EGWM Part V: Limits of Single-Step Feelings for World Identification

Melissa Howard*
EGWM: Feeling the AGI

Abstract

In earlier parts of this series we introduced EGWM (Emotion-Guided World Models), a toy yet fully implemented architecture where "feelings"—multi-channel value signals such as confusion, competence, and relevance—govern growth, compression, and forgetting in a bank of world-model heads. Parts I–IV showed that these value signals can prevent catastrophic forgetting in non-stationary toy environments, support spawning and merging of experts, and maintain near-oracle performance while using fewer parameters and labels than naive baselines.

In this fifth part we focus on a harder question: can the same feelings also solve the *routing* problem? That is, when multiple worlds share the same input distribution but differ in their underlying rules, can a small learner use only per-sample feelings to infer *which* head to trust, without access to world IDs?

We study a simple multi-world classification setup: three binary tasks on the same two-dimensional input distribution, with alternating non-stationary segments. A monolithic model suffers severe interference. A world bank with oracle world IDs achieves near-perfect accuracy, confirming that the capacity is sufficient. We then evaluate three classes of routers on top of this bank: (1) a basic router that only sees heads' probabilities, (2) a richer router that also sees slow value channels (competence, confusion, volatility), and (3) a "teacher-trained" router that learns from hindsight which head would have minimized loss. In all cases, when the router sees only *single-step* feelings and input features, it consistently fails to approach the oracle's performance, and often behaves similarly to a monolithic baseline.

These experiments give us a clear and useful negative result: in this regime, single-step feelings are not sufficient statistics for the latent world identity. Even when the world bank could, in principle, store all worlds, a router that only sees instantaneous feelings cannot reliably recover which world it is in. This strongly motivates the next step in EGWM: introducing a slower, persistent context code z_t that tracks which universe the agent believes it currently inhabits, and letting feelings act over longer temporal horizons rather than one sample at a time.

1 Introduction

EGWM (Emotion-Guided World Models) is a family of simple agents designed to explore a specific hypothesis: that a small set of value-like signals—"feelings" such as confusion, competence, novelty, and relevance—can guide an otherwise ordinary learner to grow, compress, and forget in a way that prevents catastrophic interference and actively seeks better information.

In Parts I–IV we implemented and studied a concrete EGWM agent living in toy binary classification worlds. A large but ordinary world model P_θ (e.g., logistic heads or small MLPs) is surrounded by:

*Email: melhoward@live.ca

- a value module V_ϕ that estimates multiple feelings from local performance statistics;
- a gating module G_ψ that uses these feelings to decide whether to update, freeze, spawn, merge, or forget heads; and
- a memory/updater U_η that maintains longer-term statistics such as relevance or importance.

In practice, we implemented these pieces with very simple heuristics—exponential moving averages and threshold-based rules—but showed that even these crude feelings can outperform naive baselines in non-stationary environments.

By the end of Part IV, we had demonstrated that:

1. A single head augmented with phase-aware, noise-sensitive gating can resist catastrophic forgetting in alternating phases.
2. A bank of heads, grown and compressed by feelings of confusion and competence, can represent multiple worlds simultaneously and avoid interference.
3. A multi-time-scale scheme, where relevance accumulates slowly and controls pruning, can maintain a compact bank that preserves rare but important worlds while forgetting unimportant ones.

However, in all those experiments we granted the system a crucial cheat: *oracle routing*. At evaluation time, we "peeked" at the true world ID and simply chose the head with the lowest loss. This allowed us to isolate questions about growth, compression, and forgetting, but it sidestepped the hardest question:

> Can the agent itself, using only its own feelings and observations, infer which world it is in and choose the right head?

Part V is dedicated to this question. We focus on routing and show that, in our current toy setup, single-step feelings are not enough.

