Nested Learning

The Illusion of Deep Learning Architectures

Paper discussion

Paper authors: Ali Behrouz, Meisam Razaviyayn,
Peilin Zhong, and Vahab Mirrokni

arXiv:2512.24695v1 | 10 October 2026

Design what learns, how often it learns, and how knowledge persists.

Deployment leaves a gap between two memories

Persistent parameters

FFNs and projections retain pre-training knowledge.

Usually frozen during inference.

Immediate context

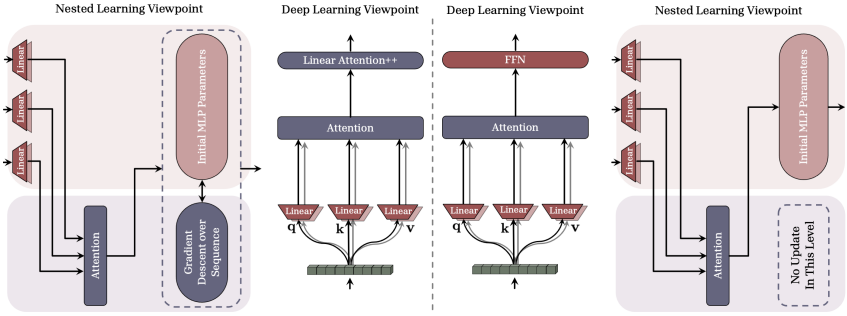
Attention reads the current KV history.

Knowledge disappears when this context is discarded.

- More network layers do not specify how new knowledge is consolidated.
- NL adds a design axis: learning processes at different time scales.

Paper, Secs. 1–1.1, PDF pp. 1–4. Brain analogies motivate the design.

A learning level differs from a network layer



- A learned initial memory supplies persistent knowledge.
- An additional update process lets that memory adapt within a sequence.

Paper, Fig. 3, PDF p. 10; Secs. 3.1–3.2, pp. 7–13.

Associative memory unifies several update rules

$$\mathcal{M}^* = \arg\min_{\mathcal{M}} \widetilde{\mathcal{L}}(\mathcal{M}(\mathcal{K}); \mathcal{V})$$

Linear-layer training

$$\nabla_W \mathcal{L} = \boldsymbol{\delta}\boldsymbol{x}^\top$$

$$W^+ = W - \eta\boldsymbol{\delta}\boldsymbol{x}^\top$$

Input $\boldsymbol{x}$ is associated with a local error signal $\boldsymbol{\delta}$.

Linear attention

$$\widetilde{\mathcal{L}} = -\langle \mathcal{M}\boldsymbol{k}, \boldsymbol{v} \rangle$$

$$\mathcal{M}^+ = \mathcal{M} + \boldsymbol{v}\boldsymbol{k}^\top$$

Key $\boldsymbol{k}$ is associated with value $\boldsymbol{v}$ (step size 1).

Backpropagation is self-referential: $\boldsymbol{\delta}$ depends on current weights.

Paper, Sec. 3.1, PDF pp. 7–9; Sec. 4.1, Eqs. (29)–(31), p. 15.

Optimizers learn from a stream of gradients

$$\boldsymbol{m}_{t+1} = \alpha_t \boldsymbol{m}_t - \eta_t \nabla_W \mathcal{L}_t, \qquad W_{t+1} = W_t + \boldsymbol{m}_{t+1}$$

- Momentum compresses gradient history; weights use the resulting update.
- The network generates the optimizer's context, so their designs interact.
- NL suggests richer objectives, nonlinear memory, feature maps, and multiple time scales for momentum.

A useful distinction

An associative-memory interpretation is a design tool. Optimality for an internal regression objective does not imply optimality for the original task.

Paper, Secs. 4.2–4.4, PDF pp. 15–20; Appendix B, pp. 50–51.

