E79: Coupled Memory-Modulation Matrix System

A Mathematical Analysis of Hierarchical Delta Rules

Formal verification in Lean 4 with Mathlib

1 Introduction

E79 represents the culmination of 79 architectural experiments in recurrent neural network design. Its key innovation is **coupled delta rules**: two $n \times n$ matrix states where the second learns to predict the residuals of the first.

This document provides:

1. Complete mathematical specification of E79
2. Analysis of how M modulates S
3. Jacobian and gradient flow analysis
4. Generalizations to K-level hierarchies
5. Conditions for simplification

2 Mathematical Specification

2.1 State Definition

E79 maintains two matrix states:

$$\boldsymbol{S} \in \mathbb{R}^{n \times n} \quad \text{Content Memory (primary associative storage)} \tag{1}$$

$$\boldsymbol{M} \in \mathbb{R}^{n \times n} \quad \text{Modulation Memory (controls S's gating)} \tag{2}$$

Total state: $2n^2$ real values. For $n = 32$, this is 2048 elements.

2.2 Input Vectors

At each timestep, E79 receives:

- $\boldsymbol{k} \in \mathbb{R}^n$: Key vector for content addressing
- $\boldsymbol{v} \in \mathbb{R}^n$: Value to store
- $\boldsymbol{q} \in \mathbb{R}^n$: Query for output
- $\boldsymbol{m} \in \mathbb{R}^n$: Modulation key for M addressing

2.3 The E79 Update Rule (Actual Implementation)

Input: State $(\boldsymbol{S}, \boldsymbol{M})$, vectors $(\boldsymbol{k}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{m})$, biases $(\boldsymbol{b}_S, \boldsymbol{b}_M)$

Step 1: Normalize keys

$$\hat{\boldsymbol{k}} = \frac{\boldsymbol{k}}{\|\boldsymbol{k}\|_2}, \quad \widehat{\boldsymbol{m}} = \frac{\boldsymbol{m}}{\|\boldsymbol{m}\|_2} \tag{3}$$

Step 2: M controls S's decay gates (M → S coupling)

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

$$\boldsymbol{g}^S_{\text{col}} = \sigma\big(\boldsymbol{M}^\top\hat{\boldsymbol{k}} + \boldsymbol{b}_S\big) \quad \text{(col decay from M)} \tag{5}$$

Step 3: S delta rule update with M-controlled gating

$$\boldsymbol{\delta}_S = \boldsymbol{v} - \boldsymbol{S}\hat{\boldsymbol{k}} \tag{6}$$

$$\boldsymbol{S}' = \big(\boldsymbol{g}^S_{\text{row}}\boldsymbol{g}^S_{\text{col}\{\}^\top}\big) \odot \boldsymbol{S} + \boldsymbol{\delta}_S\hat{\boldsymbol{k}}^\top \tag{7}$$

Step 4: S controls M's decay gates (S → M coupling)

$$\boldsymbol{g}^M_{\text{row}} = \sigma(\boldsymbol{S}\widehat{\boldsymbol{m}} + \boldsymbol{b}_M) \tag{8}$$

$$\boldsymbol{g}^M_{\text{col}} = \sigma(\boldsymbol{S}^\top\widehat{\boldsymbol{m}} + \boldsymbol{b}_M) \tag{9}$$

Step 5: M delta rule update (M predicts S's changes)

$$\boldsymbol{\delta}_M = \boldsymbol{\delta}_S - \boldsymbol{M}\widehat{\boldsymbol{m}} \tag{10}$$

$$\boldsymbol{M}' = \big(\boldsymbol{g}^M_{\text{row}}\boldsymbol{g}^M_{\text{col}\{\}^\top}\big) \odot \boldsymbol{M} + \boldsymbol{\delta}_M\widehat{\boldsymbol{m}}^\top \tag{11}$$

Step 6: Output with self-gating

$$\boldsymbol{o} = (\boldsymbol{S}'\boldsymbol{q}) \odot \text{silu}(\boldsymbol{S}'\boldsymbol{q}) \tag{12}$$

