Palimpsa

Learning to Remember, Learn, and Forget in Attention-Based Models

Djohan Bonnet[1,2] Jamie Lohoff[1,2] Jan Finkbeiner[1,2]
Elidona Skhikerujah[2] Emre Neftci[1,2]

[1]Forschungszentrum Jülich (PGI-15) [2]RWTH Aachen | ICML 2026

From KV cache to a fixed-size state

Softmax self-attention

$$y_t = V_t \,\mathrm{Softmax}(K_t^\top q_t)$$

- Caches every (k, v) pair.
- Memory & compute grow with length. [1]

Linear attention — drop the softmax

$$S_t = S_{t-1} + v_t \otimes k_t\,, \qquad y_t = S_t\, q_t$$

- Fixed-size state $S \in \mathbb{R}^{d_v \times d_k}$.
- ”Hebbian” update in place. [2]

The catch: Non-orthogonal keys overwrite past memories — writing to S is now an **online continual learning** problem.

[1] Vaswani et al. Attention is all you need. NeurIPS, 2017.
[2] Schlag, Irie & Schmidhuber. Linear transformers are secretly fast weight programmers. ICML, 2021.

Bayesian inference over the state

Treat the state S not as a point estimate, but as a **probability distribution**.

Each token is projected to a data point $d_t = \{(k_t, \beta_t), v_t\}$.

We update the posterior by Bayes' rule:

$$p(S \mid d_{1:t}) \;\propto\; \underbrace{p(d_t \mid S)}_{\text{likelihood}}\; \underbrace{p(S \mid d_{1:t-1})}_{\text{prior}}$$

- States are Gaussian, $q_\theta(S) \sim \mathcal{N}(\mu, \Sigma)$ — variance encodes **uncertainty**.
- Output = posterior mean: $y_t = \mathbb{E}[S]\, q_t = \mu_t\, q_t$.

Controlled forgetting resolves the dilemma

Naive update → too plastic

Ignore uncertainty and overwrite the state: *catastrophic forgetting* — new keys wipe out old associations.

Plain Bayesian → too rigid

The posterior concentrates as t grows: *catastrophic remembering* — the state saturates and ignores new evidence.

Stay Bayesian, but forget [3]

Truncate the posterior to a window of N_t tokens — approximated online by down-weighting the accumulated prior:

$$p_w(S \mid d_{1:t}) \;\propto\; p(d_t \mid S)\, p_w(S \mid d_{1:t-1}) \left(\frac{p_w(S \mid d_{1:t-1})}{p(S)} \right)^{-\frac{1}{N_t}}$$

A natural **forgetting gate** $\alpha_t = 1 - 1/N_t$ tunes how fast the prior decays.

[3] Bonnet et al. Bayesian continual learning and forgetting in neural networks. Nature Communications, 2025.

Attention as variational inference

Casting the (weighted) posterior update as optimization, each state minimizes a variational free energy $\mathcal{F}_{t,i}$ per row i:

$$\mathcal{F}_{t,i}(\mu_i) = \underbrace{\frac{\beta_{t,i}}{2}\|\mu_i^\top k_t - v_{t,i}\|^2}_{\text{plasticity}} + \underbrace{(1-\alpha_t)\frac{\|\mu_i\|^2}{\sigma^2_{\text{prior}}}}_{\text{forgetting}} + \underbrace{\alpha_t(\mu_{t-1,i}-\mu_i)^\top \Sigma^{-1}_{t-1,i}(\mu_{t-1,i}-\mu_i)}_{\text{stability}}$$

- **Plasticity** — fit the new key–value pair.
- **Forgetting** — α_t discards stale data over window N_t.
- **Stability** — precision-weighted retention of the past.

Palimpsa: a dual-state, metaplastic update

Solving $\partial \mathcal{F}_{t,i} = 0$ with a diagonal covariance gives the closed-form **Palimpsa** update:

$$
\begin{aligned}
I_t &= \alpha_t I_{t-1} + (1 - \alpha_t) I_{\text{prior}} + \beta_t \otimes k_t^2 \\
\mu_t &= \alpha_t \frac{I_{t-1}}{I_t} \odot \mu_{t-1} + \frac{1}{I_t} \odot \left[(\beta_t \odot v_t) \otimes k_t \right] \\
I_t &= \frac{1}{\sigma_t^2}
\end{aligned}
$$

A **second state** I_t — the precision, or importance of each state — gives the learning rate of μ_t.

The variance is the learning rate of the synapse.

Mamba2 is a special case of Palimpsa

Strong forgetting $\Rightarrow$ posterior stays near the prior, $I_t \cong I_{\text{prior}}$, and the update collapses to:

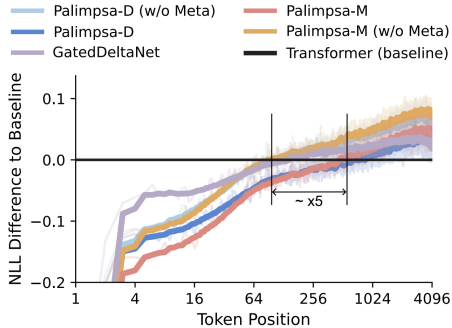
$$\mu_t = \alpha_t \mu_{t-1} + \frac{\beta \odot v_t}{I_{\text{prior}}} \otimes k_t \qquad \textbf{= the Mamba2 update [4].}$$

The continuum: Palimpsa $\leftrightarrow$ Mamba2

- One continuum, set by the importance I_t — so **any** Mamba2 can be upgraded.
- Reparameterize $v_t^\star = \beta_t \odot v_t$: then $\beta_t \to 0$ recovers $I_t \cong I_{\text{prior}}$, i.e. plain Mamba2.
- Initialize β_t small, then fine-tune — β_t grows only where metaplasticity helps.

[4] Dao & Gu. Transformers are SSMs: generalized models and efficient algorithms through structured state space duality. arXiv:2405.21060, 2024.

Metaplasticity expands memory capacity



Per-token NLL vs. a Transformer baseline (760M).
Metaplastic variants degrade **far more slowly** — a memory advantage $\sim 5\times$ longer.

Takeaways

- ICL is a **continual learning** problem in a fixed-size state.
- **Palimpsa:** Metaplasticity + Bayesian forgetting — a second state gives the learning rate.
- **Mamba2** is a static-learning-rate special case — can be fine-tuned to be metaplastic.
- **It works:** Upgrading just 8 layers of a 2.7B Mamba2 lifts long-context performance.

Code & Paper

github.com/djo1996/Palimpsa

Read the paper for all the experiments —
MQAR, Commonsense Reasoning, LongBench, RULER.