EFFECTS OF TOKEN-LEVEL LOSS REWEIGHTING ON LONG-RANGE RECALL IN CHUNKED TRAINING OF MAMBA-2

Anonymous authors
Paper under double-blind review

ABSTRACT

Truncated backpropagation through time trains a stateful sequence model by carrying the state across chunks while stopping gradients at chunk boundaries, which biases learning toward short-range dependencies. A standard approach is to use a longer gradient window. In this study, we investigate whether extending the gradient path is necessary for improving long-range recall. Our hypothesis is that, with the state carried forward, which tokens the objective emphasizes can matter more for long-range recall than how far the gradient reaches. We test this hypothesis in continued training of Mamba-2 (1.3B and 370M) on PG-19, varying two choices independently: (1) whether the loss is the standard cross-entropy or is reweighted toward rare tokens absent from the recent context, and (2) whether gradients stop at chunk boundaries or flow through the whole sequence. We deliberately use a simple reweighting scheme as a probe of the hypothesis: a deterministic mask on token ids with a single weight set by a static calibration rule. With the gradient window fixed at one chunk, reweighting raises variable-tracking accuracy at the 8K training length from 18.3 to 33.1 and needle retrieval at four times the training length from 67.7 to 94.0. At 1.3B and the 8K training length, full backpropagation gives no statistically significant improvement under the standard objective. The gains are task-dependent and trade off against perplexity and accuracy on an aggregation task (frequent-word extraction). In this setting, long-range recall improved by changing what the objective emphasizes, without extending the gradient path.

1 INTRODUCTION

Sequence models with a recurrent state—recurrent networks, state-space models, linear attention—are typically trained on long sequences by truncated backpropagation through time (TBPTT; Williams & Peng, 1990): the sequence is split into chunks, the state at the end of each chunk is carried into the next, and the gradient is stopped at the boundary (Zaremba et al., 2014; Merity et al., 2018). The forward pass still sees the whole sequence; only the learning signal is confined to a chunk. Instead, as a trade-off the truncated gradient omits every dependency that crosses a boundary, so it is biased toward short-range structure (Tallec & Ollivier, 2017). The usual response is to widen the window the gradient flows through.

We ask whether extending the gradient path is necessary to improve long-range recall in chunked training. Our hypothesis is that, once the state is carried forward, *which tokens the objective emphasizes* can matter more than *how far the gradient reaches*: a model trained with a short gradient window may learn to keep and use distant information if the objective concentrates the learning signal on the tokens that require it. In our setting, the hypothesis concerns recall of a token that can be predicted only from evidence far back in the context, rather than long-context ability in general.

We test this hypothesis in continued training of Mamba-2 (1.3B and 370M) on PG-19, varying two factors independently. One factor is the *gradient path*: chunked training with the state carried and the gradient truncated at one chunk, or full backpropagation through the whole sequence. The other is the *objective*: standard cross-entropy, or a reweighted cross-entropy that upweights targets that are rare in the corpus and absent from the recent context (§3). We test the hypothesis with a deliberately

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.

minimal reweighting objective, used as a probe rather than an optimized method. It changes which tokens are emphasized without changing how far gradients flow: the mask is a deterministic function of token ids—no reference model or synthetic data—with a single weight fixed by calibration.

We run the test on Mamba-2 (Dao & Gu, 2024), a state-space model whose fixed-size recurrent state is its entire memory of the prefix, so the premise of the hypothesis holds literally. We continue training at 1.3B (primary) and 370M for one epoch of PG-19 with 8K sequences in 2K chunks, and evaluate on a frozen RULER instance set (Hsieh et al., 2024) within the training length (8K) and beyond it (32K), reporting perplexity and an aggregation task (frequent-word extraction) as trade-offs (§4).

Two findings follow. (1) With the truncation window fixed at one chunk, reweighting raises variable-tracking accuracy at 8K from 18.3 to 33.1 and needle-retrieval accuracy at 32K from 67.7 to 94.0 (Table 1) with a trade-off between recall and perplexity at 1.3B (§5.4). (2) At 1.3B and the 8K training length, extending the gradient path to the full sequence under the standard objective yields no statistically significant improvement on any of seven RULER tasks; at shorter lengths and at 370M its effects are mixed in both directions (§5). We do not conclude that the gradient path is irrelevant, only that lengthening it did not help in the primary setting.