Return: New state $(\boldsymbol{S}', \boldsymbol{M}')$, output $\boldsymbol{o}$

Algorithm 1: E79 Forward Pass - Mutual Gating Control

2.4 Explicit Matrix Form

The key insight is the **factorized gating**:

$$\boldsymbol{S}' = \underbrace{\big(\boldsymbol{g}^S_{\text{row}}\boldsymbol{g}^S_{\text{col}\{\}^\top}\big)}_{\text{M-controlled decay}} \odot \boldsymbol{S} + \boldsymbol{\delta}_S\hat{\boldsymbol{k}}^\top \tag{13}$$

Where the decay gate is an outer product of M's outputs:

$$\boldsymbol{g}^S_{\text{row}}\boldsymbol{g}^S_{\text{col}\{\}^\top} = \sigma\big(\boldsymbol{M}\hat{\boldsymbol{k}}\big)\sigma\big(\boldsymbol{M}^\top\hat{\boldsymbol{k}}\big)^\top \tag{14}$$

This means **M directly influences S's update** and thus the output. Similarly:

$$\boldsymbol{M}' = \big(\boldsymbol{g}^M_{\text{row}}\boldsymbol{g}^M_{\text{col}\{\}^\top}\big) \odot \boldsymbol{M} + \boldsymbol{\delta}_M\widehat{\boldsymbol{m}}^\top \tag{15}$$

With S controlling M's decay gates.

3 How M Modulates S

3.1 The Coupling Mechanism (Actual Implementation)

Unlike the simplified description in E79_RESULTS.md, the **actual** E79 implements **mutual gating control**:

> **Key Insight:** M directly controls S's decay gates, and S controls M's decay gates. This creates a bidirectional dynamical coupling where each memory controls what the other forgets.

3.1.1 M → S Coupling (M controls S's forgetting)

The decay factors for S come from M:

$$\boldsymbol{g}^S_{\text{row}} = \sigma\Big(\boldsymbol{M}\hat{\boldsymbol{k}} + \boldsymbol{b}_S\Big) \in (0,1)^n \tag{16}$$

$$\boldsymbol{g}^S_{\text{col}} = \sigma\Big(\boldsymbol{M}^\top\hat{\boldsymbol{k}} + \boldsymbol{b}_S\Big) \in (0,1)^n \tag{17}$$

The S update becomes:

$$S'_{ij} = g^S_{\text{row},i} \cdot g^S_{\text{col},j} \cdot S_{ij} + (\delta_S)_i \hat{k}_j \tag{18}$$

M controls what S retains. When $\boldsymbol{M}\hat{\boldsymbol{k}}$ is large and positive, $\boldsymbol{g}^S_{\text{row}} \to 1$ and S preserves its rows. When negative, S forgets.

3.1.2 S → M Coupling (S controls M's forgetting)

Symmetrically, S controls M's decay:

$$\boldsymbol{g}^M_{\text{row}} = \sigma(\boldsymbol{S}\widehat{\boldsymbol{m}} + \boldsymbol{b}_M) \tag{19}$$

$$\boldsymbol{g}^M_{\text{col}} = \sigma(\boldsymbol{S}^\top\widehat{\boldsymbol{m}} + \boldsymbol{b}_M) \tag{20}$$

S controls what M retains. This creates a feedback loop where the memories regulate each other.

3.2 Gradient Flow Through M

Theorem (M Gets Gradients Through S)**.** M influences the output through the gating path:

$$\text{Loss} \to \boldsymbol{o} \to \boldsymbol{S}' \to \boldsymbol{g}^S_{\text{row}}, \boldsymbol{g}^S_{\text{col}} \to \boldsymbol{M} \tag{21}$$

Specifically:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{M}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{S}'} \cdot \frac{\partial \boldsymbol{S}'}{\partial \boldsymbol{g}^S} \cdot \frac{\partial \boldsymbol{g}^S}{\partial \boldsymbol{M}} \tag{22}$$

Proof. From Equation 7: $\boldsymbol{S}' = \Big(\boldsymbol{g}^S_{\text{row}}\boldsymbol{g}^S_{\text{col}\{\}^\top}\Big) \odot \boldsymbol{S} + \boldsymbol{\delta}_S\hat{\boldsymbol{k}}^\top$

The gradient of $\boldsymbol{S}'$ with respect to $\boldsymbol{g}^S_{\text{row}}$ is:

$$\frac{\partial S'_{ij}}{\partial g^S_{\text{row},i}} = g^S_{\text{col},j} \cdot S_{ij} \tag{23}$$

