benchmark (HCLB), we showed that flat controllers face a structural dilemma when simultaneously confronting growth, compression, and tool-acquisition pressures. A Flat Global agent, driven only by a single global error signal, remains below a structural performance ceiling and never acquires tools. A stronger Flat-Plus baseline, given access to the same rich signals as HTF, can exploit family-specific meta-frustration MF_B to acquire a tool and achieve high competence, but does so with a bulkier, less efficient structure. HTF achieves the same tool-level competence while maintaining a more compact head bank, yielding higher structural reward. In this setting, structural stability emerges only when epistemic and teleological feelings are hierarchically decoupled.

We view this work as a proof-of-concept for a broader thesis: that AGI systems will require not just large world models, but explicit architectures for **self-governance**. HTF offers one such blueprint. It provides a concrete mechanism by which an agent can recognise the limits of its current ontology, distinguish between local repair and structural insufficiency, and escalate to new tools or modules when necessary. Future work can scale HTF to richer neural implementations and benchmarks, relax the assumption of known task families, and explore learned value channels. The central idea, however, remains simple: *structural autonomy requires hierarchical feelings about both what is known and whether the current way of being is enough.*