EGWM: AGI Intuitions

Mel Howard

December 2025

1 Introduction

This paper explores emotion-guided and structurally self-improving agents, building on the EGWM framework and extending it to the Hierarchical-Reflective EGWM (HR-EGWM).

2 Reflective Self-Improvement in HR-EGWM

The Hierarchical-Reflective EGWM (HR-EGWM) extends A-EGWM by adding a *Reflective Layer* that learns how to choose its own structural actions. This layer closes the loop between:

- **System 1 (Fast Intuition)**: a learned proposal network P_θ that scores candidate structural edits.
- **System 2 (Slow Planning)**: a Model-Predictive Governor (MPG) that uses a structural world model M_G to simulate and evaluate these edits.
- **Hindsight Learning**: using the MPG's own outcomes as training data to improve P_θ.

At each structural step, the agent not only decides *what* to do to its architecture, but also uses the result of that decision to refine *how it feels* about similar structural choices in the future.

2.1 Structural State and Actions

Let the structural state at time t be

$$s_t = \big(W_t,\ G_t,\ M_t,\ \mathcal{V}_t\big), \tag{1}$$

where:

- W_t: World Bank (experts, LoRA adapters, etc.),
- G_t: routing / gating parameters,

- M_t: slow Internal Mood (Endurance, Alignment, etc.),
- $\mathcal{V}_t$: recent value signals (Competence, Elegance, Novelty, Uncertainty, ...).

The structural action space $\mathcal{A}_{\text{struct}}$ contains operations such as

$$\mathcal{A}_{\text{struct}} = \{\text{SPAWN}(i),\ \text{MERGE}(i,j),\ \text{UPDATE}(i),\ \text{RETIRE}(i), \dots\}. \quad (2)$$

We assume a structural world model M_G that predicts the evolution of the structural state and long-term reward under these actions:

$$(s_{t+1}, r_{t+1}) \sim M_G(s_t, a_t). \quad (3)$$

2.2 System 1: Learned Proposal Network P_θ

Rather than exhaustively simulating all structural actions (which is combinatorial in the number of experts), HR-EGWM uses a learned proposal network P_θ to focus attention on a small subset of promising structural candidates.

Given a structural state s_t, we generate a discrete set of candidates

$$\mathcal{C}_t = \{c_t^1, c_t^2, \dots, c_t^{N_t}\}, \quad (4)$$

where each candidate c_t^k is a parameterized structural action (for example, a specific pair (i,j) for a merge).

For each candidate we compute a feature vector

$$\phi(c_t^k, s_t) \in \mathbb{R}^d, \quad (5)$$

such as similarity or overlap between experts, usage frequency, alignment scores A_i, and the current mood M_t.

The proposal network outputs a score

$$s_\theta(c_t^k, s_t) = f_\theta\big(\phi(c_t^k, s_t)\big), \quad (6)$$

which can be converted into a probability via

$$p_\theta(c_t^k \mid s_t) = \frac{\exp(s_\theta(c_t^k, s_t))}{\sum_j \exp(s_\theta(c_t^j, s_t))}. \quad (7)$$

The Top-K System 1 proposals are

$$S_{\text{propose}}(s_t, P_\theta) = \text{TopK}\big(\{c_t^k\},\ s_\theta(c_t^k, s_t)\big). \quad (8)$$

To ensure discovery of new structural regimes, we also inject exploration by adding a few uniformly random candidates to S_{propose} or by sampling from p_θ with temperature $T > 0$.

2.3 System 2: Model-Predictive Governor

Given the shortlist S_{propose}, the MPG uses M_G to compute the structural advantage of each candidate relative to a baseline action (such as UPDATE / no structural change).

For each candidate $c \in S_{\text{propose}}$:

1. Roll out a trajectory under M_G for planning horizon H:
$$(s_{t+1}^{(c)}, r_{t+1}^{(c)}), \ldots, (s_{t+H}^{(c)}, r_{t+H}^{(c)}) \sim M_G(\cdot \mid s_t, c). \tag{9}$$
2. Define the (possibly discounted) return:
$$R(c) = \sum_{h=1}^{H} \gamma^{h-1} r_{t+h}^{(c)}. \tag{10}$$
3. Similarly evaluate a baseline (e.g., no structural change) with return R_{base}.