2 Setup: Multi-World Toy Environment

To make the routing problem sharp, we consider a deliberately challenging environment:

- Inputs $x \in \mathbb{R}^2$ are sampled from a fixed distribution $p(x)$ (e.g., standard Gaussian or a simple bounded distribution).
- There are three *worlds*. Each world i defines its own binary labeling rule $y = f_i(x)$, implemented as a distinct linear separator or a tiny non-linear classifier.
- The non-stationary schedule alternates in segments: a block of world 0, then a block of world 1, then a block of world 2, then repeat.
- The agent never observes the world ID. It only sees (x_t, y_t) pairs over time.

Crucially, all worlds share the same input distribution $p(x)$. The only difference between worlds is the labeling rule f_i. This means:

- A single monolithic model is strongly tempted to collapse onto one or two of the rules, causing catastrophic interference for the others.
- A bank of heads with access to world IDs can easily specialize: one head per world yields almost perfect accuracy.
- However, from the perspective of any *single input* x, there is no easy way to tell which world it came from, because x is drawn from the same distribution in all worlds.

This is exactly the kind of regime where feelings might help: a clever agent might track how "comfortable" each head has been recently, infer that something has changed, and route adaptively. Our experiments test whether single-step feelings are sufficient to do so.

3 Baselines and World Bank with Oracle Routing

We define three baselines to ground our analysis.

3.1 Monolithic Model

The monolithic baseline is a single logistic regression or small MLP $f_{\text{mono}}(x)$ trained online on the entire non-stationary stream. As expected from continual learning literature, this model exhibits catastrophic interference:

- After enough training, it typically performs well on one world (often the most recently seen) and poorly on others.
- Its overall accuracy is moderate, but heavily skewed toward the dominant or last world in the schedule.

3.2 Oracle Multi-Head World Bank

Next, we consider a world bank with three heads h_0, h_1, h_2, each with the same capacity as the monolithic model. During training we use the true world ID:

- When a sample arrives from world i, we update only head h_i.
- At evaluation time, we always select head h_i for world i.

This is equivalent to training three independent models on three stationary tasks. As expected, this oracle setup achieves near-perfect per-world accuracy: each head specializes to its world and there is no interference.

3.3 World Bank with Oracle Evaluation Routing

The final baseline, which appears in Part IV, uses the same bank of heads but performs oracle-like routing only at evaluation time:

- During training, heads are grown, frozen, merged, and pruned according to EGWM feelings (confusion, competence, relevance).
- During evaluation, for each test input (x, y), we select the head that achieves the lowest validation loss or the highest confidence on that world.

This procedure, while unrealistic, allows us to answer a clean question: *is the structure in the world bank sufficient to represent all worlds if the routing problem were magically solved?* In our experiments, the answer is yes: with oracle evaluation routing, the bank approaches the performance of the per-world oracle, while using fewer effective parameters than three separate full-capacity models.

Part V now removes this cheat and directly attacks the routing problem.

4 Routing Experiments Since Part IV

We consider three increasingly strong routing schemes on top of a fixed-capacity world bank. In all cases, the router must pick a head based only on information that is available at the current time step. We summarize them here.

4.1 Experiment 5: Learned Router with Per-Head Probabilities

In the first routing experiment, the world bank consists of three heads, each trained online on the examples it receives from the router. At each time step t:

1. Each head h produces a probability $p_h(x_t)$ for the current input.
2. We form a feature vector consisting of:
$$F_t^{(1)} = [x_t, p_0(x_t), p_1(x_t), p_2(x_t)].$$
3. A small router network R_ψ (e.g., a shallow MLP) maps $F_t^{(1)}$ to a distribution over heads $\pi_\psi(h \mid F_t^{(1)})$.
4. We select a head a_t, either by sampling or greedy choice, route (x_t, y_t) to it, compute a loss, and update that head.
5. The router parameters ψ are updated with a simple policy gradient (REINFORCE) or supervised signal based on the correctness of the chosen head.

