The Autopoietic Ladder

Self-Modulating Memory Architectures

From Fixed Decay to Pure Self-Reference

1 Introduction

The term **autopoiesis** (from Greek: self-creation) describes systems that produce and maintain themselves. In the context of recurrent neural networks, we ask: **how can a memory system modulate its own dynamics?**

This document presents a hierarchy of increasingly autopoietic architectures, each building on the last. The central question: how far can we push the principle of self-modulation while maintaining trainability and efficiency?

2 The Hierarchy of Automodulation

We present eight levels of automodulation, from fixed decay to continuous self-referential dynamics.

2.1 Level 0: Fixed Gating (E74)

E74: Fixed Decay Delta Rule

$$\boldsymbol{S}' = \alpha \cdot \boldsymbol{S} + \left(\boldsymbol{v} - \boldsymbol{S}\hat{\boldsymbol{k}}\right)\hat{\boldsymbol{k}}^\top \tag{1}$$

- $\alpha \in (0, 1)$: Fixed scalar decay
- No automodulation
- Decay is a hyperparameter, not learned

Properties:

- Simplest possible delta rule
- Decay cannot adapt to input or state
- Baseline for comparison

2.2 Level 1: Vector Gating (E75)

E75: Input-Dependent Vector Gate

$$\boldsymbol{g} = \sigma\left(\boldsymbol{W}_\beta \boldsymbol{x} + \boldsymbol{b}_\beta\right) \in (0, 1)^n \tag{2}$$

$$\boldsymbol{S}' = \mathrm{diag}(\boldsymbol{g}) \cdot \boldsymbol{S} + \left(\boldsymbol{v} - \boldsymbol{S}\hat{\boldsymbol{k}}\right)\hat{\boldsymbol{k}}^\top \tag{3}$$

- Per-row decay controlled by input
- n degrees of freedom in gating
- External modulation (input $\longrightarrow$ gate)

Properties:

- Input-dependent forgetting
- Still no state-dependence in gating
- Gate is "open loop" — doesn't see what S contains

2.3 Level 2: Cross-Matrix Gating, Rank-1 (E79)

E79: Mutual Rank-1 Gating

$$\boldsymbol{g}^S_{\text{row}} = \sigma\big(\boldsymbol{M}\hat{\boldsymbol{k}} + \boldsymbol{b}_S\big), \quad \boldsymbol{g}^S_{\text{col}} = \sigma\big(\boldsymbol{M}^\top\hat{\boldsymbol{k}} + \boldsymbol{b}_S\big) \tag{4}$$

$$\boldsymbol{S}' = \Big(\boldsymbol{g}^S_{\text{row}}(\boldsymbol{g}^S_{\text{col}})^\top\Big) \odot \boldsymbol{S} + \boldsymbol{\delta}_S\hat{\boldsymbol{k}}^\top \tag{5}$$

Symmetrically, S gates M:

$$\boldsymbol{g}^M_{\text{row}} = \sigma(\boldsymbol{S}\widehat{\boldsymbol{m}} + \boldsymbol{b}_M), \quad \boldsymbol{g}^M_{\text{col}} = \sigma(\boldsymbol{S}^\top\widehat{\boldsymbol{m}} + \boldsymbol{b}_M) \tag{6}$$

$$\boldsymbol{M}' = \Big(\boldsymbol{g}^M_{\text{row}}(\boldsymbol{g}^M_{\text{col}})^\top\Big) \odot \boldsymbol{M} + \boldsymbol{\delta}_M\widehat{\boldsymbol{m}}^\top \tag{7}$$

Properties:
- State-dependent gating (M sees S, S sees M)
- Mutual modulation — bidirectional coupling
- **Constraint:** Gate is rank-1 (outer product of two vectors)
- $2n$ parameters control n^2 decay rates

Insight: The rank-1 constraint means rows and columns cannot be gated independently. If row i decays, ALL of row i decays regardless of column.

2.4 Level 3: Cross-Matrix Gating, Full Rank (E80)

E80: Full-Rank Mutual Gating

$$\boldsymbol{G}^S = \sigma\Big(\boldsymbol{M} + \text{outer}\Big(\boldsymbol{M}\hat{\boldsymbol{k}}, \hat{\boldsymbol{k}}\Big) + \boldsymbol{B}_S\Big) \in (0,1)^{n\times n} \tag{8}$$

$$\boldsymbol{S}' = \boldsymbol{G}^S \odot \boldsymbol{S} + \boldsymbol{\delta}_S\hat{\boldsymbol{k}}^\top \tag{9}$$

The gate $\boldsymbol{G}^S$ is a full $n \times n$ matrix, not rank-1.