And $\boldsymbol{g}^S_{\text{row}} = \sigma\Big(\boldsymbol{M}\hat{\boldsymbol{k}} + \boldsymbol{b}_S\Big)$, so:

$$\frac{\partial g^S_{\text{row},i}}{\partial M_{il}} = \sigma'(\ldots) \cdot \hat{k}_l \tag{24}$$

Composing these gives a non-zero gradient path from Loss to M. □

3.3 Interpretation: Mutual Control Dynamical System

The E79 coupling creates a **self-organizing** memory system:

Aspect	Mechanism
M ⟶ S	M decides what S should forget based on current key
S ⟶ M	S decides what M should forget based on modulation key
S delta	Standard delta rule with M-modulated decay
M delta	Learns S's prediction errors for meta-learning

This is analogous to:

- **Neural gating** (LSTM): Forget gates control information flow
- **Attention** (Transformers): Context-dependent routing
- **Neuromodulation** (Biology): One system modulates another's plasticity

3.4 Why Mutual Control Helps

Proposition (Adaptive Forgetting)**.** With M-controlled gating, S can learn **input-dependent** forgetting:

- For familiar keys: M outputs high gates ⟶ S preserves old information
- For novel keys: M outputs low gates ⟶ S makes room for new content

This is impossible with fixed decay α_S.

Proposition (Meta-Learning Through Coupling)**.** M can learn to recognize **when** S should update strongly vs. weakly:

- Systematic input patterns ⟶ M learns predictable gating
- Noisy inputs ⟶ M learns to gate conservatively

This is a form of "learning to learn" for associative memory.

4 Jacobian Analysis

4.1 State Space Jacobian

The full E79 state is $\boldsymbol{z} = \text{vec}([\boldsymbol{S}; \boldsymbol{M}]) \in \mathbb{R}^{2n^2}$.

Theorem (Lower-Triangular Jacobian)**.** The Jacobian of the E79 update has block lower-triangular structure:

$$\frac{\partial \boldsymbol{z}'}{\partial \boldsymbol{z}} = \begin{pmatrix} \boldsymbol{J}_S & \boldsymbol{0} \\ \boldsymbol{J}_{MS} & \boldsymbol{J}_M \end{pmatrix} \tag{25}$$

where:

- $\boldsymbol{J}_S = \frac{\partial \boldsymbol{S}'}{\partial \boldsymbol{S}}$: How S affects S'
- $\boldsymbol{J}_M = \frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{M}}$: How M affects M'
- $\boldsymbol{J}_{MS} = \frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{S}}$: How S affects M' (the coupling!)

- $\frac{\partial \boldsymbol{S}'}{\partial \boldsymbol{M}} = \mathbf{0}$: M does not affect S' directly

Proof. From Equation 7, $\boldsymbol{S}'$ depends only on $\boldsymbol{S}$, not $\boldsymbol{M}$:

$$\boldsymbol{S}' = \alpha_S \boldsymbol{S} + \left(\boldsymbol{v} - \boldsymbol{S}\hat{\boldsymbol{k}}\right)\hat{\boldsymbol{k}}^\top \tag{26}$$

Therefore $\frac{\partial \boldsymbol{S}'}{\partial \boldsymbol{M}} = \mathbf{0}$.

From Equation 11, $\boldsymbol{M}'$ depends on both $\boldsymbol{S}$ and $\boldsymbol{M}$:

$$\boldsymbol{M}' = \alpha_M \boldsymbol{M} + \left(\underbrace{\boldsymbol{v} - \boldsymbol{S}\hat{\boldsymbol{k}}}_{\boldsymbol{\delta}_S} - \boldsymbol{M}\widehat{\boldsymbol{m}}\right)\widehat{\boldsymbol{m}}^\top \tag{27}$$

The dependence on $\boldsymbol{S}$ comes through $\boldsymbol{\delta}_S$:

$$\frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{S}} = \frac{\partial}{\partial \boldsymbol{S}}\left[\left(\boldsymbol{v} - \boldsymbol{S}\hat{\boldsymbol{k}} - \boldsymbol{M}\widehat{\boldsymbol{m}}\right)\widehat{\boldsymbol{m}}^\top\right] = -\hat{\boldsymbol{k}}\widehat{\boldsymbol{m}}^\top \tag{28}$$

