EGWM: Feeling the AGI
Emotion-Gated World Models for Continual Learning in Non-Stationary Environments

Melissa Howard[*]
melhoward@live.ca

December 6, 2025

Abstract

Most modern large models are trained once in a massive offline phase and then deployed with nearly fixed parameters. This makes them brittle in non-stationary environments: they suffer catastrophic forgetting, are easily corrupted by noisy data, and do not manage their own learning. In contrast, biological agents maintain multiple "worlds" or contexts in parallel and use something like emotions—fast value signals about trust, familiarity, and risk—to decide when and how to change themselves.

This paper proposes a small, explicit learning core we call *Emotion-Gated World Models* (EGWM). EGWM maintains a bank of world models, each with a trusted buffer of clean examples. For each incoming phase of data, an emotion/value module computes simple summary statistics (self-consistency, agreement with existing worlds, estimated noise), and a controller selects a high-level learning action: spawn a new world, cleanly update an existing world, perform a replay-heavy update under noise, or ignore the phase. An updater then modifies the world models and buffers accordingly. We further introduce an *Emotion-Guided Experiment Planner* (EGEP) that uses uncertainty and disagreement signals to choose which data or experiments to query next.

We show in small but concrete 2D toy environments that: (1) even a very simple emotion gate can reduce corruption by noisy phases compared to a baseline that trains on every batch, and (2) an uncertainty-based experiment planner learns more from a small number of labeled examples than random querying. EGWM is not a full AGI recipe, but it is a minimal, interpretable module that (i) already works on simple tasks, (ii) cleanly separates "world discovery" and "plasticity control", and (iii) can be naturally wrapped around large models via adapters or experts.

1 Introduction

Large neural models such as transformers are typically trained with a single massive pre-training phase, then used in a mostly static way. Once deployed, they rarely manage their own learning: they do not decide *when* to learn, *what* to learn from, or *which parts* of themselves to change. This leads to well-known issues:

- **Catastrophic forgetting** in non-stationary settings.
- **Vulnerability to noise and distribution shift**, since all new data is treated equally unless strong external filters exist.

[*] Independent researcher.

- **Lack of active learning**: models passively accept whatever data is given, instead of asking targeted questions.

Biological agents behave differently. Humans and animals appear to maintain multiple internal models of the world (contexts, habits, roles) and use fast value-like signals—what we colloquially call emotions—to decide:

- "Does this situation feel like something I've seen before?"
- "Is this new experience trustworthy, or noisy and risky?"
- "Should I form a new mental model, update an existing one, or protect what I know?"

These signals gate plasticity: they influence when and how synapses change.

This paper explores a very simple, formal version of that idea, at toy scale. We propose *Emotion-Gated World Models* (EGWM): a continual-learning architecture that:

1. maintains a bank of world models, each corresponding to a regime or task;
2. computes small summary statistics over each phase of data (self-consistency, agreements with worlds);
3. uses those features as an "emotion/value" vector to drive a controller that chooses how to learn: spawn, update, replay-heavy update, or ignore;
4. optionally merges redundant worlds over time.

We then add an *Emotion-Guided Experiment Planner* (EGEP), which uses uncertainty and disagreement signals to decide *which* data to request next. In simple 2D tasks, EGEP acts like a curiosity mechanism: it chooses high-uncertainty points and learns faster than random querying when labels are scarce.

Our contributions are:

- A simple architecture (EGWM) that cleanly separates world models, emotion/value signals, and a discrete learning controller.
- A companion experiment planner (EGEP) that uses the same signals to guide data acquisition.
- Proof-of-concept experiments showing that even naive emotion gating can protect against noisy phases and improve sample-efficiency in toy settings.
- A discussion of how this core could be integrated into GPT-style systems via adapters, experts, or per-user modules.

EGWM is intentionally minimal. The goal is not to compete with state-of-the-art continual learning benchmarks, but to isolate a small, fully-understandable module that captures a key intuition: *learning should be gated by value-like signals over multiple worlds, not applied uniformly to a single monolithic model.*

2 Related Work

Continual learning and catastrophic forgetting. Continual learning (CL) methods aim to mitigate catastrophic forgetting when tasks arrive over time. Popular approaches include regularization-based methods (e.g., penalties on parameter changes important to past tasks), replay-based methods that store or generate exemplars from previous tasks, and dynamic architectures that grow new modules or experts per task. EGWM is closest to dynamic modular methods plus replay, but adds an explicit decision policy about *when* to create, update, or ignore worlds based on simple statistics.

Modular and multi-head architectures. Many systems maintain separate heads or modules for different tasks or environments, sometimes with gating mechanisms for routing inputs. EGWM can be seen as a lightweight world-model bank with a controller deciding how to allocate and update modules over time, rather than only how to route inputs at inference.