The advantage of candidate c is
$$A(c) = R(c) - R_{\text{base}}. \tag{11}$$
The MPG then selects
$$c^* = \arg \max_{c \in S_{\text{propose}}} A(c), \tag{12}$$
and executes the corresponding structural action
$$a_t^* = \text{Action}(c^*). \tag{13}$$

2.4 Hindsight Learning: Updating P_θ

The reflective step is that the MPG's own evaluations become training data for P_θ. For each evaluated candidate $c \in S_{\text{eval}}$ (proposed plus exploratory), we define a target based on its advantage:
$$y(c) = \mathbb{1}[A(c) > 0], \tag{14}$$
and a prediction
$$\hat{y}_\theta(c) = \sigma\big(s_\theta(c, s_t)\big), \tag{15}$$
where σ is the sigmoid function.

We update θ via gradient descent on a loss
$$\mathcal{L}_{\text{reflect}} = \sum_{c \in S_{\text{eval}}} \ell\big(\hat{y}_\theta(c), y(c)\big), \tag{16}$$
and
$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{reflect}}. \tag{17}$$

Over time, P_θ becomes a better predictor of which structural actions have high advantage. System 2 (the MPG) produces labels via rollouts, and System 1 (the proposal network) learns from these labels to propose better candidates. The improved proposals allow the MPG to focus compute on a smaller set of high-value edits. In this sense, the Reflective Layer is the mechanism by which the agent learns how to feel about its own structural changes.

Algorithm 1 Reflective Structural Step in HR-EGWM

Require: Structural state $s_t = (W_t, G_t, M_t, \mathcal{V}_t)$; proposal network P_θ; structural world model M_G; planning horizon H; proposal budget $K = K_{\text{exploit}} + K_{\text{explore}}$.

Ensure: Updated structural state s_{t+1} and parameters θ.

$\mathcal{C}_t \leftarrow \text{EnumerateCandidates}(s_t)$
for all $c \in \mathcal{C}_t$ **do**
  $\phi(c, s_t) \leftarrow \text{FeatureVector}(c, s_t)$
  $\text{score}(c) \leftarrow s_\theta(c, s_t) = f_\theta(\phi(c, s_t))$
end for
$S_{\text{exploit}} \leftarrow \text{TopK}(\mathcal{C}_t, \text{score}(c), K_{\text{exploit}})$
$S_{\text{explore}} \leftarrow \text{SampleUniform}(\mathcal{C}_t \setminus S_{\text{exploit}}, K_{\text{explore}})$
$S_{\text{propose}} \leftarrow S_{\text{exploit}} \cup S_{\text{explore}}$
$S_{\text{eval}} \leftarrow S_{\text{propose}}$
for all $c \in S_{\text{eval}}$ **do**
  $R(c) \leftarrow \text{RolloutReturn}(M_G, s_t, c, H)$
end for
$R_{\text{base}} \leftarrow \text{RolloutReturn}(M_G, s_t, \text{baseline}, H)$
for all $c \in S_{\text{eval}}$ **do**
  $A(c) \leftarrow R(c) - R_{\text{base}}$
end for
$c^* \leftarrow \arg\max_{c \in S_{\text{eval}}} A(c)$
$a_t^* \leftarrow \text{ActionFromCandidate}(c^*)$
$s_{t+1} \leftarrow \text{ApplyStructuralAction}(s_t, a_t^*)$
for all $c \in S_{\text{eval}}$ **do**
  $y(c) \leftarrow \text{TargetFromAdvantage}(A(c))$ {1 if $A(c) > 0$, else 0}
  $\hat{y}(c) \leftarrow \sigma(\text{score}(c))$
end for
$\mathcal{L}_{\text{reflect}} \leftarrow \sum_{c \in S_{\text{eval}}} \ell(\hat{y}(c), y(c))$
$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{reflect}}$
return s_{t+1}, θ

2.5 Algorithm: Reflective Structural Step

2.6 Empirical Summary

Experiment	**Target Problem**	**Policy / Intuition**	**Final Sta**
Exp 1: Merge Pain (3→1)	Local Optimum Trap	Reactive Governor	3 experts,
Exp 2: Zombie Heads (3→1)	Proactive Structural Pain	MPG + Endurance	1 expert, T
Exp 3: Crowded Room (20→17)	Combinatorial Explosion	MPG + Static Intuition	≈ 19.66 ex
Exp 4: Hidden Redundancy (20→16)	Learned Structural Attention	MPG + Adaptive P_θ	16 experts,

Table 1: Progression of structural intelligence in EGWM: from reactive control to proactive, intuition-guided, and finally self-improving structural meta-control in HR-EGWM.