DGD adds explicit shrinkage along the input direction

Let $\boldsymbol{u}_t = -\nabla_{\boldsymbol{y}_t}\mathcal{L}(W_t; \boldsymbol{x}_t)$ be fixed for this update.

$$W^+ = \arg\min_W \left\{ \tfrac{1}{2}\|W\boldsymbol{x}_t - \boldsymbol{u}_t\|^2 + \tfrac{1}{2\eta_t}\|W - W_t\|_F^2 \right\}$$

$$\boxed{W^+ = W_t(I - \gamma_t \boldsymbol{x}_t\boldsymbol{x}_t^\top) - \gamma_t \nabla_{\boldsymbol{y}_t}\mathcal{L}\,\boldsymbol{x}_t^\top} \qquad \gamma_t = \frac{\eta_t}{1 + \eta_t\|\boldsymbol{x}_t\|^2}$$

- Reduce the old component along $\boldsymbol{x}_t$, then write a new error signal.
- Unit-norm inputs give $\gamma_t = \eta_t/(1+\eta_t)$, as in the main text.
- This changes the update objective; ordinary GD already depends on state.

Paper, Eqs. (56)–(57), PDF p. 20. General-norm solution re-derived here; Appendix C has inconsistent intermediate steps.

CMS provides several rates of adaptation and retention

$$\boldsymbol{y}_t = \mathrm{MLP}^{(f_k)} \circ \cdots \circ \mathrm{MLP}^{(f_1)}(\boldsymbol{x}_t)$$

Memory	Update schedule	Intended role
Fast	Short interval	Adapt to recent evidence
Intermediate	Longer interval	Retain across local contexts
Slow	Rare updates	Preserve more persistent knowledge

- Update each block only at its scheduled boundary.
- Connect levels through reads, gradients, or learned initial states.
- Slower updates help only if useful knowledge reaches those states.

Paper, Sec. 7.1, Eqs. (70)–(74), PDF pp. 27–29. Schedule table is an explanatory summary.

Hope's fast memory generates its own update targets

$$\boldsymbol{k}_t = \mathcal{M}_{k,t-1}(\boldsymbol{x}_t), \qquad \boldsymbol{v}_t = \mathcal{M}_{v,t-1}(\boldsymbol{x}_t),$$
$$\eta_t = \mathcal{M}_{\eta,t-1}(\boldsymbol{x}_t), \qquad \alpha_t = \mathcal{M}_{\alpha,t-1}(\boldsymbol{x}_t).$$

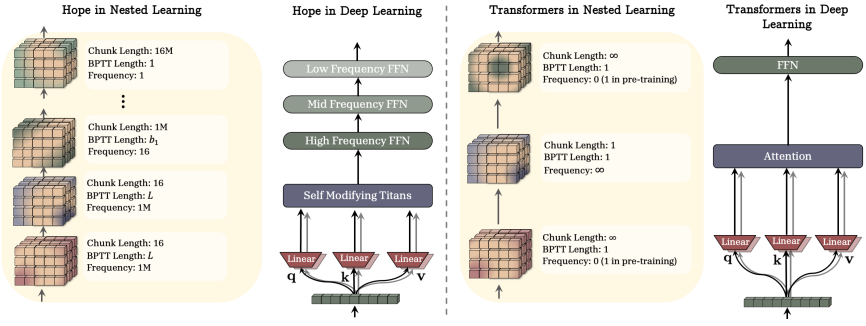
For each memory module $\mathcal{M}_\square$:

$$\widehat{\boldsymbol{v}}_{\square,t} = \mathcal{M}_{\square,t-1}(\boldsymbol{v}_t), \qquad \widetilde{\mathcal{L}}_\square = \|\mathcal{M}_\square(\boldsymbol{k}_t) - \widehat{\boldsymbol{v}}_{\square,t}\|_2^2.$$

- Each module learns a key-to-target mapping using DGD and retention.
- The state influences both the representation and its learning signal.
- Two-layer residual MLP memories; initial states learned across contexts.

Paper, Sec. 8.1, Eqs. (83)–(89), PDF pp. 31–32. Targets are generated from the pre-update state.

Hope combines an adaptive memory with CMS



- **Hope:** self-modifying Titans $\rightarrow$ CMS.
- **Hope-Attention:** global softmax attention $\rightarrow$ CMS.

Paper, Fig. 5, PDF p. 30; Sec. 8.3, Eqs. (94)–(97), p. 33. Diagram values are illustrative, not shared experiment settings.

Chunking trades immediate adaptation for parallel work

1. Freeze the state available at the start of a chunk.
2. Generate projections, targets, and gradients for that chunk in parallel.
3. Apply scheduled memory updates and advance to the next chunk.

