# Improved State Mixing in Higher-order and Block Diagonal Linear Recurrent Networks

Igor Dubinin[1] Antonio Orvieto[2] Felix Effenberger[1]

[1]Natural Intelligence & Ernst Strüngmann Institute [2]ELLIS Institute Tübingen & MPI-IS

NeurIPS 2026, Sydney















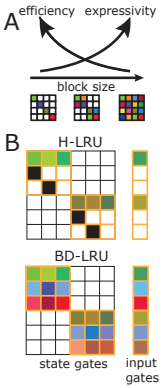
Paper: arXiv:2602.12021 Code: github.com/IgorDn8/mad-lab, github.com/IgorDn8/plainLM

# Motivation: the efficiency–expressivity trade-off

- Linear RNNs and SSMs (Mamba, LRU) are efficient because the state transition is **diagonal**.
- A diagonal transition **cannot mix states**, so these models are provably less expressive.
- Dense recurrences (LSTM) are expressive but less efficient.

**Can we buy expressivity without giving up efficiency?**

Our answer: **structured state mixing** with small dense blocks of size $m$, between the two extremes.




**A.** Block size $m$ moves from diagonal to dense. **B.** Selective gates in block form (colored: input-dependent).

# Two architectures: mixing across channels or across time

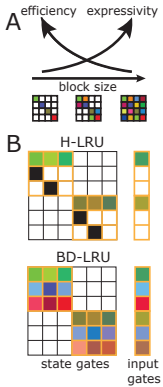
**BD-LRU**: dense **channel mixing** inside each block $k$ ($N$ blocks of size $m$, $H = Nm$)

$$\mathbf{h}_t^k = \mathbf{A}_t^k \, \mathbf{h}_{t-1}^k + \mathbf{a}_{0,t}^k \odot \mathbf{v}_t^k$$

**H-LRU**: **temporal mixing**, an $m$-th order recurrence (window $m$)

$$\mathbf{h}_t = \sum_{i=1}^{m} \mathbf{a}_{i,t} \odot \mathbf{h}_{t-i} + \mathbf{a}_{0,t} \odot \mathbf{v}_t$$

- Stacking the last $m$ states gives the same block form with a companion matrix.
- All recurrent connections are **selective gates**: they depend on the input.
- Gates per block: BD-LRU $m(m+1)$, H-LRU only $m+1$ (parameter-efficient).



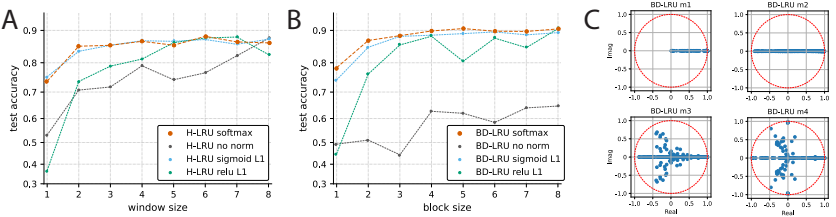

**A.** Block size $m$ moves from diagonal to dense. **B.** Selective gates in block form (colored: input-dependent).

# Joint selective gate normalization

Unconstrained selective gates make the recurrence unstable. We L1-normalize each row of state gates **together with** its input gate:

$$\textbf{BD-LRU: } a_{i,j,t} = \frac{f(a'_{i,j,t})}{\sum_{l=0}^{m} |f(a'_{i,l,t})|} \qquad \textbf{H-LRU: } a_{j,t} = \frac{f(a'_{j,t})}{\sum_{l=0}^{m} |f(a'_{l,t})|}$$

- We prove stable recurrence and bounded forward pass for any $m$.




**A/B.** Compression accuracy vs. $m$ for gate parametrizations $f$: only with normalization do both models improve with $m$. **C.** BD-LRU eigenvalues on $S_5$ become complex from $m = 3$.