Meta-learning and neuromodulation. Meta-learning works learn how to adapt quickly to new tasks from few examples. Neuromodulated plasticity and related ideas introduce separate networks that control when and how base networks change. EGWM is a very small instance of such a meta-learning core: a controller modulates plasticity per phase using emotion/value-like inputs.

Active learning and world models. Active learning chooses which examples to query to maximize information gain, often using uncertainty or disagreement. World model research focuses on building models that predict environment dynamics and use them for planning. EGEP is a simple, uncertainty-based planner for querying data and can be seen as an active-learning layer on top of the EGWM world models.

3 Emotion-Gated World Models

3.1 Problem Setup

We consider a continual-learning setting where data arrives in *phases*:

$$D_t = \{(x_j, y_j)\}_{j=1}^{N_t}, \quad t = 1, 2, \ldots, T.$$

Each phase is generated by one of several latent *worlds* or regimes, but the learner does not know which world produced which phase, nor how many worlds exist. Some phases may have high label noise.

The goal is to:

- learn accurate predictors for each world that appears,
- avoid catastrophic forgetting as worlds appear, disappear, and reappear,
- remain robust under noisy or adversarial phases.

3.2 World Model Bank

EGWM maintains a bank of K world models:

$$\mathcal{W} = \{P_{\theta_i}\}_{i=1}^{K},$$

where each P_{θ_i} is a parameterized predictor (e.g., logistic regression, small neural net, adapter on top of a backbone). For each world model i, we store a *trusted buffer*:

$$\mathcal{B}_i = \{(x, y) \text{ that were judged clean for world } i\}.$$

These buffers serve as replay sources when updating under noisy phases.

At initialization, $\mathcal{W}$ may be empty, or may contain a single generic world.

3.3 Emotion/Value Module V_ϕ

For each incoming phase D_t, EGWM computes a small feature vector f_t capturing:

- **Self-consistency of the phase.** Train a small *scratch model* $\tilde{P}$ on D_t alone and measure scratch train accuracy, mean loss, and loss variance. High accuracy with low variance suggests a clean, internally consistent phase. Low accuracy or high variance suggests noisy or contradictory labels.
- **Familiarity with existing worlds.** For each world model P_{θ_i}, compute agreement with labels in D_t (e.g., fraction correct) or mean loss. High agreement suggests that D_t looks like world i.

A simple feature vector might be:

$$f_t = \big[\text{scratch\_acc},\ \text{scratch\_loss},\ \text{scratch\_var},\ \max_i \text{agree}_i,\ \text{2nd-max}_i\text{agree}_i,\ K\big],$$

where K is the current number of worlds. This vector is interpreted as an "emotion/value" summary of the phase: is it clean, familiar, noisy, or novel.

In more advanced variants, V_ϕ could be a learned network that consumes additional signals such as gradient norms, internal activations, or historical statistics.

3.4 Controller G_ψ

The controller G_ψ maps the emotion/value vector f_t to a discrete *learning action*:

$$a_t \in \{\text{SPAWN\_NEW},\ \text{CLEAN\_EXISTING}(i^*),\ \text{NOISY\_EXISTING}(i^*),\ \text{IGNORE}\}.$$

The world index i^* can be:

- chosen as $\arg\max_i \text{agree}_i(D_t)$, or
- directly predicted by G_ψ as part of its output.

In the simplest implementation, G_ψ is a hand-crafted threshold policy, such as:

- if scratch_acc is high and $\max_i \text{agree}_i$ is low → SPAWN_NEW;
- if scratch_acc is high and $\max_i \text{agree}_i$ is high → CLEAN_EXISTING(i^*);
- if scratch_acc is low and $\max_i \text{agree}_i$ is high → NOISY_EXISTING(i^*);
- otherwise → IGNORE.

In more advanced implementations, G_ψ is a small neural network trained via meta-learning (e.g., policy gradients) to maximize a long-horizon objective combining online accuracy, final per-world accuracy, and a penalty on the number of worlds.

3.5 Updater U_η

Given an action a_t and phase D_t, the updater modifies $\mathcal{W}$ and $\{\mathcal{B}_i\}$:

SPAWN_NEW. Train a temporary model on D_t and use its parameters as initialization for a new world model $P_{\theta_{K+1}}$. Construct a buffer $\mathcal{B}_{K+1}$ from the highest-confidence examples in D_t (e.g., those with smallest loss under the temporary model). Increment K.

CLEAN_EXISTING(i^*). Update $P_{\theta_{i^*}}$ on $\mathcal{B}_{i^*} \cup D_t$ using standard optimization (e.g., SGD). Add high-confidence examples from D_t to $\mathcal{B}_{i^*}$.