Symmetrically for M:

$$\boldsymbol{G}^M = \sigma(\boldsymbol{S} + \text{outer}(\boldsymbol{S}\widehat{\boldsymbol{m}}, \widehat{\boldsymbol{m}}) + \boldsymbol{B}_M) \tag{10}$$

$$\boldsymbol{M}' = \boldsymbol{G}^M \odot \boldsymbol{M} + \boldsymbol{\delta}_M\widehat{\boldsymbol{m}}^\top \tag{11}$$

Properties:
- Full n^2 degrees of freedom in gating
- Each element (i, j) can have independent decay
- The gate is computed FROM the other matrix but is not itself a hidden state

Variation — Rank-r Gating:

$$\boldsymbol{G} = \sigma\left(\sum_{i=1}^{r}\boldsymbol{u}_i\boldsymbol{v}_i^\top\right) \tag{12}$$

This interpolates between rank-1 ($r = 1$, E79) and full-rank ($r = n$, E80).

2.5 Level 4: Gate Matrix as Hidden State (E81)

E81: Evolving Gate Matrix
Two hidden states: $\boldsymbol{S}$ (content) and $\boldsymbol{G}$ (gate), both $n \times n$.

$$\boldsymbol{S}' = \sigma(\boldsymbol{G}) \odot \boldsymbol{S} + \boldsymbol{\delta}_S \hat{\boldsymbol{k}}^\top \tag{13}$$

$$\boldsymbol{G}' = \sigma(\boldsymbol{S}) \odot \boldsymbol{G} + \boldsymbol{\delta}_G \widehat{\boldsymbol{m}}^\top \tag{14}$$

where $\boldsymbol{\delta}_G = \boldsymbol{\delta}_S - \boldsymbol{G}\widehat{\boldsymbol{m}}$ (G learns to predict S's changes).

Properties:
- The gate itself is a hidden state that evolves over time
- G has memory — it accumulates information about good gating strategies
- Mutual modulation: S gates G, G gates S
- Both matrices use delta rule updates

Insight: In E81, the gate is not just computed — it is **learned online** as a hidden state. G develops a "theory" of when S should forget.

2.6 Level 5: Self-Gating Matrix (E82)

E82: Pure Self-Modulation
Single matrix $\boldsymbol{S} \in \mathbb{R}^{n \times n}$ that gates itself:

$$\boldsymbol{G} = \sigma\Big(\boldsymbol{S}\widehat{\boldsymbol{m}}\hat{\boldsymbol{k}}^\top + \alpha \cdot \boldsymbol{S}\Big) \tag{15}$$

$$\boldsymbol{S}' = \boldsymbol{G} \odot \boldsymbol{S} + \boldsymbol{\delta}_S \hat{\boldsymbol{k}}^\top \tag{16}$$

The gate is computed from S itself — no separate modulation matrix.

Properties:
- Minimal architecture: single matrix
- Maximum autopoiesis: S determines its own forgetting
- Fixed point dynamics: S must find self-consistent evolution
- Risk: degenerate solutions (all-forget or all-remember)

Stabilization strategies:
- Use different key projections for gating vs. content
- Add skip connection: $\boldsymbol{G} = \sigma(\ldots) + \varepsilon \cdot \boldsymbol{I}$
- Regularize toward moderate gating

2.7 Level 6: Circular K-Tower (E83)

E83: Circular Mutual Gating
K matrices $\boldsymbol{M}_0, \boldsymbol{M}_1, \ldots, \boldsymbol{M}_{K-1}$, each $n \times n$.

Each matrix is gated by the next (modulo K):

$$\boldsymbol{G}_i = \sigma\Big(\boldsymbol{M}_{(i+1) \bmod K} \hat{\boldsymbol{k}}_i \hat{\boldsymbol{k}}_i^\top + \boldsymbol{B}_i\Big) \tag{17}$$

$$M_i' = G_i \odot M_i + \delta_i \hat{k}_i^\top \tag{18}$$

For $K = 3$: $M_0 \leftarrow M_1 \leftarrow M_2 \leftarrow M_0$ (circular)

Properties:
- No "top" of the hierarchy — circular dependency
- Distributed autopoiesis across K matrices
- Each matrix is both controller and controlled
- Richer dynamics than pairwise coupling

The K=2 case recovers E79/E80 (mutual pair).

2.8 Level 7: Continuous Dynamics (E84)

E84: Neural ODE Automodulation
Continuous-time evolution:

$$\frac{dS}{dt} = -S + \sigma(G) \odot S + \text{outer}\left(v - S\hat{k}, \hat{k}\right) \tag{19}$$

$$\frac{dG}{dt} = -G + \sigma(S) \odot G + \text{outer}(\delta_S - G\hat{m}, \hat{m}) \tag{20}$$

Integrate from $t = 0$ to $t = T$ using ODE solver.