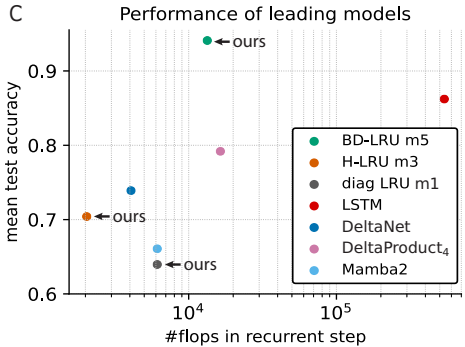
# State tracking: permutation composition

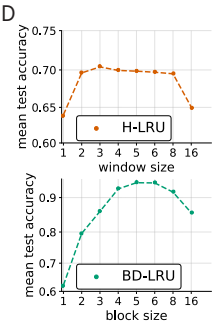
Input: a sequence of permutations from $S_n$. Target at every step: their composition so far.

| **Model** | $\mathbf{S_3}$ (10k) | $\mathbf{S_3}$ (250) | $\mathbf{S_4}$ (50k) | $\mathbf{S_4}$ (3k) | $\mathbf{S_5}$ (100k) | **Overall** |
|---|---|---|---|---|---|---|
| LSTM | **1.000** | 0.316 | **1.000** | 0.368 | **1.000** | 0.737 |
| Mamba2 | 0.657 | 0.281 | 0.429 | 0.119 | 0.261 | 0.349 |
| DeltaNet[-1,1] | **1.000** | 0.264 | 0.507 | 0.138 | 0.140 | 0.410 |
| DeltaProduct$_4$[-1,1] | **1.000** | 0.279 | **1.000** | 0.136 | 0.140 | 0.511 |
| H-LRU m5 | 0.622 | 0.318 | 0.448 | 0.194 | 0.377 | 0.392 |
| BD-LRU m1 | 0.563 | 0.381 | 0.343 | 0.218 | 0.215 | 0.344 |
| BD-LRU m2 | **1.000** | 0.488 | 0.699 | 0.356 | 0.338 | 0.576 |
| BD-LRU m3 | **1.000** | **1.000** | **1.000** | 0.426 | 0.484 | 0.782 |
| BD-LRU m4 | **1.000** | **1.000** | **1.000** | **1.000** | 0.882 | 0.976 |
| BD-LRU m5 | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** |

Columns: group and number of training samples. **Each block size unlocks a larger group**; BD-LRU is the only model that solves every task, also with scarce data.

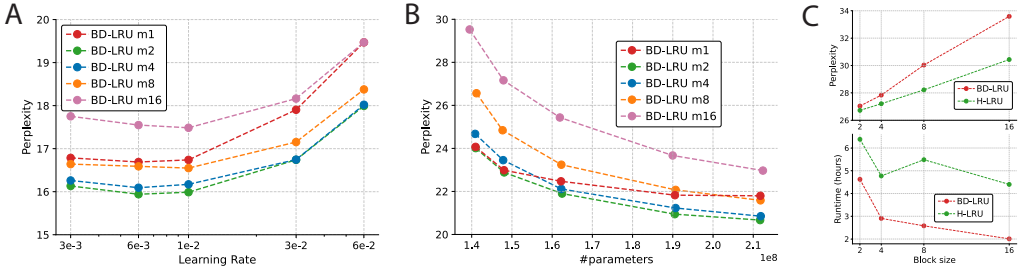
# Across synthetic tasks: performance vs. cost




- Mean accuracy over our set of synthetic tasks vs. FLOPs per recurrent step.
- **BD-LRU matches or beats LSTM, Mamba2, DeltaNet and DeltaProduct at lower cost**; H-LRU is the most parameter-efficient.
- Performance improves with window or block size $m$ and saturates at moderate $m$.

# Language modeling (FineWeb-Edu)




**A.** Perplexity vs. learning rate (210M parameters, 10B tokens). **B.** Scaling with hidden size (2.5B tokens). **C.** Runtime vs. perplexity (140M).

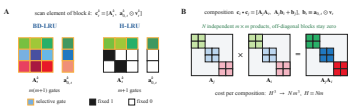
- **Block size improves perplexity** over the diagonal model and keeps improving with hidden size where the diagonal model saturates.