Benefit

Fewer sequential nonlinear gradient evaluations; matrix operations inside chunks.

Cost and limitation

Extra adaptive states and update work. Chunk-start gradients differ from fresh token-wise gradients.

Paper, Sec. 8.2, PDF pp. 32–33; text describes the previous chunk's state. Eq. (90)'s printed index is inconsistent with that description.

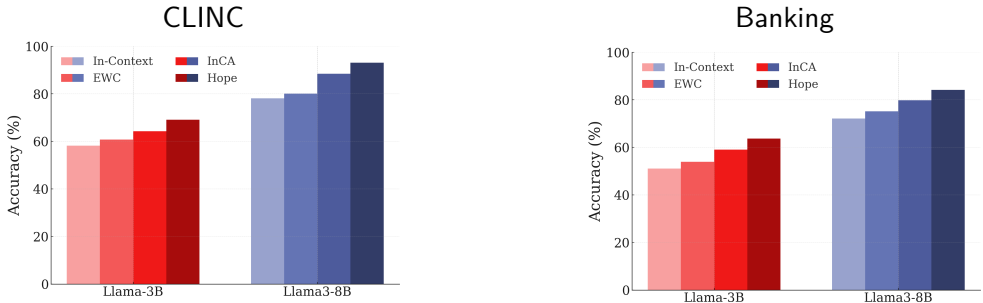
The experiments evaluate distinct systems and budgets

Question	System	Training / evaluation protocol
Continual learning	Llama-based Hope-Attention	Class-incremental: 3B/8B backbones, 15B extra tokens; matched ICL token budget
Long context	Hope from scratch	About 50B tokens; BABILong small models additionally task-fine-tuned
Language modeling	Hope and architecture baselines	760M / 30B tokens; 1.3B / 100B tokens; AdamW
Optimizer quality	M3 on ViT; timing on Transformer	ViT 24M/86M; timing at 140M/1.3B

M3 is a separate experiment: the Hope language models use AdamW.

Paper, Secs. 9.1–9.3 and 9.7, PDF pp. 34–40.

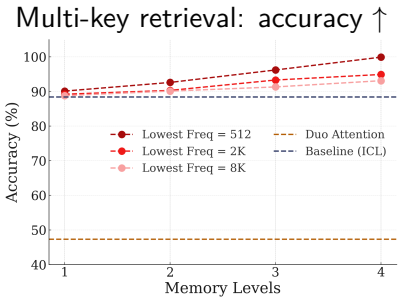
CMS improves the reported class-incremental results

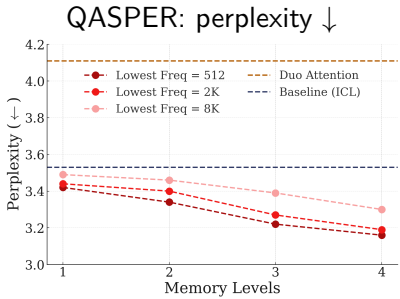


- Hope-Attention leads ICL, EWC, and the external-learner baseline InCA.
- Both 3B and 8B groups improve; DBpedia follows the same pattern.
- This experiment directly tests the CMS adaptation of pre-trained FFNs.

Paper, Fig. 6 (selected original panels), PDF p. 33; setup in Sec. 9.1, p. 34. No exact values inferred from bar heights.

More memory levels improve these context tasks

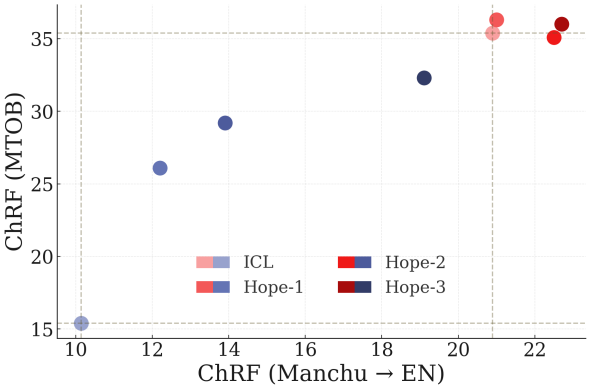




- Higher level counts improve both metrics in the shown configurations.
- The 512 group performs best; the paper proposes 2K as an efficiency compromise.

