EGWM: Feeling the AGI

Multi–Time-Scale Value Signals for Growth, Compression, and Near-Perfect Competence in Toy Worlds

Melissa Howard
melhoward@live.ca

December 7, 2025

Abstract

Current large models are powerful pattern recognizers but remain brittle: they forget, they miscalibrate confidence, and they lack any explicit notion of "how much this situation matters" or "how aggressively to change themselves." In earlier work (EGWM Parts I–III) we proposed a simple architecture — Emotion / Value-Guided World Models (EGWM) — built around a world model P_θ, a multi-channel value / "feeling" module V_ϕ, a governor G_ψ, and a memory updater U_η controlling a bank of specialist heads. We showed that fast value signals can gate plasticity and spawning, while slower ones can drive compression and forgetting, preventing catastrophic interference in simple toy tasks.

In this new paper we synthesize those ideas and extend them with a sequence of concrete experiments. First, we introduce *time-structured relevance* signals that combine age and task importance to decide which specialists to retire. Second, inspired by Sutton's Oak architecture, we let confusion and novelty trigger feature growth and subproblem creation. Third, we treat value signals as drivers of *information-seeking behaviour*: an uncertainty channel controls when the agent queries labels, while a relevance channel decides which worlds should be protected from forgetting. Finally, we evaluate a variant with small neural heads on non-linear tasks and show that a simple uncertainty "feeling" can reach essentially the same accuracy as an always-query baseline while using a fraction of the labels.

Across these toy settings, one consistent picture emerges: (1) capacity must match the environment; (2) once it does, a small set of multi-time-scale value signals — confusion, uncertainty, competence, elegance, and relevance — are sufficient to grow a modular world-bank, compress it safely, selectively forget unimportant structure, and act to obtain better data, all while maintaining near-perfect competence on important tasks.

1 Introduction

Scaling up monolithic models has delivered remarkable capabilities, but it has not solved three old problems: catastrophic forgetting, poor calibration in novel regimes, and the lack of principled mechanisms for deciding *when* to change internal structure. In animals, these decisions are not made by the cortex alone. Fast, low-dimensional "emotional" systems modulate learning rates, exploration, risk-taking and long-term consolidation.

In earlier work (EGWM Parts I–III) we explored a minimal analogue of this idea. We coupled a simple world model with a value / "emotion" module and a governor that could choose, on each phase of experience, whether to learn, to spawn a new specialist head, to merge specialists, or to freeze. The core qualitative result was that:

- Fast phase-consistency signals can prevent learning from junk phases and reduce interference.

- A bank of heads ("worlds") plus a confusion signal can localize learning and avoid global overwriting.
- Slower competence and elegance signals can gradually compress the bank by merging redundant heads, if merges are constrained not to reduce performance.

In this paper we ask: *Given this architecture, what is the best way to approach "100% competence" in simple worlds, without simply turning off forgetting and querying labels everywhere?* We answer this by integrating and extending three themes:

1. **Relevance and forgetting**: introduce age-aware and reward-weighted relevance signals to decide which heads can be safely forgotten.
2. **Feature and option growth**: incorporate an Oak-like pipeline where confusion triggers new features and subproblems, not just new heads.
3. **Information-seeking**: use value signals (especially uncertainty) to decide when to query labels or seek extra information, and combine this with relevance so that high-importance worlds are preserved.

We work entirely in toy settings — low-dimensional classification tasks composed of multiple "worlds" with different decision boundaries. Despite their simplicity, these experiments expose non-trivial trade-offs: capacity versus elegance, recency versus importance, and passive versus active data collection. Our conclusion is not that we have built an AGI, but that a particular shape of architecture looks promising:

> A sufficiently expressive world-bank, paired with multi-time-scale value signals that modulate growth, compression, forgetting and querying, can maintain near-perfect competence on important tasks while using far less data and capacity than naive baselines.

2 Background and Prior EGWM Work

2.1 Core EGWM equation

We frame EGWM as a recurrent decision system acting on a stream of inputs x_t and producing outputs y_t with an internal memory state m_t. The core update is:

$$(y_t, m_{t+1}) = U_\eta\Big(m_t,\ x_t,\ G_\psi\big(x_t,\ P_\theta(x_t,\ m_t,\ \Pi_\omega(x_t, m_t),\ T_\gamma(x_t, \Pi_\omega, m_t)),\ V_\phi(\cdot)\big)\Big) \tag{1}$$

where:

- P_θ is the main world model / predictor.
- V_ϕ is a value / "emotion" module that outputs multiple channels (reward, risk, novelty, uncertainty, elegance, relevance).
- G_ψ is a governor that chooses an internal action: answer, think more, spawn a head, merge heads, query the environment, etc.
- U_η updates the memory and the parameters of heads, according to the governor's choice.
- Π_ω is a planner and T_γ a set of tools; in this paper we mostly work in simple supervised settings and use these implicitly.

