The Clocks of Composition: Built by the Signal, Lost to Weight Decay, Kept by Rehearsal

Anonymous authors
Paper under double-blind review

Abstract

A network can learn a skill it will not keep. In a looped network, a signal on the intermediate state builds two-hop composition over facts seen only alone; when the signal stops, with the training data unchanged, the composition fades, and plain training does not build it (OOD 0.42). We measure the three clocks that govern its life. It is *built* on the signal's clock: at dense signal rates the time to compose barely depends on weight decay. It is *lost* on the weight-decay clock: the time to lose it scales as $(\eta\lambda)^{-1.04}$ across a tenfold range of learning rates and a 33-fold range of weight decays, and without weight decay no seed loses it. It is *kept* only by rehearsal above a threshold; with a short second-moment memory, which removes the damage that rare rehearsal steps do under Adam's default, the threshold rises as $\lambda^{2.2}$. The law travels: registered in advance, it predicts the rehearsal threshold of a transformer's fact memory, a different architecture, task and quantity, with no parameter fitted (exponent 2.3). In a transformer with chain-of-thought, composition fails at high weight decay because no training row exercises the out-of-distribution facts where chain-of-thought reads them; presenting every fact once in that position, with no knowledge of the split, restores it from 0.38 to 0.95.

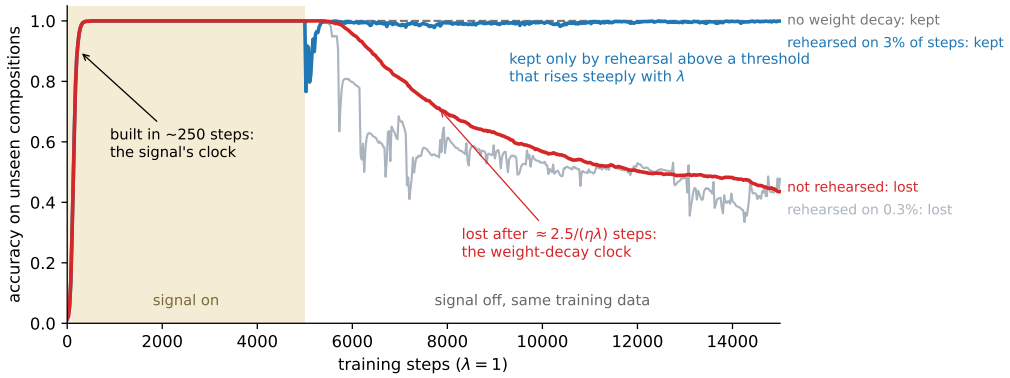

Figure 1: A composition the training loss does not exercise, over its life (looped toy, $\lambda = 1$, medians over seeds). A signal on the intermediate state builds it within a few hundred steps. When the signal stops, with the training data unchanged, weight decay erases it; rehearsing the signal on 3% of steps keeps it, on 0.3% does not; without weight decay nothing removes it. Runs `wd_erosion`, `wd_rehearsal`.

1 Introduction

Give a looped network a signal on its intermediate state and within a few hundred steps it composes facts it has only ever seen alone; take the signal away, keep the training data exactly the same, and the composition fades (Figure 1). Plain training on that data does not build it (out-of-distribution

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

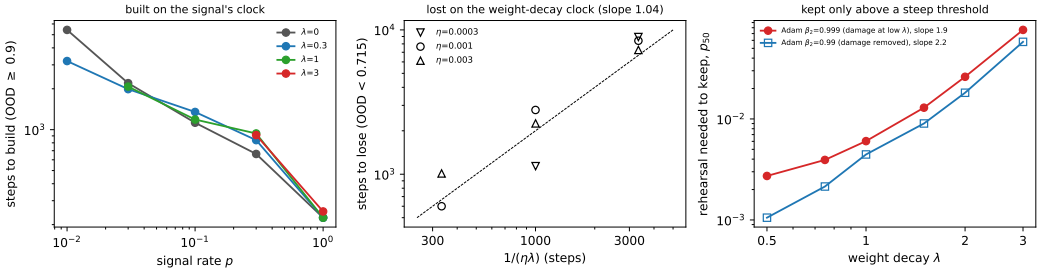

Figure 2: The three clocks, measured (looped toy). Left: with a signal on the intermediate state on a fraction p of steps, the time to build composition falls with p and barely depends on λ (cells where every seed composes). Middle: once the signal stops, the time to lose it scales as $1/(\eta\lambda)$ across three learning rates (dashed: $\eta\lambda t = 2$). Right: the rehearsal rate needed to keep it, on networks forked from one installation. With Adam's default $\beta_2 = 0.999$, rare rehearsal damages the network and the threshold flattens at low λ; with $\beta_2 = 0.99$ from the fork there is no damage and it rises as $\lambda^{2.2}$ (p_{50}: rate at which median OOD after 10k steps crosses 0.715). Runs `wd_acq`, `wd_erosion`, `wd_eta`, `wd_fork`, `wd_fork99`.

accuracy 0.42). The composition is held by the optimiser rather than by the task (Wang & Aitchison, 2024), and its life runs on three clocks the task does not set: it is *built* on the clock of the signal, *lost* on the weight-decay clock $1/(\eta\lambda)$, and *kept* only by rehearsal above a threshold that rises steeply with λ (Figure 2).

The same pattern shapes composition in transformers. A transformer trained on facts it learned separately memorises the question–answer pairs it was shown and composes out of distribution, if at all, much later (Wang et al., 2024; Zhang et al., 2026), a delay usually credited to weight decay slowly favouring the cheaper, structured solution (Varma et al., 2023; Lyu et al., 2023; Kim, 2026). The clocks say what that solution costs to build and to keep, and where the cost falls: on the parts of the path that no training example exercises.

These clocks set the terms for the goal behind this work, a training algorithm that composes early: where the task loss does not supply the signal that builds composition, the algorithm has to, and keep supplying it at a rate that weight decay and the optimiser set. The transformer shows where this bites in practice: at high weight decay, composition fails exactly where no training row exercises the facts it reads.

Contributions.