Paper, Fig. 7, PDF p. 34; discussion pp. 35–36. "Lowest Freq" labels are retained as printed; their units are not consistently specified.

Additional levels reduce interference across languages



- Learn Manchu and Kalamang, then translate into English.
- Red: learn and test separately.
- Blue: learn sequentially, then test.
- More CMS levels move the sequential result toward the upper right.

Paper, Fig. 8, PDF p. 35; CTNL protocol, Sec. 9.1, p. 36. Both axes are ChrF scores (higher is better).

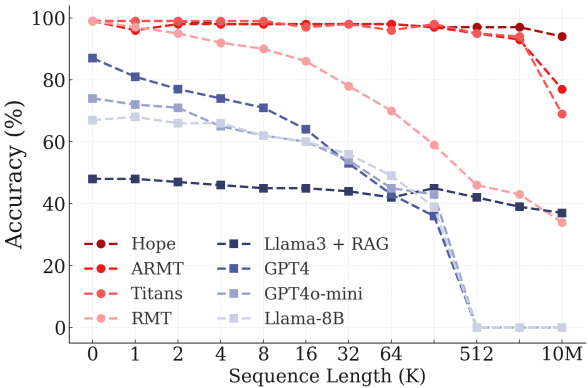
At 16K, Hope improves compressed-memory retrieval

Model	Number	UUID	Multi-key	Multi-query
Transformer	94.2	40.8	61.4	29.8
Hope-Attention	94.4	42.4	60.8	30.6
RWKV-7	12.6	5.8	9.6	8.6
Comba	13.4	7.4	8.2	6.4
Titans	75.4	21.2	8.2	9.4
Hope	**78.2**	**24.8**	**14.8**	**14.2**

- Better than the shown attention-free baselines at this length.
- A large precision gap remains: multi-key 14.8 vs. Transformer 61.4.
- At 4K, Titans slightly beats Hope on number and UUID retrieval.

Paper, Table 1, PDF p. 35. Selected 16K accuracies (%). Setup: about 50B training tokens, Sec. 9.2.

The 10M BABILong result uses task fine-tuning



- Hope retains strong performance at the longest plotted context.
- Red curves: task-fine-tuned small models.
- Blue curves: zero-shot large models / RAG baseline.
- The paper reports substantial degradation without fine-tuning.

Paper, Fig. 9, PDF p. 35; protocol and limitations, Sec. 9.2, pp. 36–37. This is not a matched-compute comparison to general-purpose large models.

Hope improves language modeling at both reported scales

Parameters / tokens	Model	Wiki PPL↓	LMB PPL↓	Avg.↑
760M / 30B	Transformer++	24.18	24.27	50.11
	Titans	20.08	21.52	51.68
	Hope	**18.68**	**20.07**	**52.28**
1.3B / 100B	Transformer++	17.92	17.73	53.38
	Titans	15.60	11.41	56.82
	Hope	**14.39**	**10.08**	**58.04**

- At 1.3B vs. Titans: Wiki PPL -7.76%; average accuracy $+1.22$ pp.
- Eight-task average; individual tasks still have exceptions (e.g., BoolQ).
- Model size and token budget change together across the two scales.

Paper, Table 2, PDF p. 37; training details pp. 37–38. PPL = perplexity; LMB = LAMBADA; pp = percentage points.

Natural recall still favors full attention

Short-context recall: scores ↑

Model	SWDE	NQ	FDA
Transformer	71.4	22.0	67.3
Titans	60.8	20.3	37.6
Hope	65.9	21.2	41.9

Synthetic MAD: scores ↑

Hope vs. Transformer:

- Fuzzy ICR: 52.1 vs. 47.9
- Selective copy: 99.7 vs. 96.2
- Memory: 85.2 vs. 83.7
- Standard ICR: all models 100

Different strengths: precise access to stored context versus learned state updates.

Paper, Tables 3–4, PDF p. 38. Selected columns; Table 3 does not specify a common EM/F1 scoring convention.

State tracking benefits from nonlinear recurrence

Model	Parity (Bin0/Bin1)	$(aa)^*$	$(abab)^*$
Transformer	46.4 / 0.0	0.0 / 0.0	0.0 / 0.0
DeltaNet	98.2 / 10.1	0.0 / 0.0	0.0 / 0.0
LSTM	100 / 100	100 / 100	100 / 100
SRWM	100 / 100	100 / 100	100 / 100
Hope	**100 / 100**	**100 / 100**	**100 / 100**

- The shown non-star-free languages distinguish the models.
- All models score 100 on the table's three counter-language tasks.
- Hope also supports chunked training; the table provides no speed comparison.