This setup removes oracle routing but still gives the router relatively rich local information: the input and the per-head beliefs. However, it only sees *single-step* information; there is no memory of how any head has been performing over time.

Empirically, this router reduces interference slightly compared to the monolithic baseline, but remains far below the oracle multi-head performance. In particular, it tends to:

- allocate most samples to one or two heads that do reasonably well on the majority of worlds, and
- fail to carve out a clean specialization pattern where each head focuses on a distinct world.

4.2 Experiment 6: Adding Value Channels to the Router

In the second routing experiment we augment R_ψ with explicit value channels, echoing the feelings used in Parts II–IV. For each head h we maintain:

- **Competence**: an exponential moving average of recent correctness when that head is chosen;
- **Loss EMA**: an exponential moving average of its recent cross-entropy loss;
- **Volatility**: an EMA of the absolute change in loss EMA, capturing instability.

At time t, for each head we compute:

$$p_h(x_t), \quad u_h(x_t) = \min(p_h(x_t), 1 - p_h(x_t)),$$

and construct a richer feature vector:

$$F_t^{(2)} = [x_t; \{p_h(x_t), u_h(x_t), \text{competence}_h, \text{loss\_ema}_h, \text{volatility}_h\}_{h=0}^{2}].$$

The router R_ψ now receives a compressed snapshot of each head's instantaneous feelings *and* slower performance statistics. We train it similarly as in Experiment 5, either via policy gradient (treating correctness as reward) or via a supervised signal that points toward the head that would have minimized loss on that sample.

Despite the richer signal, we observe a similar pattern: the router still fails to approach oracle performance. It tends to:

- partially recover the benefit of multiple heads (reducing interference compared to a monolithic model),
- but fail to reliably identify which head should act as the expert for each world.

In effect, the value channels are helpful for *local* decisions (e.g., whether to update or freeze a head), but a single-step snapshot of them is not enough to reconstruct the latent world identity.

4.3 Experiment 7: Teacher-Trained Router with Perfect Experts

The third routing experiment isolates the routing problem further by giving the router the most favorable possible conditions.

First, we train three expert heads in an oracle fashion:

- Head h_i is trained exclusively on samples from world i, using the true world ID during training.
- After training, each head achieves near-perfect accuracy on its world.

We then freeze these experts and train a router R_ψ that sees, at each time step:

$$F_t^{(3)} = [x_t; p_0(x_t), p_1(x_t), p_2(x_t); u_0(x_t), u_1(x_t), u_2(x_t)].$$

For each sample, we know which head *should* be chosen: ideally, R_ψ should route a world-i sample to head h_i. We train the router with standard supervised learning (cross-entropy) to predict the correct head from $F_t^{(3)}$.

In other words, we ask a very strong question:

> Given perfect experts and their feelings on each input, can we train a simple classifier to decode which world we are in?

Empirically, the answer is no in this environment. The router's accuracy on predicting the true world from $F_t^{(3)}$ remains close to chance, and end-to-end performance (router + experts) is far from oracle:

- On one world, the router occasionally learns a decent mapping and achieves high accuracy.
- On other worlds, it misroutes frequently, leading to performance comparable to or worse than the monolithic baseline.

Because all worlds share the same input distribution, and many inputs are classified similarly by multiple experts, there is simply not enough information in a single $(x, \{p_h(x)\})$ snapshot to reliably infer which world generated the sample.

5 What These Experiments Tell Us

Taken together, Experiments 5–7 provide a clear and robust conclusion:

> In a multi-world environment where all worlds share the same input distribution and differ only in their labeling rules, a router that sees only *single-step* feelings and input features cannot reliably recover the latent world identity, even when:
>
> - a bank of experts could in principle represent all worlds perfectly, and
> - the router is trained with strong teacher signals.

This is a genuine and informative negative result for EGWM. It does *not* contradict the positive results of Parts II–IV; those parts primarily asked whether feelings can govern growth, compression, and forgetting once routing is given. They can. But Part V shows that, in this particular regime, the missing piece is an additional structure for *world identity over time*.