2 Related Work

State-space models and their recall limits. Structured state-space models (SSMs; Gu et al., 2022) and their selective variants (Gu & Dao, 2024; Dao & Gu, 2024) process a sequence with a fixed-size recurrent state, which is what makes chunked training natural: the state at a chunk boundary carries the model's entire memory of the prefix. The same property bounds what such a model can retrieve. A large share of the perplexity gap between attention and gated-convolution models is attributable to in-context recall (Arora et al., 2024); SSMs copy and retrieve from long contexts less reliably than Transformers (Jelassi et al., 2024; Waleffe et al., 2024); and a recurrent model trained on sequences that are short relative to its state size never learns to discard old information, so its state collapses on contexts longer than those it was trained on (Chen et al., 2025). Ben-Kish et al. (2024) extend the usable context of a trained Mamba without further training by dropping the tokens to which the model's selective gate assigns the least weight. We study the complementary question of what continued training does to recall, and we vary the training procedure rather than the architecture or the inference procedure.

Chunked training and state carry. Chunked training is truncated backpropagation through time (Williams & Peng, 1990) applied to an SSM: the recurrent state is carried across chunks but detached, the standard recipe of recurrent language modeling (Zaremba et al., 2014; Merity et al., 2018). These chunks are not those of the Mamba-2 SSD kernel (Dao & Gu, 2024), which tiles the same recurrence into 256-token blocks as an exact reformulation through which gradients flow in full. The same stop-gradient carry appears in attention models: Transformer-XL caches the activations of the previous segment under a stop-gradient (Dai et al., 2019), and the Compressive Transformer compresses that cache further (Rae et al., 2020). Truncation biases the gradient toward dependencies shorter than the window (Tallec & Ollivier, 2017); the same work removes the bias in expectation with random truncation lengths and compensating weights. Rather than correcting the truncation, we ask whether removing it altogether, by backpropagating through the whole sequence, changes recall, and whether changing the objective does.

Token-level loss reweighting. Prior work reweights the per-token loss by a rule that depends on the model: focal loss by the model's confidence (Lin et al., 2017), RHO-Loss and Rho-1 by the excess loss over a reference model (Mindermann et al., 2022; Lin et al., 2024), and, closest to our setting, LongCE by contrasting predictions with and without the long context (Fang et al., 2025), LongRecipe by the shift in predictions over training (Hu et al., 2024), and the token-weighting schemes of Helm et al. (2025), which score tokens by the confidence gap between a long-context and a short-context model and generalize LongCE. Our reweighting (§3) is computed from token ids alone, corpus frequency and a lookback window, with no reference, long-context, or short-context model, so the weights are fixed before training and identical under chunked and full backpropagation; the objective factor of our design is therefore independent of the gradient-path factor.

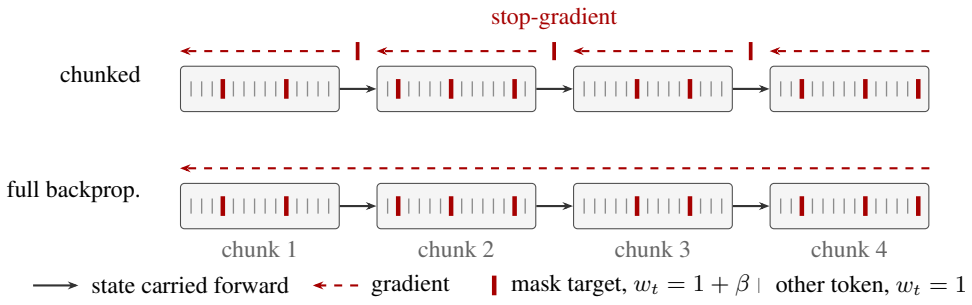

Figure 1: The two factors on one training sequence of four chunks. Rows: the gradient path. In both rows the recurrent state is carried across every chunk boundary (solid arrows), so the forward pass is a function of the whole sequence; in chunked training the gradient (dashed) stops at each boundary, and in full backpropagation it reaches every earlier position. Inside the chunks: the objective. Ticks are token positions; the heavy ticks are the mask's targets, targets that are rare in the corpus and absent from the recent input window, whose cross-entropy is multiplied by $1 + \beta$ (Eq. (1)). The mask depends on the token ids only and is the same under either gradient path.

Long-context evaluation. Needle-in-a-haystack tests (Kamradt, 2023) and the RULER suite (Hsieh et al., 2024) probe retrieval, multi-hop tracking, and aggregation at controllable context lengths. Eq. (1) upweights rare targets that are absent from the recent input window; RULER's retrieval and tracking tasks test precisely whether such a target can be produced from far back, and its length control lets us evaluate within and beyond the 8K training length. We use RULER at a pinned commit with frozen instance sets shared by every arm (§4). Fang et al. (2025) document that perplexity and long-context benchmark performance can move independently, which is why we report both and treat perplexity as a cost rather than a proxy. Synthetic probes measure recall in isolation from natural-language comprehension; we do not evaluate on benchmarks built from natural documents, and we return to this in §8.