Paper, Table 5 and Sec. 9.5, PDF pp. 38–39. Accuracies (%); exact Bin0/Bin1 length ranges are not specified in v1.

Ablations support several components, with a q tradeoff

Variant	LM PPL↓	Reasoning accuracy↑
Hope	12.24	58.1
Without DGD	13.41	56.5
Without momentum	13.58	56.9
Without CMS	13.04	57.3
Fixed v projection	13.90	55.1
Fixed q projection	**12.19**	57.4

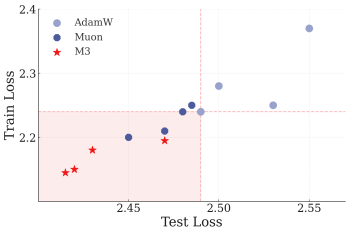
- Value-projection adaptation has the largest accuracy effect among these rows.
- Removing q adaptation slightly improves PPL but reduces accuracy.

Paper, Table 6, PDF p. 39. These averaged PPL values differ from individual datasets in Table 2. The final q configuration is inconsistent across the text and ablation.

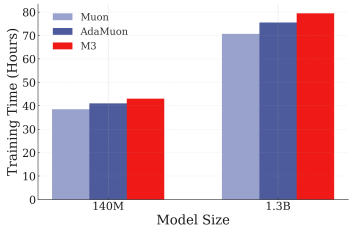
M3 extends gradient memory, at a runtime cost

$$\Delta\Theta_t \propto -\frac{\mathrm{NS}(M_t^{(1)}) + \alpha\,\mathrm{NS}(M_t^{(2)})}{\sqrt{V_t + \epsilon}}$$

24M ViT: lower left is better



Transformer training time: lower is better



Fast/slow momentum + orthogonalization + second-moment scaling.

M3 improves the displayed loss tradeoff but takes longer than Muon.

Paper, Algorithm 1, PDF p. 29; Figs. 11–12 and Sec. 9.7, pp. 39–40. Formula summarizes the mechanism. Timing compares optimizers, not Hope vs. Transformer.

The design opportunity is persistent, multi-rate learning

1. **Represent the whole learning system.**
   Weights, fast memories, and optimizer states all retain different histories.
2. **Design update rates and transfer paths together.**
   CMS supplies multiple time scales; self-generated targets make fast memory adaptive.
3. **Judge the complete tradeoff.**
   Reported gains coexist with recall gaps, adaptation costs, and unresolved implementation details in v1.

Finite-capacity compression still forgets. The paper explicitly says forgetting is not solved.

Paper, Sec. 10, PDF p. 40; assessment based on Tables 1–6 and the method analysis.

Paper and discussion material

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, Vahab Mirrokni.

Nested Learning: The Illusion of Deep Learning Architectures.
arXiv:2512.24695v1, submitted 31 December 2025.
`https://arxiv.org/abs/2512.24695v1`

- All experimental numbers and plots are from this version.
- Figures retain the authors' original axes and legends.
- Derivations and interpretive qualifications are identified where used.

The following slides provide backup detail for discussion.

Companion: reading-notes.zh.md. The PDF title page uses singular "Architecture"; the arXiv record uses plural "Architectures".

Backup: levels need an explicit knowledge-transfer path

Connection	Operation	Example
Conditioning	Read another state or its context	Fast-weight retrieval; attention
Backpropagation	Propagate a shared task signal through modules	Sequential CMS
Initialization	Learn the start state of a faster process	MAML; memory initialization
Generation	Produce weights or training signals for another process	Hypernetworks; network-generated gradients

For each level, specify the context, objective, parameter space, update schedule, and reset policy.

Paper, Sec. 3.3, PDF pp. 13–15; CMS variants, Sec. 7.1, p. 28.