1. **Loss:** once its signal stops, composition is lost on the $1/(\eta\lambda)$ clock, with crossing time $\propto (\eta\lambda)^{-1.04}$, and not within 10^4 steps without weight decay (Section 4).
2. **Gain:** it is built on the clock of the signal, nearly independently of λ while the signal is dense (Section 5).
3. **Keep:** rehearsal must exceed a threshold that rises steeply with λ, and the toy's law, registered in advance, predicts the threshold of a transformer's fact memory with nothing fitted (exponent 2.3); under Adam's default moments rare rehearsal damages the network and the threshold bends, and with a shorter second-moment memory it rises as $\lambda^{2.2}$ (Section 6).
4. **Where it bites:** in a transformer at high weight decay, presenting every fact once where chain-of-thought reads it raises composition from 0.38 to 0.95 (Section 7).

2 Why clocks

Under AdamW, $\theta_{t+1} = (1 - \eta\lambda)\theta_t - \eta u_t$ with $u_t = \hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon)$ (Loshchilov & Hutter, 2019), so the weights are an exponential moving average of the updates with timescale $1/(\eta\lambda)$ (Wang & Aitchison, 2024). Whatever the updates stop reaching therefore decays on that clock, and whatever they reach is built at the rate they reach it. Adam adds one twist, inherited from Adagrad (Duchi et al., 2011; Kingma & Ba, 2014): it normalises by a running second moment, so a signal that arrives rarely

is paid more per arrival, and how much more depends on the optimiser's memory. The sections below measure what these three facts imply for a composition the training data does not exercise: how fast it is built, how fast it is lost, and how often it must be rehearsed to be kept.

3 Setting

The looped toy. Sixty entities and eight relations define a random fact table tail(e, r); 70% of facts are in-distribution (ID). One block $f(x, r)$ (a two-hidden-layer MLP on $[x; R_r]$, width 256, $d = 64$) is applied once for an atomic query (e, r) and twice for a two-hop query (e_0, r_0, r_1), reading out by similarity to the learned entity table E. Training covers every atomic fact and 80% of the two-hop queries whose facts are both ID. The out-of-distribution (OOD) test asks two-hop queries whose *second* fact was seen only atomically: answering them requires the second application to read the first application's output as if it were an entity. AdamW, $\eta = 10^{-3}$, full batch. The *format signal* $\|h_1 - \text{sg}(E[\text{bridge}])\|^2$ pulls the first application's output to the true bridge entity, on the training queries only: no held-out query, and no intermediate supervision for one, is ever trained on. It installs composition (OOD 1.00) within a few hundred steps; plain training plateaus at median OOD 0.42 (12 seeds, $\lambda = 0.1$).

The looped transformer. Wang et al.'s knowledge-graph two-hop task (Wang et al., 2024), with about 12.6 training two-hop rows per ID atomic fact, and an 8-layer pre-LN transformer, $d = 768$, layers i and $i{+}4$ tied, AdamW (lr 10^{-4}, batch 512). OOD strata separate a first-hop fact seen only atomically (hop-1 OOD) from a second-hop one (hop-2 OOD). Composition is scored by greedy decoding: with chain-of-thought the model writes the bridge itself, and a query counts only if both the bridge and the answer are right.

Each experiment's predictions and kill criteria were registered before its results were read. Appendix A summarises the registered predictions for the experiments reported here and their outcomes, including the failures.

4 Loss: composition decays on the weight-decay clock

We install composition with the format signal for 5,000 steps and then switch it off, keeping the training data unchanged, at $\lambda \in \{0, 0.1, 0.3, 1, 3\}$ with 12 seeds each (Figure 3). The *crossing time* $t_\times$ is the number of steps after switch-off until OOD accuracy falls below 0.715, halfway between the installed 1.00 and the plateau of plain training (0.43).

- **No weight decay, no erosion.** At $\lambda = 0$, OOD stays at 1.00 in 12 of 12 seeds for the whole 10^4-step continuation.
- **With weight decay, one clock.** In steps, the crossing time varies fourteenfold: 600 at $\lambda = 3$, 2,788 at $\lambda = 1$ and 8,412 at $\lambda = 0.3$. In units of $1/(\eta\lambda)$ it is nearly constant: $\eta\lambda\, t_\times = 1.8$, 2.8 and 2.5. The rescaled curves coincide early (Figure 3, right). Read as exponential decay, $\eta\lambda\, t_\times \approx 2.5$ means composition fails after the unexercised part has decayed by about $e^{-2.5}$, a similar factor at every λ.
- **The clock is $\eta\lambda$, not λ.** Repeating the sweep at $\eta = 3 \cdot 10^{-4}$ and $3 \cdot 10^{-3}$ at matched $\eta\lambda$ (Table 1), across eight cells spanning $10\times$ in η and $33\times$ in λ, the crossing time scales as $(\eta\lambda)^{-1.04}$ (Figure 2, middle). At fixed $\lambda = 1$, a tenfold larger η shortens it from 8,925 to 1,012 steps.
- **The block is not idle.** The block that composes is trained on every step by the atomic and ID queries; only its OOD use goes unexercised, and that use decays on the $1/(\eta\lambda)$ clock. The weight norm falls only modestly after switch-off ($50 \to 43$, $41 \to 34$, $31 \to 25$ at $\lambda = 0.3, 1, 3$).

5 Gain: composition is learned on the signal's clock

If the pathway changes only through signals that reach the intermediate state, acquisition should run on the *signal's* clock. We turn the format signal on for a random fraction p of steps from the start, with no installation phase, at $\lambda \in \{0, 0.3, 1, 3\}$ (Table 2, Figure 2 left; 4 seeds).

Table 1: The dimensionless crossing time $\eta\lambda\, t_\times$ (median; seeds that crossed / seeds) at matched $\eta\lambda$ and three learning rates. At $\eta = 3 \cdot 10^{-4}$, $\lambda = 10$ the format signal never installed composition, so that cell is empty. Source: `tables/eta.tex`, runs `wd_erosion`, `wd_eta`.