Properties:
- Adaptive computation: harder inputs → more integration steps
- Smooth dynamics, potentially better gradients
- The system finds its own "clock"
- Use adjoint method for memory-efficient gradients

3 Comparison Table

Level	Gate Rank	State Size	Gate DOF	Key Property
E74	0 (scalar)	n^2	1	Fixed decay
E75	diag	n^2	n	Input-dependent
E79	1	$2n^2$	$2n$	Mutual, rank-1
E80	n	$2n^2$	n^2	Mutual, full-rank
E81	n	$2n^2$	n^2 evolving	Gate as state
E82	n	n^2	n^2 self	Self-gating
E83	n each	Kn^2	Kn^2	Circular tower
E84	n	$2n^2$	continuous	Neural ODE

4 The Information-Theoretic View

4.1 Bits of Control

Each level provides different amounts of information for gating:

- **E74:** 0 bits (fixed)
- **E75:** $n \log_2\left(\frac{1}{\varepsilon}\right)$ bits (n scalar gates at precision ε)

- **E79:** $2n \log_2\left(\frac{1}{\varepsilon}\right)$ bits (2n values → rank-1 gate)
- **E80:** $n^2 \log_2\left(\frac{1}{\varepsilon}\right)$ bits (full gate matrix)

4.2 The Compression Principle

Insight: There's a tradeoff: more gating flexibility requires more parameters/computation, but may enable better compression of the input sequence into fixed-size state.

The optimal level depends on:
1. Sequence complexity (more structure → benefit from richer gating)
2. Training budget (richer gating → harder to optimize)
3. Inference budget (richer gating → more compute per step)

5 Gradient Flow Analysis

5.1 E79: Rank-1 Constraint

Gradient from loss to M:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{M}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{S}'} \cdot \frac{\partial \boldsymbol{S}'}{\partial \boldsymbol{g}} \cdot \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{M}} \tag{21}$$

The bottleneck: $(\partial \frac{\boldsymbol{S}\}'}{\partial \boldsymbol{g}}$ only has rank-1 structure.

5.2 E80+: Full-Rank Gradient

With full-rank gating, every element of G receives independent gradient signal. This may enable:
- Faster learning of complex gating patterns
- Better credit assignment
- But also: risk of overfitting the gating

6 Stability Considerations

6.1 Fixed Points

Self-gating systems (E82) must avoid degenerate fixed points:
- $\boldsymbol{G} = \boldsymbol{0}$: Complete forgetting (S → 0)
- $\boldsymbol{G} = \boldsymbol{1}$: No forgetting (S accumulates without bound)

Mitigation:
- Initialize gate biases for moderate decay ($\sigma^{-1}(0.9) \approx 2.2$)
- Add regularization toward $\boldsymbol{G} \approx 0.5$
- Use spectral normalization on S

6.2 Circular Dependencies (E83)

The circular gating $\boldsymbol{M}_0 \leftarrow \boldsymbol{M}_1 \leftarrow \ldots \leftarrow \boldsymbol{M}_0$ creates:
- No clear "ground truth" — all matrices bootstrap each other
- Potential for oscillation or divergence
- Need careful initialization and learning rate scheduling

7 Implementation Considerations

7.1 Computational Cost

Level	Forward Cost	Backward Cost
E74	$O(n^2)$	$O(n^2)$
E75	$O(n^2 + nd)$	$O(n^2 + nd)$
E79	$O(n^2) \times 2$	$O(n^2) \times 2$
E80	$O(n^2) \times 2$	$O(n^2) \times 2$
E81	$O(n^2) \times 2$	$O(n^2) \times 2$
E82	$O(n^2)$	$O(n^2)$
E83	$O(Kn^2)$	$O(Kn^2)$
E84	$O(n^2 \times \text{steps})$	$O(n^2 \times \text{steps})$

7.2 CUDA Kernel Strategy

For each level, the kernel structure is similar:

1. Load state matrices into shared memory
2. Compute gates (level-specific)
3. Apply gated decay + delta rule update
4. Store results

The main difference is how gates are computed:

- E79: Two matrix-vector products $\longrightarrow$ outer product
- E80: Full matrix computation for gate
- E81: Same as E80, but gate persists across timesteps
- E82: Self-referential gate computation

8 Open Questions

1. **Optimal rank for gating:** Is there a sweet spot between rank-1 (E79) and full-rank (E80)?
2. **Initialization for self-gating:** How to initialize E82 to avoid degenerate fixed points?
3. **Circular vs. hierarchical:** Does the circular structure (E83) outperform linear hierarchy?
4. **Continuous vs. discrete:** When does E84's adaptive computation help?
5. **Biological plausibility:** Do neural circuits implement any of these patterns?

9 Conclusion

The autopoietic ladder reveals a spectrum of self-modulation strategies:

Level	Key Insight
E74	Baseline: no self-reference
E75	External modulation only
E79	Mutual modulation, rank-constrained
E80	Full-rank mutual modulation
E81	Gate itself evolves
E82	Pure self-reference

E83	Distributed circular control
E84	Continuous self-modulation

Each step up the ladder increases the system's ability to control its own structure. The research question is: **where is the sweet spot** between expressiveness and trainability?

The E79 benchmark results (1.51 loss, beating E1′s 1.53) suggest that even rank-1 mutual gating provides benefit. The higher levels remain to be empirically validated.