In practice, most of our experiments instantiate this equation in a stripped-down form: a bank of specialist heads, a confusion / uncertainty signal, and a simple governor that decides which head to update, when to spawn a new one, when to merge or prune heads, and when to query labels.

2.2 Parts I–III: from phase gating to world banks

Part I: phase-consistency gating. We first studied a single model learning across discrete phases (e.g., tasks A, B, C). A fast *phase-consistency* feeling judged whether a phase was internally self-consistent; if not, updates were suppressed. Even in simple logistic regression tasks this reduced interference and prevented learning from adversarial or mixed phases.

Part II: world bank and feelings. We then moved to a *world bank*: a pool of heads, each trained to specialize on a certain regime. A confusion signal drove the spawning of new heads when no existing head could explain current data; a competence signal kept track of how well each head performed; an elegance signal encouraged using fewer, more general heads. We found that:

- "Spawn-happy, compress later" worked better than aggressively enforcing elegance from the start.
- Naive elegance rewards (e.g., regularizing the number of heads) could collapse learning by discouraging new specialists.

Part III: time-structured grow–compress–forget. Finally, we introduced explicit time structure:

- Fast confusion $\Rightarrow$ spawn new heads and route data to them.
- Intermediate competence + elegance $\Rightarrow$ merge heads only if global performance does not drop.
- Slow relevance (implicitly) $\Rightarrow$ allow the system to gradually retire heads that are never used.

We also annealed the strength of elegance over time: early life is allowed to be messy and over-parameterized, while later life is pressured to be more compact.

2.3 Oak architecture and options

In parallel, recent work (e.g., Sutton's "Oak" architecture) has argued that an agent should grow its own structure from experience: discover features, define subproblems, learn options and models, and then curate them. Conceptually, EGWM can be seen as an "Oak with feelings": we explicitly model the value channels that might decide when to grow or prune this tree of options and models.

3 Architecture in This Paper

For clarity, we summarize the concrete instantiation of EGWM used in this paper's experiments.

3.1 World-bank heads

We consider a set of K heads $\{h_k\}_{k=1}^{K}$, where each head is either:

- a logistic regression model on hand-crafted features (linear or quadratic), or
- a tiny multi-layer perceptron (e.g., 2–8–1).

Each head is intended to specialize on a particular "world" (task). In most experiments we use oracle-style routing during evaluation: given an input, we pick the head with lowest loss or highest confidence. This isolates the effect of growth / compression / forgetting from the problem of online world-identification.

3.2 Feelings and time scales

We instantiate several feelings as simple scalar statistics:

- **Confusion**: average cross-entropy loss of the best head on the current batch or segment.
- **Uncertainty**: for a binary prediction p, $u = \min(p, 1 - p)$.
- **Competence**: rolling estimate of accuracy or loss per head.
- **Elegance**: a pressure to use fewer heads, implemented indirectly: we only merge or prune when it does not harm competence.
- **Relevance**: a slow signal combining *age* (time since a head was last used) and *importance* (cumulative reward/weight of tasks served by that head).

These feelings operate on different time scales:

Fast (per batch/step): confusion, uncertainty. Used to decide when to spawn heads, when to query labels.

Medium (per episode / periodically): competence, elegance. Used to decide when to merge heads.

Slow (across many segments): relevance. Used to decide when to retire or distill heads.

3.3 Governor and updater in toy form

The governor G_ψ is instantiated as a set of simple rules:

- If confusion is high and no head explains the current segment, spawn a new head and train it.
- Otherwise, route updates to the best-competent head.
- Periodically, consider pairs of heads: if merging them into one head does not hurt validation performance beyond a small tolerance, merge them (competence-constrained elegance).
- For forgetting, periodically compute each head's relevance; if relevance is below a threshold and the head has not been used for a long time, retire it.

- For information-seeking, compute uncertainty on each unlabeled input; if uncertainty exceeds a threshold, query the label and update, otherwise skip.

The memory updater U_η implements SGD updates to heads, creation of new heads, merging (e.g., retraining a combined head), and deletion of heads.

4 Experimental Setting

All experiments are conducted on simple synthetic classification tasks designed to isolate specific questions about growth, compression, forgetting and querying.

4.1 Worlds

We use two main task families:

Linear worlds. Each world is a 2D linearly separable problem: a hidden weight vector w and bias b define labels as $y = \mathbb{I}[w^\top x + b > 0]$.

Non-linear worlds. Each world is a curved decision boundary. We instantiate this in two ways:

- Logistic regression on quadratic features $\phi(x) = [x_1, x_2, x_1^2, x_2^2, x_1 x_2, 1]$.
- A small ground-truth MLP (2–8–1) with a tanh hidden layer.