(in the appropriate tensor form). □

4.2 Individual Block Jacobians

4.2.1 Content Memory Jacobian $\boldsymbol{J}_S$

For the delta rule update with decay:

$$\boldsymbol{S}' = \alpha_S \boldsymbol{S} + \left(\boldsymbol{v} - \boldsymbol{S}\hat{\boldsymbol{k}}\right)\hat{\boldsymbol{k}}^\top = \alpha_S \boldsymbol{S} + \boldsymbol{v}\hat{\boldsymbol{k}}^\top - \boldsymbol{S}\hat{\boldsymbol{k}}\hat{\boldsymbol{k}}^\top \tag{29}$$

Taking the derivative with respect to $\boldsymbol{S}$:

$$\boldsymbol{J}_S = \alpha_S \boldsymbol{I} - \hat{\boldsymbol{k}}\hat{\boldsymbol{k}}^\top \otimes \boldsymbol{I}_n \tag{30}$$

In terms of action on a perturbation $\delta\boldsymbol{S}$:

$$\delta\boldsymbol{S}' = \alpha_S \delta\boldsymbol{S} - \left(\delta\boldsymbol{S}\hat{\boldsymbol{k}}\right)\hat{\boldsymbol{k}}^\top \tag{31}$$

Theorem (S Jacobian Spectral Properties)**.** With $\left\|\hat{\boldsymbol{k}}\right\|_2 = 1$, the Jacobian $\boldsymbol{J}_S$ (as a linear map on $n \times n$ matrices) has eigenvalues:

- α_S with multiplicity $n^2 - n$ (eigenvectors: matrices with $\hat{\boldsymbol{k}}$ in null space)
- $\alpha_S - 1$ with multiplicity n (eigenvectors: outer products with $\hat{\boldsymbol{k}}$)

For stability, we need $|\alpha_S| \leq 1$ and $|\alpha_S - 1| \leq 1$, giving $\alpha_S \in [0, 1]$.

4.2.2 Modulation Memory Jacobian $\boldsymbol{J}_M$

Similarly:

$$\boldsymbol{J}_M = \alpha_M \boldsymbol{I} - \widehat{\boldsymbol{m}}\widehat{\boldsymbol{m}}^\top \otimes \boldsymbol{I}_n \tag{32}$$

Same spectral structure with $\widehat{\boldsymbol{m}}$ replacing $\hat{\boldsymbol{k}}$.

4.3 Gradient Flow Through the Coupling

The coupling term $\boldsymbol{J}_{MS}$ enables **gradient sharing**:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{S}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{S}'} \boldsymbol{J}_S + \frac{\partial \mathcal{L}}{\partial \boldsymbol{M}'} \boldsymbol{J}_{MS} \tag{33}$$

The second term means: **M's gradient signal flows back to S**.

> **Corollary.** When training E79 end-to-end, S receives gradients from:
> 1. Direct output path: $\boldsymbol{o} \to \boldsymbol{S}'$
> 2. Indirect coupling path: $\boldsymbol{o} \to \boldsymbol{M}' \to \boldsymbol{S}$ (through $\boldsymbol{\delta}_S$)
>
> This coupling allows M to "tell" S about systematic errors.

5 Exact Retrieval and Capacity

5.1 Single Write Exact Retrieval

> **Theorem** (Exact Retrieval)**.** If $\boldsymbol{S} = \boldsymbol{0}$ (empty memory) and we write $(\boldsymbol{v}, \boldsymbol{k})$ with $\|\boldsymbol{k}\| = 1$, then:
>
> $$\boldsymbol{S}'\hat{\boldsymbol{k}} = \boldsymbol{v} \tag{34}$$
>
> *Proof.*
>
> $$\begin{aligned} \boldsymbol{S}'\hat{\boldsymbol{k}} &= \left[\boldsymbol{0} + \left(\boldsymbol{v} - \boldsymbol{0} \cdot \hat{\boldsymbol{k}}\right)\hat{\boldsymbol{k}}^\top\right]\hat{\boldsymbol{k}} \\ &= (\boldsymbol{v})\hat{\boldsymbol{k}}^\top\hat{\boldsymbol{k}} \\ &= \boldsymbol{v}\left(\hat{\boldsymbol{k}}^\top\hat{\boldsymbol{k}}\right) \\ &= \boldsymbol{v} \cdot 1 = \boldsymbol{v} \end{aligned} \tag{35}$$