# Block-diagonal parallel scan in PyTorch and Triton

A parallel scan composes steps $\mathbf{c}_t = [\mathbf{A}_t, \mathbf{b}_t]$ in $\log_2 T$ levels:

$$\mathbf{c}_i \bullet \mathbf{c}_j = [\mathbf{A}_j \mathbf{A}_i,\ \mathbf{A}_j \mathbf{b}_i + \mathbf{b}_j]$$

Dense: $H^3$ per composition. Ours stays block-diagonal: $Nm^3$, within $O(m^2)$ of a diagonal scan.




| Impl. (ms / MB) | 256 | 1k | 4k | 16k | 64k |
|---|---|---|---|---|---|
| Sequential | 78.4 / 178 | 300 / 461 | 2383 / 1597 | – | – |
| Hopscan (PyTorch) | 16.9 / 217 | 20.8 / 618 | 49.7 / 2219 | 195 / 8616 | 776 / 34199 |
| Hopscan + `compile` | 3.2 / 132 | 3.4 / 227 | 9.9 / 406 | 37.4 / 1366 | – |
| Triton | 3.2 / 208 | 3.3 / 590 | 7.1 / 2109 | 26.7 / 8181 | 106 / 32456 |
| Triton + `compile` | 3.7 / 132 | 3.8 / 228 | 4.5 / 546 | 16.5 / 1932 | 64.8 / 7478 |

Columns: sequence length $T$. Training-step time / peak memory, BD-LRU $m = 4$, $B = 4$, ∼1M parameters, H100, bf16; "–": over budget.

- Parallel scans: **orders of magnitude faster** than the sequential reference.
- Compiled Triton kernel is **fastest at every sequence length**.

# Block size vs. Householder rank: the cost of expressivity

| T | Model (ms / MB) | rank 1 | rank 2 | rank 4 | rank 8 |
|---|---|---|---|---|---|
| 4096 | BD-LRU (block size $m$) | **4.0** / 620 | **4.5** / 516 | **4.5** / 546 | **6.4** / 1447 |
| | H-LRU (window size $m$) | 4.1 / 620 | 5.7 / 790 | 7.7 / 1333 | 17.6 / 6678 |
| | DeltaNet / DeltaProduct ($H_n$) | 7.7 / 1203 | 8.4 / 1646 | 10.2 / 1983 | 11.8 / 2204 |
| | LSTM | 89.2 / 299 | – | – | – |
| 16384 | BD-LRU (block size $m$) | **14.4** / 2228 | **16.3** / 1821 | **16.5** / 1932 | **24.2** / 5539 |
| | H-LRU (window size $m$) | 14.3 / 2228 | 20.7 / 2910 | 29.1 / 5083 | 69.1 / 26462 |
| | DeltaNet / DeltaProduct ($H_n$) | 22.1 / 4512 | 33.5 / 6330 | 43.7 / 7683 | 49.2 / 8570 |
| | LSTM | 346 / 919 | – | – | – |

Time / memory per training step at matched parameters (∼1M, $B = 4$) vs. block or window size (ours) or Householder rank $H_n$ (DeltaNet: $H_n = 1$).

- **Moderate block sizes ($m \leq 4$) cost about the same as a diagonal model**
- At every matched rank **BD-LRU is faster and leaner than the DeltaNet family**, and the margin grows with $T$.

# Key findings

- **The structure of state mixing drives performance;** joint gate normalization is what lets both models improve with $m$.
- **Fit the inductive bias to the task:** channel mixing (BD-LRU) for state tracking, temporal mixing (H-LRU) for compression.
- **Language modeling:** block size improves perplexity.
- **Moderate block sizes perform best** and, with our scans, cost about as much as diagonal models.
- **Block size buys expressivity cheaper than Householder rank.**

Paper: arXiv:2602.12021 Code: github.com/IgorDn8/mad-lab, github.com/IgorDn8/plainLM
Poster session: Tuesday 8 December, 5–8 PM, Hall 1–4 Contact: igor@nisys.ai