4.2 Training regimes

We consider several regimes:

- **Segmented life**: an episode consists of a fixed sequence of segments, each generated from a single world (e.g., W0, W1, W2, W3, then W1 again, etc.). Some worlds are "alive" (recur later), others are "dead" (appear once).
- **Task selection**: the agent can choose which world to sample on each step, under a fixed total budget of training steps.
- **Partially observable labels**: inputs arrive unlabeled; the agent chooses when to query labels subject to an implicit label budget.

In all cases we measure:

- Per-world test accuracy at the end of training.
- The number of active heads (a proxy for elegance).
- When applicable, the fraction of steps on which labels were queried.

5 Results

We summarize the main experimental findings thematically rather than in chronological order.

5.1 Relevance and forgetting

We first studied *age-aware* relevance in a segmented life with "alive" and "dead" worlds. Heads track the last segment they were updated on; a head that has not been used for a fixed number of segments becomes a deletion candidate.

In linear worlds with no re-emergence, age-aware pruning:

- Removes heads for dead worlds.
- Reduces the number of active heads.
- Preserves accuracy on currently active worlds almost perfectly.

In non-linear worlds with re-emergent tasks, the story changes. When world W2 appears early, disappears for several segments, and then reappears, age-only relevance:

- Often deletes the original head for W2.
- Forces the system to re-learn W2 from scratch when it returns.
- Reduces the total number of heads, but introduces small drops in accuracy across all worlds due to interference and limited capacity.

Reward-weighted relevance. To address this, we introduced a slow relevance feeling combining age and importance. Each world is assigned an importance weight (e.g., W2 is safety-critical and high-importance). Each head accumulates an importance sum as it trains on different worlds. Relevance is defined as

$$R = \frac{\text{importance\_sum}}{\text{age} + 1},$$

and a head is only pruned if R falls below a threshold.

In experiments with a high-importance, rare world W2 and low-importance, dead world W3, reward-weighted relevance:

- Preserves performance on W2 much closer to a no-prune upper bound.
- Still prunes heads associated with low-importance worlds.
- Reduces the overall number of heads by roughly 40% compared to no-prune.

The key lesson is that forgetting should depend not just on recency but on the *importance* of experience. Age-only forgetting is too blunt; reward-weighted relevance is a better candidate "feeling" for deciding what can be safely retired.

5.2 Feature growth and non-linear heads

Inspired by Oak, we allowed the system to grow its representation: confusion can trigger not just new heads, but new features. Concretely, when loss remains high, we test candidate features (e.g., quadratic terms) and promote those that significantly reduce training loss.

This experiment showed that:

- Adding non-linear features when confused improves accuracy on under-represented and tricky worlds.

- It allows the system to maintain or improve performance with fewer heads than a purely linear baseline.
- However, interference remains when a small number of heads must share multiple curved worlds; capacity still matters.

We then moved to tiny neural heads (2–8–1 MLPs) on non-linear worlds generated by similar MLPs. Here, capacity is well-matched to the task family. With an always-query baseline and sufficient training, these heads approach 95–99% accuracy per world.

5.3 Information seeking: uncertainty as a feeling

We next treated uncertainty as an explicit value channel guiding information-seeking actions. In a single-world setting with unlabeled data, the agent can query labels or skip. Uncertainty is computed as $u = \min(p, 1-p)$ for a predicted probability p.

In linear and non-linear toy worlds, we compared:

- Always-query: ask for every label and update.
- Random querying: query with a fixed probability.
- Uncertainty-based querying: query only when u exceeds a threshold.

Across settings we found that:

- Always-query reaches the highest accuracy but uses all labels.
- At matched label budgets, uncertainty-based querying performs at least as well as random querying and often slightly better.
- In the non-linear MLP experiments, uncertainty-based querying attained essentially the same accuracy as always-query while using roughly 18–25% of the labels.

In other words, when heads are expressive enough and the world is learnable, an uncertainty feeling is sufficient to approach "100%" performance with far fewer labels.

5.4 Combining uncertainty and relevance

We then combined uncertainty-driven querying with reward-weighted relevance in a multi-world setting. The agent:

- Receives unlabeled inputs from a mixture of worlds (with different frequencies and importance).
- Uses uncertainty to decide when to query labels.
- Tracks relevance per head and prunes old, low-relevance heads.

Compared to:

- A no-prune, always-query upper bound.
- An age-only prune with uncertainty.

the combined system:

- Uses roughly half the labels of the upper bound.
- Preserves nearly the same importance-weighted performance as always-query.
- Protects the rare, high-importance world better than age-only forgetting, while allowing cheap worlds to degrade modestly.