5.2 Orthogonal Keys Preserve Information

> **Theorem** (Selective Update)**.** If $\hat{\boldsymbol{k}}_1 \perp \hat{\boldsymbol{k}}_2$ (orthogonal keys), then writing $(\boldsymbol{v}_2, \boldsymbol{k}_2)$ does not affect retrieval with $\boldsymbol{k}_1$:
>
> $$\boldsymbol{S}'\hat{\boldsymbol{k}}_1 = \boldsymbol{S}\hat{\boldsymbol{k}}_1 \tag{36}$$
>
> *Proof.* The update adds $\left(\boldsymbol{v}_2 - \boldsymbol{S}\hat{\boldsymbol{k}}_2\right)\hat{\boldsymbol{k}}_2^\top$. Applying to $\hat{\boldsymbol{k}}_1$:
>
> $$\left[\left(\boldsymbol{v}_2 - \boldsymbol{S}\hat{\boldsymbol{k}}_2\right)\hat{\boldsymbol{k}}_2^\top\right]\hat{\boldsymbol{k}}_1 = \left(\boldsymbol{v}_2 - \boldsymbol{S}\hat{\boldsymbol{k}}_2\right)\left(\hat{\boldsymbol{k}}_2^\top\hat{\boldsymbol{k}}_1\right) = \boldsymbol{0} \tag{37}$$
>
> since $\hat{\boldsymbol{k}}_2^\top\hat{\boldsymbol{k}}_1 = 0$.

5.3 Capacity Analysis

> **Proposition** (E79 Capacity)**.** With orthonormal keys $\left\{\hat{\boldsymbol{k}}_i\right\}_{i=1}^n$ for S and $\left\{\widehat{\boldsymbol{m}}_j\right\}_{j=1}^n$ for M:
> - S can store n independent (value, key) pairs: n^2 real values
> - M can store n independent (residual, modulation-key) pairs: n^2 real values
> - Total: $2n^2$ real values (equal to state size)
>
> This is **optimal** capacity utilization.

6 Generalizations

6.1 K-Level Hierarchies

E79 is the $K = 2$ case of a general construction:

Definition (K-Level Coupled Memory)**.** For $K \geq 1$, define matrices $\boldsymbol{M}_0, \boldsymbol{M}_1, ..., \boldsymbol{M}_{K-1} \in \mathbb{R}^{n \times n}$ with:

$$\boldsymbol{r}_0 = \boldsymbol{v} - \boldsymbol{M}_0 \hat{\boldsymbol{k}}_0 \quad \text{(Level 0 residual)} \tag{38}$$

$$\boldsymbol{r}_i = \boldsymbol{r}_{i-1} - \boldsymbol{M}_i \hat{\boldsymbol{k}}_i \quad \text{for } i = 1, ..., K-1 \tag{39}$$

$$\boldsymbol{M}_i' = \alpha_i \boldsymbol{M}_i + \boldsymbol{r}_i \hat{\boldsymbol{k}}_i^\top \quad \text{(Each level's update)} \tag{40}$$

K	**Description**
1	Standard delta rule (E74). Single matrix S.
2	E79. Content memory S + Modulation memory M.
3	Triple hierarchy. S + M + N where N predicts M's residuals.
K	Chain of K residual predictors.

6.2 Diminishing Returns Conjecture

Conjecture. For a fixed compute budget, there exists an optimal K^* such that:
- $K < K^*$: Adding levels improves performance
- $K > K^*$: Additional levels have negligible benefit

K^* depends on:
1. **Task complexity**: How much structure exists in residuals
2. **Training time**: Deeper hierarchies need more convergence time
3. **State budget**: Each level costs n^2 parameters

The benchmark showing $n = 32$ optimal for 10-minute training suggests E79 is near optimal for that regime.