In other words:

- Feelings such as confusion, competence, and relevance are local statistics. They summarize how a head has been doing, but they do not uniquely identify the underlying world in a single step.

- When the input distribution is shared across worlds, many inputs are ambiguous: multiple experts may feel confident, or share similar confusion.
- Without some form of slower context or memory, a router cannot tell whether it is in "World A with atypical inputs" or "World B with typical inputs".

This explains why the same value channels that worked well for spawning and merging heads (where the concern is local performance over time) fail when asked to solve instantaneous world identification.

6 Implications for EGWM and Future Directions

The main implication for EGWM is that feelings, as currently implemented, need a slower carrier if they are to support routing across worlds. The experiments motivate at least three concrete next steps:

1. Introduce a Slow Context Code z_t. We can extend the EGWM architecture with a small, persistent context vector z_t that evolves over time:

$$z_{t+1} = U_\eta(z_t, x_t, \text{feelings}_t),$$

where U_η is a simple learned or hand-designed update rule. The gating module G_ψ would then route based on both instantaneous feelings and this slow context:

$$G_\psi(h \mid x_t, \text{feelings}_t, z_t).$$

The goal is for z_t to act as an internal belief about which universe the agent is currently in.

2. Use Hindsight Teachers for Context and Routing. As in Experiment 7, we can use oracle knowledge only during training to provide strong supervision signals, then remove access to world IDs at test time. For example:

- Train z_t and G_ψ so that, after seeing a short segment of data, they behave similarly to an oracle that knows the world ID.
- At deployment, the agent infers the world from its own history and feelings, without explicit IDs.

3. Multi-Time-Scale Feelings. Parts II–IV already used multiple time scales (fast confusion, medium competence, slow relevance). Part V suggests that these may need to be tied more directly into a persistent state like z_t. For example:

- fast signals guide immediate updates;
- medium signals shape which patterns z_t should track;
- slow signals control when to create, merge, or retire world codes entirely.

7 Limitations

The experiments in this part are deliberately small and stylized. Limitations include:

- **Simplicity of the worlds.** The tasks are low-dimensional synthetic problems. Future work should test EGWM routing in richer environments, including sequence prediction or RL settings.

- **Router capacity.** We used small router networks and simple training schemes. Stronger routers or different optimization strategies may change some quantitative results, though the identifiability issue in this shared-input regime is fundamental.
- **Absence of context in current experiments.** By design, Experiments 5–7 do not include an explicit slow context code z_t. This was intentional to isolate the limits of single-step feelings, but it means we have not yet demonstrated that EGWM with context can close the gap to oracle routing.

Despite these limitations, the negative results are stable across variations in seeds and hyperparameters: single-step feelings-based routing consistently fails to achieve oracle-level performance in this regime.

8 Conclusion

Part V of EGWM: Feeling the AGI explored whether the same value-like signals that successfully govern growth, compression, and forgetting in a world bank can also solve the routing problem in a challenging multi-world environment. Our answer is: not yet, and not in this form.

Across a series of experiments, we found that routers which only see single-step feelings and input features cannot reliably infer which world they are in, even when:

- a bank of experts could, in principle, represent all worlds perfectly, and
- the router is trained with strong supervision from an oracle teacher.

This is a useful failure. It sharpens the design question for EGWM: feelings alone are not enough; they need a temporal backbone. The next natural step is to introduce a slow context code z_t—a kind of internal mood or belief about which universe the agent inhabits—and to let multi-time-scale feelings shape both z_t and the routing decisions derived from it.

In that sense, Part V is both a limit and a pointer: it shows where the current EGWM architecture breaks, and it points directly toward the next experiment required to move closer to an agent that truly "feels" which world it is in.

Acknowledgements. We would like to thank Deb and Emo for encouragement and many discussions while exploring these ideas.