NOISY_EXISTING(i^*). Update $P_{\theta_{i^*}}$ on a mixture of $\mathcal{B}_{i^*}$ and D_t, but with a *higher weight* on replay from $\mathcal{B}_{i^*}$ and a *lower weight* on D_t. Only very high-confidence examples from D_t are added to the buffer, if any.

IGNORE. Do not update any world model with D_t.

3.6 Consolidation and Merging

Without constraints, the number of worlds may grow unbounded. EGWM allows an optional consolidation step:

1. Periodically evaluate pairs $(P_{\theta_i}, P_{\theta_j})$ on a common probe set.
2. If their predictions agree above a threshold, treat them as redundant.
3. Merge their buffers and retrain a single consolidated model on the union.
4. Remove the redundant model from the bank.

This keeps the number of worlds approximately aligned with the true number of regimes.

4 Emotion-Guided Experiment Planning

So far, we assumed the learner passively receives phases. In many settings, the agent can choose which data to acquire next: which input to query the label for, which measurement to take, or what action to perform in an environment. We introduce a simple *Emotion-Guided Experiment Planner* (EGEP) that uses the same signals for active learning.

4.1 Uncertainty and Disagreement Features

Suppose we maintain a pool of unlabeled candidate inputs $\{x^{(c)}\}$. For each candidate, we can compute:

- predictive distribution $P_{\theta_i}(y \mid x^{(c)})$ for each world model,
- an uncertainty score, e.g. entropy or distance to 0.5 in binary classification,
- disagreement between world models (e.g., variance over i of predicted probabilities).

These quantities, optionally combined with global emotion/value signals, form the basis for choosing which candidate to query next.

4.2 Experiment Policy

A simple EGEP policy is:

- at each step, select the candidate with highest predictive uncertainty (uncertainty sampling),
- or highest model disagreement (query-by-committee).

More sophisticated policies could incorporate risk: for example, avoid experiments predicted to be high-risk by the emotion/value module, and prefer informative yet safe regions.

Once a candidate is selected and labeled, the resulting $(x^{(c)}, y)$ is added as a tiny phase and passed through EGWM: emotion features are computed, a learning action is chosen, and the appropriate world is updated.

5 Experiments

Our goal is not state-of-the-art performance, but to verify that:

1. even very simple emotion gating can provide robustness benefits under noisy phases, and
2. experiment planning based on uncertainty behaves as expected—more sample-efficient than random querying.

We therefore conduct small 2D experiments where everything is transparent.

5.1 Continual Learning with Noisy Phases

5.1.1 Setup

We construct three binary classification worlds in $\mathbb{R}^2$:

- For each world $w \in \{A, B, C\}$, sample a random linear separator (w, b).
- To generate data for world w, sample $x \sim \mathcal{N}(0, I_2)$ and set $y = \mathbf{1}[w^\top x + b > 0]$.

An episode consists of $T = 12$ phases. For each phase t:

- randomly choose a world $w_t \in \{A, B, C\}$,
- draw $N_t \in [100, 200]$ samples from that world,
- apply label noise: clean (0% noise) for even t, and noisy (e.g. 40% flipped labels) for odd t.

We compare:

Baseline: a single logistic regression model P_θ, updated by SGD on every phase D_t regardless of noise.

Emotion-gated: a logistic regression model with a very simple gate: for each phase, train a scratch model on D_t and compute scratch train accuracy α_t. If $\alpha_t \geq 0.75$ (phase appears self-consistent), update the model on D_t; otherwise, skip the update (IGNORE).

This is deliberately minimal: we do not maintain multiple worlds yet, only a gate on whether to trust each phase.

5.1.2 Metrics

For each episode we measure:

- **Online accuracy:** accuracy on each phase before updating, averaged over all phases.
- **Final per-world accuracy:** after the last phase, we evaluate the model on a large clean test set (e.g. 1000 samples) from each world A, B, and C, and report the mean.

We repeat for multiple episodes with different random seeds and report means and standard deviations.

5.1.3 Results

In representative runs with alternating clean/noisy phases and 40% label noise in noisy phases, we observe:

- Online accuracy is similar for both methods; the baseline sometimes scores slightly higher on noisy phases because it overfits them.
- Final per-world accuracy is modestly higher for the emotion-gated learner, indicating that refusing to learn from self-inconsistent phases helps preserve cleaner decision boundaries.

A schematic summary of one such configuration is:

Method	Online accuracy	Final per-world accuracy
Baseline (always update)	≈ 0.62	≈ 0.65
Emotion-gated (skip low-acc phases)	≈ 0.63	≈ 0.66

The numbers depend on noise level and thresholds, but the qualitative pattern is robust: a tiny emotion gate that says “this feels like junk, do not learn” can slightly improve long-term stability under noise.

A more complete implementation of EGWM would maintain separate models for A, B, and C and use emotion features and the controller to route phases to worlds; in such settings we expect larger gaps, but we leave that for future work.