$\eta\lambda$	$\eta = 3 \cdot 10^{-4}$	10^{-3}	$3 \cdot 10^{-3}$
0.0003	2.68 (6/6)	2.52 (8/12)	2.18 (6/6)
0.001	1.14 (6/6)	2.79 (12/12)	2.25 (6/6)
0.003	not installed	1.80 (12/12)	3.04 (6/6)

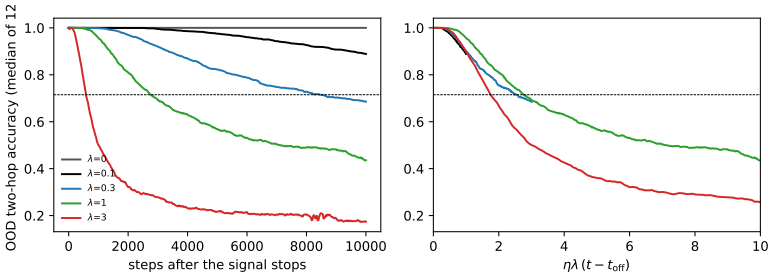

Figure 3: Erosion after the format signal stops. Left: OOD accuracy in steps (median over 12 seeds; dashed: the crossing level 0.715). Right: the same curves in units of $1/(\eta\lambda)$. The crossing time against $1/(\eta\lambda)$ at three learning rates is Figure 2, middle. Runs `wd_erosion`.

- **Weight decay barely matters.** At $p = 1$, composition arrives at 225–250 steps for every λ, *including* $\lambda = 0$. At $p = 0.3$ the four median times differ by $1.42\times$.
- **The signal sets the time.** Where every seed composes, the median time falls as a power of p, with log–log slopes -0.66, -0.54 and -0.60 at $\lambda = 0$, 0.3 and 1.
- **The clocks compete.** When the signal is sparse and decay strong, decay wins: at $\lambda = 3$, no seed composes within 8,000 steps for $p \le 0.03$ and two of four do at $p = 0.1$.

Composition is thus learned on one clock and lost on another: the signal sets the rate of gain and $\eta\lambda$ the rate of loss.

Table 2: Median steps to OOD ≥ 0.9 over the seeds that reach it (count in parentheses), with the signal on a fraction p of steps from the start. Source: `tables/acquisition.tex`, runs `wd_acq`.

λ	$p = 0.01$	0.03	0.1	0.3	1
0	5425 (4/4)	2200 (4/4)	1125 (4/4)	662 (4/4)	225 (4/4)
0.3	3200 (4/4)	1988 (4/4)	1350 (4/4)	838 (4/4)	225 (4/4)
1	7750 (1/4)	2075 (4/4)	1188 (4/4)	938 (4/4)	225 (4/4)
3	– (0/4)	– (0/4)	1775 (2/4)	912 (4/4)	250 (4/4)

6 KEEP: REHEARSAL MUST EXCEED A STEEP THRESHOLD

How often must the signal recur to keep composition? After installation we keep the format signal on a random fraction p of the remaining steps and read the final OOD after a 10k-step continuation; p_{50} is the rate at which it crosses 0.715. A sharp threshold appears (Table 3): at $\lambda = 1$, rehearsal on 0.3% of steps leaves OOD at 0.48, 1% keeps 0.84 and 3% keeps 1.00. The threshold is $p_{50} = 0.0066$ at $\lambda = 1$ and 0.066 at $\lambda = 3$, an exponent of 2.1 that rejects the $p^* \propto \lambda$ we had registered.

On shared networks the threshold bends. To measure it at more weight decays without installation noise, we forked one installed network per seed into six weight decays (11 rehearsal rates, 4 seeds; Figure 2, right, red). The threshold rises 28-fold, from $p_{50} = 0.0027$ at $\lambda = 0.5$ to 0.076 at $\lambda = 3$, but its log–log slope is 1.13 over the lower three weight decays and 2.56 over the upper three. Rehearsing

Table 3: Final OOD (median of 6 seeds) when the signal recurs on a fraction p of steps after installation ($p = 0$: the erosion runs, 12 seeds). Source: `tables/rehearsal.tex`, runs `wd_rehearsal`.

λ	$p = 0$	0.003	0.01	0.03	0.1	0.3
0.3	0.69	0.92	1.00	1.00	1.00	1.00
1	0.44	0.48	0.84	1.00	1.00	1.00
3	0.17	0.27	0.32	0.46	0.85	1.00

toward the entity table frozen at installation gives a similar curve (slope 1.72 against 1.88), and extending installation and continuation fourfold moves the $\lambda = 1$ threshold by $1.09\times$.

Shortening Adam's memory removes the damage and most of the bend. That a long second-moment memory harms learning after a change in the data, and that resetting it helps, is known in reinforcement learning (Asadi et al., 2023); what we add is its effect on the rehearsal threshold. Adam's second-moment estimate satisfies $v_t \geq (1 - \beta_2)g_t^2$, so the memory β_2 bounds how large a step a rare gradient can produce. With $\beta_2 = 0.999$ rare rehearsal transiently knocks out facts the network already knows: 51 of the 264 rehearsed forks lose more than a tenth of their atomic facts, and the damage sits where the curve is shallow: just below each threshold, median minimum atomic accuracy is 0.69–0.78 at $\lambda \leq 1$ and 0.93–1.00 at $\lambda \geq 1.5$. Shortening the memory to $\beta_2 = 0.99$ at the fork removes the damage (0 of 264 forks, lowest 0.97) and most of the bend (Figure 2, right, blue): the threshold falls $2.59\times$, $1.83\times$ and $1.36\times$ at $\lambda = 0.5$, 0.75 and 1, stays within $1.5\times$ above, and now rises as $\lambda^{2.21}$ (half-slopes 2.06 and 2.70). The change of memory is not specific to rehearsal (without rehearsal it also lowers final OOD, from 0.45 to 0.28 at $\lambda = 1$), so this ties the bend to the second-moment memory rather than isolating damage as its cause.

The optimiser sets the price. Adam's moments move the threshold beyond what a one-coordinate account of its recurrence predicts: without momentum and with a short memory, $(\beta_1, \beta_2) = (0, 0.9)$, keeping composition at $\lambda = 1$ takes $3\times$ less rehearsal ($p_{50} = 0.0021$ against 0.0060; Appendix B). Under SGD, rare rehearsal of an Adam-trained network collapses it altogether (Appendix C): whether and how cheaply a skill can be kept is a property of the optimiser, not only of the task.