7 Simplifications

7.1 Tied Keys: $\boldsymbol{m} = \boldsymbol{k}$

Theorem (Tied Keys Reduction)**.** If $\boldsymbol{m} = \boldsymbol{k}$ (same key for both levels), then E79 reduces to a single delta rule on the combined matrix $\boldsymbol{S} + \boldsymbol{M}$.

Proof. With $\widehat{\boldsymbol{m}} = \hat{\boldsymbol{k}}$:

$$\boldsymbol{\delta}_M = \boldsymbol{v} - \boldsymbol{S}\hat{\boldsymbol{k}} - \boldsymbol{M}\hat{\boldsymbol{k}} = \boldsymbol{v} - (\boldsymbol{S} + \boldsymbol{M})\hat{\boldsymbol{k}} \tag{41}$$

Combined update:

$$\boldsymbol{S}' + \boldsymbol{M}' = \alpha_S \boldsymbol{S} + \alpha_M \boldsymbol{M} + \left(\boldsymbol{v} - \boldsymbol{S}\hat{\boldsymbol{k}}\right)\hat{\boldsymbol{k}}^\top + \left(\boldsymbol{v} - \boldsymbol{S}\hat{\boldsymbol{k}} - \boldsymbol{M}\hat{\boldsymbol{k}}\right)\hat{\boldsymbol{k}}^\top \tag{42}$$

If $\alpha_S = \alpha_M = \alpha$:

$$\boldsymbol{S}' + \boldsymbol{M}' = \alpha(\boldsymbol{S} + \boldsymbol{M}) + \left[2\boldsymbol{v} - 2\boldsymbol{S}\hat{\boldsymbol{k}} - \boldsymbol{M}\hat{\boldsymbol{k}}\right]\hat{\boldsymbol{k}}^\top \tag{43}$$

This is **not** exactly a single delta rule, but the key insight is: tied keys limit M's ability to organize independently.

7.2 Zero M Decay: $\alpha_M = 0$

Proposition (Instantaneous Modulation)**.** With $\alpha_M = 0$:

$$\boldsymbol{M}' = \boldsymbol{\delta}_M \widehat{\boldsymbol{m}}^\top \tag{44}$$

M becomes an "instantaneous" residual predictor with no memory of past residuals. This is useful when residuals have no temporal structure.

7.3 No Modulation: $M = 0$

Proposition (Reduction to E74)**.** Setting $\boldsymbol{M} = \boldsymbol{0}$ and $\alpha_M = 0$ recovers E74 (single delta rule with self-gating):

$$\boldsymbol{S}' = \alpha_S \boldsymbol{S} + \left(\boldsymbol{v} - \boldsymbol{S}\hat{\boldsymbol{k}}\right)\hat{\boldsymbol{k}}^\top \tag{45}$$

$$\boldsymbol{o} = (\boldsymbol{S}'\boldsymbol{q}) \odot \text{silu}(\boldsymbol{S}'\boldsymbol{q}) \tag{46}$$

8 Empirical Results Summary

From the benchmark (100M params, 10-minute training):

Model	Loss	tok/s	State
Mamba2	1.27	78.7K	SSM (parallel)
E79 n=32	**1.51**	**31.5K**	**$2\times32^2 = 2048$**
E1 (gated)	1.53	45.5K	vector
E42 (linear)	1.59	137K	vector
FLA-GDN	1.99	18.7K	matrix

Key observations:

- E79 beats E1 (1.51 vs 1.53): modulation helps
- n=32 optimal for 10-min training (larger n under-converged)
- 40% of Mamba2 throughput despite sequential scan

9 Conclusions

9.1 What E79 Teaches Us

1. **Hierarchical error correction works**: Even one level of residual prediction (M on S) provides measurable benefit.
2. **Separate addressing enables specialization**: $\boldsymbol{k}$ for content, $\boldsymbol{m}$ for error patterns.
3. **Gradient coupling is essential**: M receives loss-relevant gradients through $\boldsymbol{\delta}_S$, not just residual reconstruction loss.
4. **Training time vs capacity tradeoff**: Larger state needs more training to converge.

9.2 Open Questions

1. What is optimal K for K-level hierarchies?
2. Can we learn the coupling adaptively?
3. How does E79 scale beyond 100M parameters?
4. Can parallel scan be applied to coupled matrices?