This provides a concrete demonstration of the EGWM story: fast feelings (uncertainty) drive information-seeking; slow feelings (relevance) decide which knowledge must be kept.

5.5 Planning and committee disagreement

We also explored two additional ideas:

Planning which world to train on. We allowed the agent to choose which world to sample under a fixed training budget, using either:

- random choice,
- importance-weighted choice, or
- a "feelings" planner that scores worlds by importance times estimated error plus an age term.

In simple stationary multitask settings, a static importance-weighted scheduler was already strong, and the feelings planner mainly reallocated effort from cheap to important worlds without a clear advantage in overall score. This suggests that more complex planning will matter most in non-stationary or partially observable environments.

Committee disagreement. We tested a query-by-committee scheme: a small committee of logistic heads with different initializations predicts on each input, and labels are queried only when predictions disagree. In our realizable low-dimensional world, committee members quickly converged, disagreement collapsed, and the scheme either under-queried (and underfit) or queried almost everything (losing label efficiency). In this particular regime, simple single-head uncertainty sufficed.

6 Discussion

6.1 Capacity first, then feelings

A recurring theme is that no amount of clever value signals can compensate for insufficient representational capacity. Linear heads on curved worlds plateau around 60–70% accuracy regardless of how well we manage spawning, merging, and forgetting. Once we move to heads that match the task class (quadratic features or small MLPs), the same value signals suddenly become powerful: they can gate learning, compress structure, and schedule queries without preventing convergence.

This suggests a division of labour:

- *Heads* (world models, options) should be as expressive as needed to represent regularities.
- *Feelings* (value channels) should mostly decide when and how strongly these heads are allowed to change or be created or destroyed.

6.2 Multiple time scales of value

The second theme is the importance of multiple time scales:

- Fast confusion and uncertainty feelings decide whether to trust current predictions and whether to acquire more information.
- Medium-scale competence and elegance feelings decide when it is safe to merge or distill heads.
- Slow relevance feelings decide which heads can be retired once the world and the agent's priorities have changed.

This multi-time-scale structure mirrors biological systems in which fast neuromodulatory signals handle immediate risk and novelty, while slower offline processes consolidate or delete memories.

6.3 EGWM as an emotional Oak

Sutton's Oak architecture describes a tree of features, subproblems, options and models grown from experience. EGWM can be interpreted as a version of Oak in which the meta-data and curation mechanisms are explicitly framed as "feelings." In our toy experiments:

- Confusion and novelty play the role of generating new features and options.
- Competence and elegance guide which options and models are merged or distilled.
- Relevance decides which branches of the tree are pruned when they are no longer useful.
- Uncertainty drives the active selection of which parts of the tree to refine by querying labels or revisiting data.

6.4 Limitations

Our work has several obvious limitations:

- The environments are small, supervised and fully synthetic; there is no long-horizon control or rich sensory input.
- World identification is handled either by oracle routing or by simple head selection; we do not tackle the full problem of latent task inference.
- Value signals are designed by hand; there is no higher-level system learning how to produce or combine feelings optimally.

Despite these limitations, the experiments do illuminate structural questions that are likely to persist at scale: how to grow and prune specialist modules, how to modulate plasticity, and how to allocate limited data collection and memory budget.

7 Conclusion and Future Work

We started from a simple question: if we endow a model with something like "feelings" about confusion, certainty, importance and elegance, can it learn more like a brain and less like a static pattern recognizer?

Across EGWM Parts I–III and the new experiments in this paper, our answer is tentatively yes — with important caveats. In toy worlds where capacity matches the task:

- Fast feelings can prevent learning from garbage phases and trigger the creation of new specialists and features when needed.
- Slower feelings can compress structure without hurting performance, and forget genuinely unimportant worlds.
- Uncertainty alone is enough to drive efficient information-seeking, achieving near-perfect accuracy with significantly fewer labels.
- Reward-weighted relevance can protect rare but safety-critical worlds from being forgotten, even when global compression pressure is high.

Taken together, these results sketch a path toward a more dynamic, self-curating form of intelligence: a world-bank that grows, distills and selectively forgets under the guidance of a small set of multi-time-scale value signals.

Future work includes:

- Scaling these ideas to richer environments (e.g., partially observable control tasks) where planning and option discovery matter more.
- Learning the value channels themselves, rather than hand-designing them.
- Integrating explicit planning over options and worlds into the EGWM loop, bringing it closer to full Oak-style architectures.

Even in their current toy form, however, the EGWM experiments suggest that "feeling the AGI" — giving a system structured meta-signals about its own competence, confusion, importance and elegance — may be a key ingredient in bridging the gap between powerful pattern recognizers and robust, continually learning agents.