The law travels to a transformer. Before running it, we registered that the toy's threshold would carry over to a different architecture, task and quantity, with an exponent in $[1.4, 2.8]$: a 4-layer transformer memorising 40,000 facts in full batch, in which a fixed 10% stop being trained at a fork and are rehearsed on a fraction p of steps. Changing the objective abruptly collapses this model's memory under Adam; lowering the learning rate at the fork prevents it (Appendix A). The rehearsal needed to keep half the facts is $p_{50} = 0.0007$ at $\lambda = 0.3$ and 0.0119 at $\lambda = 1$ (Table 4, Figure 4): an exponent of 2.34, inside the predicted band, on the same line as the toy. A law measured in a looped MLP on composition sets, in advance, how much rehearsal a transformer needs to keep its facts.

Table 4: Transformer fact memory: mean final accuracy on the rehearsed facts (2 seeds) when they recur on a fraction p of steps after the fork, and the rehearsal rate p_{50} that keeps half. Source: `tables/t3p4.tex`, runs `tlr`, `summary_T3P4.json`.

λ	$p = 0.0003$	$p = 0.001$	$p = 0.003$	$p = 0.01$	$p = 0.03$	p_{50}
0.3	0.34	0.56	0.14	1.00	–	0.0007
1	–	0.15	0.40	0.42	0.92	0.0119

7 Exposure accounting in a transformer

We count exposures in the looped transformer from the training set alone. An atomic fact is exercised by its own atomic row and by every multi-hop training row that uses it as a first or second hop. An ID fact is used by 25.2 multi-hop rows on average (median 25). The 2,029 facts that no multi-hop row uses, which in Wang et al.'s split are exactly the OOD atomic facts, are exercised by one row: $26\times$ less.

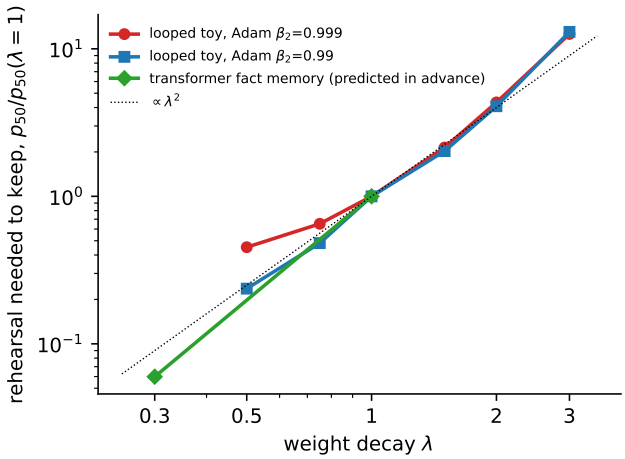

Figure 4: One law, two systems. The rehearsal needed to keep, divided by its value at $\lambda = 1$, in the looped toy (with Adam's default β_2 rehearsal damage bends it at low λ; with $\beta_2 = 0.99$ it follows a single power) and in a transformer's fact memory, predicted before its runs. Dotted: λ^2. Runs `wd_fork`, `wd_fork99`, `tlr`.

At high weight decay, composition is lost. With chain-of-thought (CoT) the model writes the bridge entity and then answers. At $\lambda = 0.1$ second-hop OOD holds at 0.98 (Table 5, Figure 5). At $\lambda = 1$ the same model composes early and then loses it: second-hop OOD peaks at 0.73–0.76 at 5k steps and ends at 0.37–0.38, with its underexposed facts at 0.84 (means of the last four evaluations, since the training loss spikes transiently in at least one run of every $\lambda = 1$ arm; `tables/criteria.json`).

At high weight decay, exercising each fact where it is read restores composition. For an OOD query, CoT composition reads the underexposed fact at the bridge position, where the second step looks. An ID fact is exercised there by every CoT row that uses it as a second hop; an OOD fact never is. So the missing exercise is positional, and the fix needs no knowledge of which facts are OOD:

- *Every fact, where it is read*: adding one training row per atomic fact in the CoT layout, $[\texttt{BOS}, \cdot, \cdot, r, e, t]$, with no first hop and no composed answer, lifts second-hop OOD from 0.38 to 0.95.
- *The facts, where they are not read*: repeating each underexposed fact 13 times in its usual atomic row brings those facts to 1.00 by the end, but composition reaches only 0.59.
- Adding targeted repetition to the layout rows gives 0.98 [0.97, 0.99], and adding as many rows of random, already exposed facts gives 0.94: once facts are exercised where they are read, which facts are repeated matters little.

The layout rows are a close relative of the identity bridge (Lin et al., 2025) and of training upper layers on OOD atomic facts (Zhang et al., 2026, Section 5.1, Appendices F and H); what the clocks add is when they are needed: at $\lambda = 0.1$ composition holds without them.

The latent reader. Without CoT no step writes the bridge, and second-hop OOD composition is 0.04 ($\lambda = 0.3$) and 0.34 ($\lambda = 1$) after 200k steps (Table 5): nothing in the data exercises those facts where the loop's second pass reads them. That is the gap an early-composing training algorithm has to close.

8 Related work

Weight decay and grokking. Circuit efficiency explains grokking as a competition between a memorising and a generalising circuit that weight decay resolves toward the cheaper one, and reports *ungrokking* when a reduced dataset makes memorising cheaper (Varma et al., 2023); it reports endpoints and trajectories, not the scaling of the erosion time with η and λ. Late-phase implicit bias induces

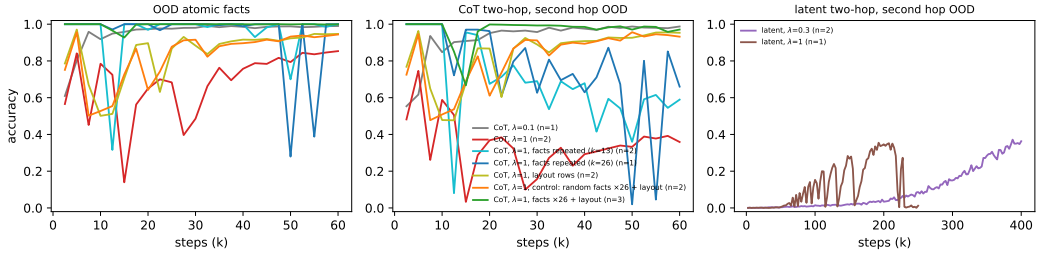

Figure 5: Exposure accounting in the looped transformer (Wang et al.'s knowledge graph). Left: OOD atomic facts, with 26× fewer training-row exposures than ID facts, at $\lambda = 1$ under repetition and CoT-layout rows. Middle: CoT second-hop OOD composition in the same arms. Right: latent composition.