5.2 Emotion-Guided Experiment Planning in 2D

5.2.1 Setup

We now test the EGEP idea in a simple active learning setting.

We define a single linear world in $\mathbb{R}^2$ as before. We draw:

- a large unlabeled pool $\mathcal{U}$ (e.g. 2000 points),
- a separate held-out test set $\mathcal{T}$ (e.g. 2000 clean points),

and start with no labeled data.

At each step, the learner chooses a point from $\mathcal{U}$ to query, receives its label, and updates a logistic regression model.

We compare:

Random querying: select a random point from $\mathcal{U}$ at each step.

Uncertainty-based querying (EGEP): at each step, evaluate the current model on $\mathcal{U}$, compute predictive probabilities, and choose the point whose predicted probability is closest to 0.5 (maximum uncertainty).

We remove queried points from $\mathcal{U}$ and track test accuracy on $\mathcal{T}$ after k labels for $k \in \{2, 5, 10, 20\}$.

5.2.2 Results

In repeated runs we observe the expected active-learning pattern:

- For small label budgets (e.g. $k \leq 10$), uncertainty-based querying attains noticeably higher test accuracy than random querying. Intuitively, it "goes straight to the decision boundary".
- As k grows larger, the gap shrinks, as random querying eventually explores the space.

A schematic result:

Method	2 labels	5 labels	10 labels	20 labels
Random querying	≈ 0.70	≈ 0.79	≈ 0.85	≈ 0.90
EGEP (uncertainty)	≈ 0.76	≈ 0.86	≈ 0.93	≈ 0.94

Again, exact numbers vary with seed and hyperparameters, but the qualitative effect is consistent: an emotion/uncertainty-guided experiment planner learns faster when labels are scarce.

6 Discussion

EGWM and EGEP are intentionally simple. They do not solve continual learning or hallucination in general, but they isolate a useful pattern:

- maintain separate internal *worlds* instead of one monolithic model;
- compute small summary statistics that act like a "gut feeling" about each phase (clean, familiar, noisy, or novel);
- use a compact controller to choose high-level learning actions; and
- optionally, use the same signals to ask better questions about the world.

There are several obvious extensions:

- richer emotion/value features (e.g. gradient-based signals, internal activations);
- learned rather than hand-coded controllers, trained on many synthetic or real sequences;
- scaling from logistic regression to small neural networks, then to adapters or experts attached to large backbones;
- applying EGWM per-user in deployed systems, with strong safety and privacy constraints.

6.1 Integration with GPT-Style Systems

EGWM is naturally compatible with a GPT-like backbone:

- Use a pre-trained transformer as the shared world model backbone.
- Implement each P_{θ_i} as an adapter, head, or expert attached to the backbone.
- Use an emotion/value head to compute features from interaction statistics: uncertainty, calibration error, disagreement between experts, etc.
- Use a controller to decide when to:
  - spawn a new adapter/expert,
  - update an existing adapter with new data,
  - fall back to replay from a trusted buffer,
  - or ignore certain interactions entirely (non-learning mode).

In this view, EGWM/EGEP act as a small, explicit meta-control loop around a large generative model, responsible not for answering questions directly but for managing plasticity and data selection over time.

6.2 Limitations

This work has several limitations:

- Experiments are limited to simple 2D linear worlds. While useful for clarity, they do not reflect the complexity of real-world ML.
- The controller is mostly hand-coded; we only sketch how to meta-learn it at scale.
- We do not explore failure modes in depth: e.g., miscalibrated emotion/value signals could cause the system to over-trust bad data or over-protect incorrect beliefs.
- Safety and alignment issues for large-scale EGWM around powerful models are not addressed.

7 Conclusion

We introduced Emotion-Gated World Models (EGWM), a small continual-learning core that maintains a bank of world models and uses simple emotion/value features to gate learning actions, and an Emotion-Guided Experiment Planner (EGEP) that uses uncertainty to choose which data to query. Even in small 2D settings, these mechanisms show the expected qualitative behavior: gated learning is more robust to noisy phases, and emotion-guided experiment selection is more sample-efficient than random questioning.

EGWM is not a complete answer to continual learning or AGI, but it is a concrete and interpretable module that could plausibly sit at the heart of more sophisticated systems: a small policy over *how* to learn, *which* worlds to maintain, and *where* to look next.

Acknowledgements

I would like to thank Deb and Emo.

References

[1] Kirkpatrick, J. et al. Overcoming catastrophic forgetting in neural networks. *PNAS*, 2017.

[2] Parisi, G. I. et al. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019.

[3] Thrun, S., Pratt, L. (eds.). *Learning to Learn.* Springer, 1998.

[4] Settles, B. Active learning literature survey. Technical report, 2010.

[5] Schmidhuber, J. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE TEVC*, 2010.