Backup: a regression interpretation of Adam-like updates

A scalar version of the internal objective:

$$\min_m \sum_{i=1}^{t} (m g_i - p_t)^2 + \lambda m^2 \quad \Longrightarrow \quad m^* = \frac{p_t \sum_i g_i}{\sum_i g_i^2 + \lambda}.$$

Choose $p_t = \sqrt{\sum_i g_i^2}$ and let $\lambda \to 0$:

$$m^* = \frac{\sum_i g_i}{\sqrt{\sum_i g_i^2}}.$$

- This produces the first-/second-moment normalization structure.
- The printed cumulative updates omit standard Adam's EMA decay and bias correction.
- The optimum is for this constructed regression, not for the model's task loss.

Paper, Appendix B, Eqs. (101)–(105), PDF pp. 50–51. Scalar simplification and qualifications are this reading's analysis.

Backup: derive DGD from the stated proximal objective

With $\boldsymbol{u}$ fixed and $\eta > 0$, stationarity gives

$$(W^+\boldsymbol{x} - \boldsymbol{u})\boldsymbol{x}^\top + \eta^{-1}(W^+ - W_t) = 0.$$

Thus

$$W^+(I + \eta\boldsymbol{x}\boldsymbol{x}^\top) = W_t + \eta\boldsymbol{u}\boldsymbol{x}^\top.$$

Using $(I + \eta\boldsymbol{x}\boldsymbol{x}^\top)^{-1} = I - \frac{\eta\boldsymbol{x}\boldsymbol{x}^\top}{1+\eta\|\boldsymbol{x}\|^2}$,

$$W^+ = W_t - \frac{\eta}{1 + \eta\|\boldsymbol{x}\|^2}(W_t\boldsymbol{x} - \boldsymbol{u})\boldsymbol{x}^\top.$$

- Set $\boldsymbol{u} = -\nabla_{\boldsymbol{y}}\mathcal{L}$ to obtain the main-slide equation.
- If the task gradient is zero, DGD can still shrink $W_t\boldsymbol{x}$.

Re-derived from paper Eq. (56), PDF p. 20 / Eq. (113), p. 52. Appendix C changes signs and proximal coefficients in intermediate steps.

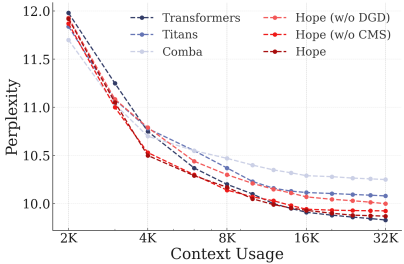
Backup: v1 leaves several implementation choices unclear

- **q projection:** p. 31 says it is non-adaptive, but the update set and Table 6 include q adaptation.
- **Chunk indexing:** Eq. (90) uses $C\lceil t/C \rceil$; the surrounding text requires the previous chunk's end state. A causal chunk-start index is $C\lfloor (t-1)/C \rfloor$.
- **M3:** Algorithm 1 prints additive accumulators, not ordinary EMA updates; the slow-state recurrence has inconsistent indices.
- **Muon interpretation:** the gradient in Eq. (44) contains a closeness term absent from Eq. (43)'s objective.

Paper, PDF pp. 17, 29, 31–33, 39. These are specification issues; no inference about unseen implementation or data leakage is made.

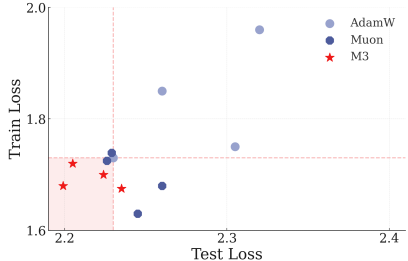
Backup: two additional empirical patterns

More context reduces PPL



Hope benefits from context up to 32K in the plotted experiment.

86M ViT shows a loss tradeoff



M3 has low test-loss points; Muon has points with lower train loss.

Paper, Figs. 10 and 11 (right panel), PDF p. 39. Curves and scatter points are unmodified author images.

Backup: preserve the distinction between three claims

Mathematical representation

Certain updates can be expressed as associative-memory optimization. Exact equalities require consistent objectives and conventions.

Design hypothesis

Multiple update rates and self-generated targets may improve adaptation and retention.

Empirical finding

Hope improves several reported tasks; exact recall, compute matching, and long-term retention remain consequential evaluation dimensions.

Discussion synthesis based on paper Secs. 3–10 and Appendices A–C, arXiv:2512.24695v1.