Table 5: Mean of the last four evaluations per run (latent arms: the four ending at 200k), then mean [min, max] over seeds. Source: `tables/transformer.tex`, runs `tclock`, `fpay`, `ppay`, `ctrl`.

Arm	seeds	OOD atomic	second-hop OOD
CoT, λ=0.1	1	0.99 [0.99, 0.99]	0.98 [0.98, 0.98]
CoT, λ=1	2	0.84 [0.84, 0.85]	0.38 [0.37, 0.38]
CoT, λ=1, facts repeated (k=13)	2	1.00 [0.99, 1.00]	0.59 [0.56, 0.61]
CoT, λ=1, facts repeated (k=26)	1	0.85 [0.85, 0.85]	0.59 [0.59, 0.59]
CoT, λ=1, layout rows	2	0.94 [0.94, 0.95]	0.95 [0.95, 0.95]
CoT, λ=1, control: random facts ×26 + layout	2	0.94 [0.94, 0.94]	0.94 [0.93, 0.94]
CoT, λ=1, facts ×26 + layout	3	1.00 [1.00, 1.00]	0.98 [0.97, 0.99]
latent, λ=0.3, at 200k	2	0.95 [0.95, 0.95]	0.04 [0.03, 0.05]
latent, λ=1, at 200k	1	0.98 [0.98, 0.98]	0.34 [0.34, 0.34]

grokking with delay $\propto \log(\alpha)/\lambda$ (Lyu et al., 2023), and an exactly solvable model derives the decoupled $1-\eta\lambda$ spectrum of null coordinates and an inverse-$\eta\lambda$ arrival time (Kim, 2026, Prop. 6); neither measures erosion. AdamW's weights are an exponential moving average of updates with timescale $1/(\eta\lambda)$ (Wang & Aitchison, 2024), contaminated examples are forgotten faster under larger weight decay (Bordt et al., 2024), and weight decay has been used as a controlled forgetting mechanism (Ramesh et al., 2026). Replay of around 1% prevents forgetting empirically (de Masson d'Autume et al., 2019; Bethune et al., 2025), and outperforms weight decay as a remedy (Bethune et al., 2025), and facts seen in pretraining are forgotten on a power law (Chang et al., 2024). No work we found measures the required rehearsal rate as a function of λ.

Adam and rare signals. Adagrad and Adam give rare features larger steps (Duchi et al., 2011); Adam's update separates a sign from a relative-variance magnitude (Balles & Hennig, 2018); Adam beats gradient descent under heavy-tailed class imbalance because gradient descent is slow on rare classes (Kunstner et al., 2024); weight decay drives AdamW's norms and rotation to an equilibrium (Kosson et al., 2024); adaptive optimisers have late instabilities of their own (Thilak et al., 2022); and resetting Adam's second moment counters the contamination of its long memory when the data change (Asadi et al., 2023). Excluding embeddings from weight decay is documented practice (Team OLMo et al., 2024). None of these measures the rehearsal rate needed to keep a rarely paid component.

Composition. Transformers compose implicitly only after long training and mostly in-distribution (Wang et al., 2024). Parameter sharing across layers fixes out-of-distribution two-hop (Zhang et al., 2026); realigning the loop's intermediate state to the decoded bridge embedding helps (Fu et al., 2026), as does supervising an identity mapping on bridge entities (Lin et al., 2025). Grokked circuits transfer poorly when new knowledge is integrated (He et al., 2026). Chain-of-thought makes such tasks learnable and can be internalised (Huang et al., 2026). Uniform skill frequencies can slow implicit composition relative to a power law (Wang et al., 2026); we equalise the exposure of individual

facts, a different intervention. Lyle et al. (2025) connect grokking's feature-learning dynamics to overwriting in continual learning.

9 Scope and what is borrowed

The clocks are measured in a small, full-batch looped toy, where weight decay is the dominant force on what the loss leaves unexercised; under fine-tuning of a pretrained LM, interference from the continued data acts faster (reversal is lost even at $\lambda = 0$), and in a full-batch transformer fact memory an abrupt change of the objective collapses the whole memory under Adam unless the learning rate is lowered at the fork (Appendix A). The toy's signal uses the true bridge: it measures the clocks rather than proposing a method, and the $\lambda^{2.2}$ of the threshold is a measured shape at a 10k-step horizon. That weight decay removes a costlier circuit is ungrokking (Varma et al., 2023); that AdamW forgets on a $1/(\eta\lambda)$ timescale is the EMA view (Wang & Aitchison, 2024); that Adam favours rare signals is Adagrad's property (Duchi et al., 2011). What is new is the measured structure: erosion collapsing in $\eta\lambda t$, acquisition independent of λ under dense signals, the steep rehearsal threshold and its dependence on the optimiser, and, in the transformer, that composition needs each fact exercised where it is read.

10 Conclusion

In our looped toy, composition that the training data does not exercise lives on three clocks the task does not set: it is built as fast as a signal reaches the intermediate state, lost on the weight-decay clock when that signal stops, and kept only by rehearsal above a threshold that rises steeply with weight decay, a law that, registered in advance, predicts the threshold of a transformer's fact memory. In a transformer the same account says where composition breaks: at the position where no training row exercises the facts it reads. For a training algorithm that composes early, this sets the terms in this regime: supply the signal the task loss does not, keep supplying it above that rate, and, under Adam, deliver it so that each rehearsal does not damage what the network already knows.

Reproducibility statement

Every number in the paper is generated from run artifacts: `src/toys/wdclock_figs.py` (toy tables and figures), `src/toys/wd_erosion.py` and `src/toys/wd_rehearsal.py` (toy analyses), `src/toys/adam_payment.py`, `src/toys/adam_predict.py` and `src/toys/wd_paired.py` (the one-coordinate model, its frozen predictions and the held-out test), and `src/factloc/payment_figs.py` and `src/factloc/dose.py` (transformer tables, figures, payment counts and dose). All pre-registrations, with their timestamps and deviations, are in `PREREG.md`. Code, pre-registrations and run logs will be released. Toy runs are CPU-only; each transformer run uses a single GPU.

Use of large language models

Large language models assisted in writing the code and the paper, in the literature review, and in running and analysing the experiments. The authors checked every reported number against the generated tables and run logs and take full responsibility for the content.

References

Kavosh Asadi, Rasool Fakoor, and Shoham Sabach. Resetting the optimizer in deep RL: An empirical study. In *Advances in Neural Information Processing Systems*, 2023.

Lukas Balles and Philipp Hennig. Dissecting Adam: The sign, magnitude and variance of stochastic gradients. In *International Conference on Machine Learning*, 2018.

Louis Bethune, David Grangier, Dan Busbridge, Eleonora Gualdoni, Marco Cuturi, and Pierre Ablin. Scaling laws for forgetting during finetuning with pretraining data injection. In *International Conference on Machine Learning*, 2025. arXiv:2502.06042.

Sebastian Bordt, Suraj Srinivas, Valentyn Boreiko, and Ulrike von Luxburg. How much can we forget about data contamination? *arXiv preprint arXiv:2410.03249*, 2024.

Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. How do large language models acquire factual knowledge during pretraining? In *Advances in Neural Information Processing Systems*, 2024.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems*, 2019.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

Hengyu Fu, Tianyu Guo, Zixuan Wang, Hanlin Zhu, Jason D. Lee, Jiantao Jiao, Stuart Russell, and Song Mei. DiscoLoop: Looping discrete embeddings and continuous hidden states for multi-hop reasoning. *arXiv preprint arXiv:2607.00341*, 2026.

Kaiyu He, Zhang Mian, Peilin Wu, Xinya Du, and Zhiyu Chen. Is grokking worthwhile? functional analysis and transferability of generalization circuits in transformers. *arXiv preprint arXiv:2601.09049*, 2026.

Yixiao Huang, Hanlin Zhu, Zixuan Wang, Jiantao Jiao, Stuart Russell, Somayeh Sojoudi, and Song Mei. Transformers provably learn to internalize chain-of-thought. *arXiv preprint arXiv:2605.28600*, 2026.

Taeyoung Kim. Grokking on the weight-decay clock: A rate hierarchy from softly broken symmetries. *arXiv preprint arXiv:2607.23967*, 2026.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Atli Kosson, Bettina Messmer, and Martin Jaggi. Rotational equilibrium: How weight decay balances learning across neural networks. In *International Conference on Machine Learning*, 2024.

Frederik Kunstner, Robin Yadav, Alan Milligan, Mark Schmidt, and Alberto Bietti. Heavy-tailed class imbalance and why Adam outperforms gradient descent on language models. In *Advances in Neural Information Processing Systems*, 2024.

Pengxiao Lin, Zheng-An Chen, and Zhi-Qin John Xu. Unveiling the mechanisms of multi-hop reasoning in transformers via identity bridge. *arXiv preprint arXiv:2509.24653*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Clare Lyle, Gharda Sokar, Razvan Pascanu, and Andras Gyorgy. What can grokking teach us about learning under nonstationarity? *arXiv preprint arXiv:2507.20057*, 2025.

Kaifeng Lyu, Jikai Jin, Zhiyuan Li, Simon S. Du, Jason D. Lee, and Wei Hu. Dichotomy of early and late phase implicit biases can provably induce grokking. *arXiv preprint arXiv:2311.18817*, 2023.

Aditya A. Ramesh, Alex Lewandowski, and Jürgen Schmidhuber. Learning to forget: Continual learning with adaptive weight decay. *arXiv preprint arXiv:2604.27063*, 2026.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, et al. 2 OLMo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

Vimal Thilak, Etai Littwin, Shuangfei Zhai, Omid Saremi, Roni Paiss, and Joshua Susskind. The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon. *arXiv preprint arXiv:2206.04817*, 2022.

Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. Explaining grokking through circuit efficiency. *arXiv preprint arXiv:2309.02390*, 2023.

Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokked transformers are implicit reasoners: A mechanistic journey to the edge of generalization. In *Advances in Neural Information Processing Systems*, 2024.

Xi Wang and Laurence Aitchison. How to set AdamW's weight decay as you scale model and dataset size. *arXiv preprint arXiv:2405.13698*, 2024.

Zixuan Wang, Xingyu Dang, Jason D. Lee, and Kaifeng Lyu. The power of power law: Asymmetry enables compositional reasoning. *arXiv preprint arXiv:2604.22951*, 2026.

Zili Zhang, Yilin Wang, Heng Wang, Herun Wan, and Minnan Luo. Why knowing both hops is not enough: Understanding two-hop generalization in language models. *arXiv preprint arXiv:2608.07261*, 2026.

A Pre-registered predictions and outcomes

Each registered experiment's design, predictions and kill criterion were written before its runs were read; some were written after runs began (`PREREG.md`, released with the code, with deviations). Its section times were typed by hand and some are about an hour late; the order of the file, the checksum of the frozen held-out predictions and the file-system times of the predictions and results are the chronology. Each prediction and its outcome, including the failures:

- **Erosion** (toy): $\lambda = 0$ holds; $\eta\lambda t_\times$ agrees across λ within $2\times$. *Passed* (12/12; spread $1.55\times$).
- **The clock is** $\eta\lambda$ (toy, η varied at matched $\eta\lambda$): $\eta\lambda t_\times$ agrees within $2\times$ across η; kill beyond $3\times$, or agreement at matched λ instead. *Passed* at $\eta\lambda = 3 \cdot 10^{-4}$ ($1.23\times$); *missed* at 10^{-3} ($2.45\times$, inside the kill bound); at $3 \cdot 10^{-3}$ only two cells are comparable ($1.69\times$), because one ($\eta = 3 \cdot 10^{-4}$, $\lambda = 10$) never installed composition, an exclusion the registration did not anticipate.
- **Acquisition** (toy): slope in $[-1.5, -0.5]$ and $< 2\times$ spread across λ at fixed p. Kill: λ moves the median time by more than $3\times$ at fixed p. *Passed for dense signals* ($p \geq 0.3$); *killed* at $p = 0.1$, where the all-seed median at $\lambda = 3$ exceeds 4,887 against 1,125 at $\lambda = 0$, and at $p = 0.03$ (above 8,000 against 1,988); at $p = 0.01$ the comparison is censored.
- **Maintenance** (toy): $p^* \propto \lambda$. *Rejected* (exponent 2.1 from two resolved thresholds). The $\sqrt{p}$ reading came after this result, and its SGD test was pre-registered before it was run.
- **Rehearsal at** $\lambda = 0$ is harmless. *Rejected*: rare rehearsal perturbs.
- **Threshold on shared forks** (6 weight decays per optimiser, current and frozen target, 40k-step horizon): Adam slope in $[1.6, 2.6]$ with no bend; SGD slope in $[0.7, 1.3]$ where the task stays intact; the frozen target removes the $\lambda = 0$ harm; horizon within $1.5\times$. *Bend: rejected* (half-slopes 1.13 and 2.56); *SGD: rejected* (rare rehearsal collapses the network); *frozen target: slope unchanged, harm not removed*; *horizon: passed* ($1.09\times$, $1.26\times$; that protocol also lengthens installation).
- **Removing the damage** (the same forks with $\beta_2 = 0.99$ from the fork; hypothesis formed from the previous item): half-slopes differ by < 0.6; thresholds at $\lambda \leq 1$ fall by $\geq 1.3\times$ on average and stay within $1.5\times$ at $\lambda \geq 1.5$; no rehearsed run below 0.9 atomic accuracy. Kill: half-slopes differ by > 1, or low-λ thresholds do not fall. *Bend: narrowly missed* (2.06 and 2.70, difference 0.64); *thresholds: passed* (mean $1.93\times$ lower at $\lambda \leq 1$; within $1.44\times$ above); *damage: passed* (0 of 264).
- **Optimiser control** (SGD with momentum): exponent ≤ 1.5, and no $\lambda = 0$ shock. *Exponent passed* (1.00) in that sweep, superseded by the shared-fork test above (collapse); *shock rejected* (0.84–0.92).
- **Sign flip** (toy): more two-hop data per entity makes composition the cheaper fit and stops erosion. *Rejected*: erosion weakens (change in OOD $-0.74 \to -0.35$ as n_{rel} goes from 4 to 16; kills ledger) but never disappears.
- **Transformer onset** $\propto 1/(\eta\lambda)$ for latent composition. *Withdrawn* after early evaluations showed that it conflates fact acquisition with reader onset.

- **Fact payment** (transformer, $k = 13$, 2 seeds): OOD facts stay ≥ 0.9, and CoT composition reaches ≥ 0.9 and holds (last four evaluations ≥ 0.85). *Facts failed*: they end at 1.00 but dip to 0.86 and 0.40 between 20k and 60k during loss spikes (`criteria.json`); *composition failed* (0.59); the prediction that the $k = 1$ baseline keeps OOD facts below 0.8 also *failed* (0.84).
- **Fact and transfer payment** ($k = 26$ plus CoT-layout rows, 2 seeds): second-hop OOD ≥ 0.9 in each of the last four evaluations of both seeds. *Passed*: every one of the last four evaluations of both seeds is at least 0.92 (`criteria.json`). A third seed, added afterwards and not registered, also holds (0.98–0.99).
- **Matched control** (random exposed facts $\times 26$ + transfer): matches transfer alone, not fact and transfer payment; OOD facts below 0.95 and composition ≥ 0.05 below. *Facts passed* (0.94); *composition margin missed narrowly* (0.04); its mean, 0.94, is descriptively close to transfer alone (0.95).
- **Randomised dose** (transformer, $k \in \{1, \ldots, 32\}$ per underexposed fact): accuracy rises with k at $\lambda = 1$ with at most one inversion, and the spread at $\lambda = 0.1$ is at most half of it. *Passed* at both checkpoints (0.93 to 1.00 at $\lambda = 1$; spread at $\lambda = 0.1$ 0.0122 at 19.4k steps and 0.0063 at 50k). Half-survival dose inside $[1, 32]$: *failed*, because $k = 1$ already keeps 0.93.
- **Held-out test of a one-coordinate model** (toy; 19 conditions of (β_1, β_2), schedule and λ; Appendix B): within $2\times$ in at least 17 of 19 and better than a β-independent null on the discriminating conditions. *Rejected* (10 of 19; the null is closer); its ordering prediction also missed (Spearman 0.76 against ≥ 0.8). The survival criterion (first passage) was chosen on already-measured thresholds before any held-out result was read.
- **Rare rehearsal without weight decay across** (β_1, β_2): the harm is smaller at $\beta_2 = 0.9$. *Rejected*: at fixed $\beta_1 = 0$ and $p = 0.01$, $\beta_2 = 0.9$ leaves lower final OOD (0.63–0.93) than $\beta_2 = 0.999$ (0.82–0.94); at $p = 0.003$ the settings are similar.
- **Dose–response on natural variation**: *uninformative*. ID exposures are concentrated at 11–45, so there is no dynamic range.
- **Transformer fact memory** (a 4-layer GPT memorising facts, a subset of which stops being trained): those facts hold at $\lambda = 0$ and erode on the weight-decay clock at $\lambda > 0$. *Rejected*: the first design was confounded (the subset filled half of each installation batch) and was redesigned before its grid ran; in the redesign the subset is forgotten at $\lambda = 0$ too (0.99 to 0.21), which rejects a decay-only explanation without separating interference from optimiser instability.
- **Pretrained LM erosion**: $\lambda = 0$ retains installed reversal. *Rejected*: interference erodes it at fine-tuning rates.
- **The clocks in a transformer's fact memory** (4-layer GPT, 40k facts, full batch; a fixed 10% stops being trained or is rehearsed on a fraction p of steps). Loss clock: 13 of 18 runs with $\lambda > 0$ collapse the whole memory after the fork (loss up to 8, accuracy on the facts still trained below 0.9) and are excluded; the rest leave too few cells for the registered slope. Without weight decay 5 of 6 runs keep the facts (*passed*). Keep threshold, predicted from the toy before any fork (exponent in $[1.4, 2.8]$): 1.24, *not confirmed, not killed*, with 29 of 30 runs spiking. Adam's $\beta_2 = 0.99$ and $\epsilon \in \{10^{-6}, 10^{-5}\}$ do not stop the spikes (*killed / inconclusive* by their registered rules). In this model any abrupt change at the fork, of η, λ or the data, collapses the memory under Adam while $\eta = 10^{-3}$. With η lowered to $3 \cdot 10^{-4}$ at the fork (registered before its runs), 6 of 16 runs spike, all at $\lambda = 0.3$ (below the registered inconclusive bar of 8), and the exponent is 2.34: *passed*. The $\lambda = 1$ keep curve is monotone; the $\lambda = 0.3$ curve, where the six spikes are, is not, and read at its last crossing it would give 0.7 (Table 4). A registered test at four weight decays and four seeds is running.

B Held-out test across Adam's moments

A one-coordinate model of Adam's recurrence, paid only on rehearsal steps, predicts the threshold: simulate Adam's recurrence under the actual schedule and count the component as lost once it falls below a critical strength (`src/toys/adam_payment.py`). Calibrated on $\lambda = 1$, it gives $p_{50} =$

0.061 at $\lambda = 3$ (measured 0.066); the survival criterion (first passage rather than final level) was chosen using the already-measured thresholds, so that agreement is not independent. Its registered predictions for 19 held-out conditions fail: 10 of 19 fall within $2\times$ (9 of the 18 not used to choose the criterion), and where it differs from a null in which p_{50} depends on neither the moments nor the schedule, the null is closer (geometric-mean error $1.44\times$ against $2.44\times$). The largest miss: at $(\beta_1, \beta_2) = (0, 0.9)$ the account predicts that $\lambda = 1$ needs $2.2\times$ *more* rehearsal; it needs $3\times$ *less* ($p_{50} = 0.0021$ against 0.0060).

Networks installed with the default Adam moments $(0.9, 0.999)$ for 5,000 steps continued under $(\beta_1, \beta_2) \in \{(0.9, 0.999), (0.9, 0.99), (0.9, 0.9), (0, 0.999), (0, 0.9)\}$, with the optimiser state carried over, and with rehearsal either random (Bernoulli) or periodic, at $\lambda \in \{1, 3\}$, ten rehearsal rates from 0.002 to 0.3 and 4 seeds (`src/slurm/wd_paired.txt`). The account's predictions were written from pre-existing data before any of these runs was read (`predictions_B.json`, checksum in the decisions ledger). Figure 6 shows the outcome: the predictions are more dispersed than the measurements at $\lambda = 1$ and less at $\lambda = 3$, and at $\lambda = 1$ they order the moment settings wrongly. Removing momentum lowers the threshold at $\lambda = 1$ with $\beta_2 = 0.9$, and raises it at $\lambda = 3$ with $\beta_2 = 0.999$, where it also destabilises training (atomic accuracy dips to 0.14 and 0.22 in two of four seeds at $p = 0.1$), so that last cell is confounded. Without rehearsal, the five moment settings cross 0.715 at 2,400–3,650 steps at $\lambda = 1$ and 100–600 at $\lambda = 3$ (`summary_B2.json`).

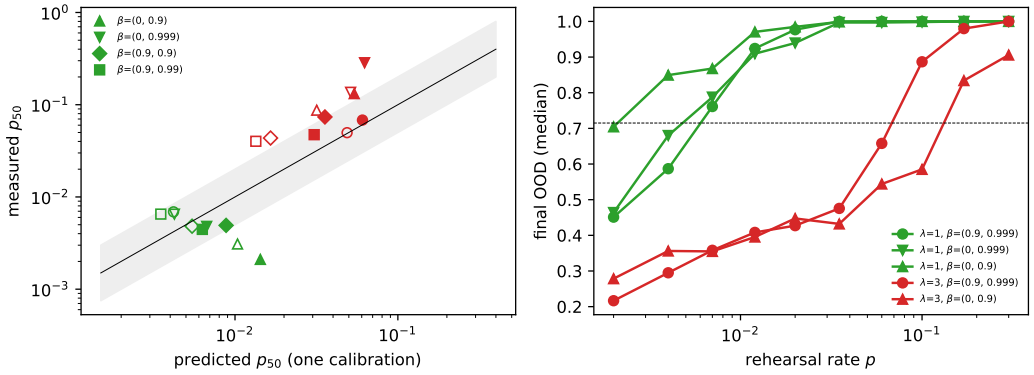

Figure 6: Left: measured against predicted half-survival rehearsal rate (one calibration, $\lambda = 1$ default moments, random schedule); filled: random schedule, open: periodic; green $\lambda = 1$, red $\lambda = 3$; shaded, within $2\times$. Right: final OOD against rehearsal rate for selected moments. Runs `wd_paired`; `summary_B.json`.

C SGD AND OTHER EFFECTS OF RARE REHEARSAL

SGD. In an SGD sweep with its own SGD installation the thresholds looked linear in λ ($p_{50} = 0.022$, 0.070, 0.217 at $\lambda = 0.01$, 0.033, 0.1; exponent 1.00; Table 6). On shared networks switched from Adam to SGD at the fork, many low-rehearsal conditions collapse the whole network (median final atomic accuracy 0.03 in most cells with $\lambda > 0$ and rehearsal rates up to 0.02; 445 of 528 rehearsed forks lose more than a tenth of their atomic facts), and five of six weight decays fail the registered requirement that the task stays intact at the threshold.

Damage across Adam's moments. In the held-out sweep, 106 of 320 rehearsed runs with $\beta_2 = 0.999$ lose more than a tenth of their atomic facts after the switch (lowest 0.08; at $\lambda = 1$, $p = 0.002$ the median minimum is 0.61), and none of 480 with $\beta_2 \leq 0.99$ do (lowest 0.93).

Rare rehearsal without weight decay. At $\lambda = 0$, where no rehearsal keeps composition at 1.00 in 12 of 12 runs, rehearsal on 0.3% of steps lowers it to 0.33–0.89, under SGD too (0.84–0.92) and with $\beta_2 = 0.9$ as well. This harm is not explained by the account.

Table 6: The rehearsal sweep under SGD with momentum (median final OOD; 4 seeds). Source: `tables/sgd.tex`, runs `wd_sgd`.

λ (SGD)	$p = 0$	0.003	0.01	0.03	0.1	0.3
0.01	0.57	0.03	0.03	1.00	1.00	1.00
0.0333	0.31	0.03	0.03	0.03	1.00	1.00
0.1	0.25	0.03	0.03	0.02	0.03	1.00