3 Method

We consider a sequence model with a recurrent state. Given input tokens $x_1, \ldots, x_L$, the model maintains a state $s_t = f_\theta(s_{t-1}, x_t)$ and predicts the next token, the target $y_t = x_{t+1}$, from s_t: $p_\theta(y_t \mid s_t)$, which we also write $p_\theta(y_t \mid x_{\le t})$ since s_t is a function of the prefix. Recurrent networks, state-space models, and linear attention have this form, and the two training choices below apply to any of them: how far the gradient reaches, and which tokens the objective emphasizes. Crossing the two gives the four training conditions, or *arms*, compared in §4. Figure 1 shows the two factors on one sequence.

3.1 Gradient path

A training sequence of length L is processed as K consecutive chunks of length $C = L/K$. In *chunked training* the state at the end of a chunk is carried into the next chunk, so the forward pass computes the same function of the whole sequence as an unchunked pass; the carried state is detached from the computation graph (a stop-gradient) at every boundary, so the loss of a chunk is backpropagated only within that chunk. This is truncated backpropagation through time with a window of one chunk: every chunk after the first is conditioned on a state that no gradient reaches, and the parameters that write the state receive no signal from the chunks that later read it, except indirectly through parameter sharing. In *full backpropagation* the detach is removed and the sequence is processed by one forward and one backward pass, so the loss at every position reaches the computation of every earlier position. In exact arithmetic and under the same objective the two differ only in the detach; the numerical difference between their implementations in our setting is measured in §4 and Appendix A.5.

3.2 OBJECTIVE

Each token's cross-entropy is multiplied by

$$w_t \;=\; 1 + \beta\, m_t, \qquad m_t \;=\; \mathbf{1}[\, y_t \notin \mathcal{V}_\tau \;\wedge\; y_t \notin \{x_{t-W+1}, \ldots, x_t\}\,], \tag{1}$$

where $\mathcal{V}_\tau$ is the smallest prefix of the vocabulary, ranked by unigram count on a sample of the training corpus, that covers a fraction τ of its token mass, and W is the length of the recency window. The mask m_t selects the positions whose target is rare in the corpus and did not occur among the W most recent inputs, the current input included, so a target that merely repeats the current input is never upweighted. The recency window is defined on the input stream and crosses chunk boundaries; positions before the start of a sequence count as non-occurrences, so the mask depends only on the sequence's own tokens. Both conditions are deterministic functions of token ids: the mask needs no reference model, no second forward pass, and no synthetic data, and it can be computed once before training. We refer to the targets at selected positions as the mask's *targets*.

The loss of a sequence is the mean over chunks of the weighted mean cross-entropy within each chunk,

$$L \;=\; \frac{1}{K}\sum_{c=1}^{K} \frac{\sum_{t\in\mathcal{C}_c} w_t\, \mathrm{CE}_t}{\sum_{t\in\mathcal{C}_c} w_t}, \qquad \mathrm{CE}_t = -\log p_\theta(y_t \mid x_{\le t}), \tag{2}$$

where $\mathcal{C}_c$ is the set of positions of chunk c. Eq. (2) normalizes each chunk by its own total weight, so a chunk's loss is fully determined before its own backward pass. This is a choice, not a necessity of chunked training: the weights do not depend on the model, so the sequence total $\sum_t w_t$ is available to either gradient path. Normalizing over the whole sequence, $\sum_t w_t\,\mathrm{CE}_t / \sum_t w_t$, coincides with Eq. (2) when $\beta = 0$ and the chunks have equal length, and otherwise differs by the ratio of chunk weight to sequence weight. Eq. (2) is the objective of every chunked arm; §4 states the normalization used by each full-backpropagation arm.

3.3 CALIBRATION

Let ρ be the fraction of positions the mask selects on a calibration sample of training sequences. We set $\beta = 1/\rho$, so that the expected weight per token is $\mathbb{E}[w] = 1 + \beta\rho = 2$: the added weight equals the base weight in expectation. The same rule with $\beta = (\mathbb{E}[w] - 1)/\rho$ gives any budget, and §8 varies it. The rule fixes the total weight budget but not its concentration: a selected token receives $1 + 1/\rho$ times the weight of an unselected one, so a sparser mask places the same budget on fewer tokens with a larger multiplier. The multiplier is thus determined by the mask and is not a free parameter. The reweighting therefore has four constants, τ, W, the budget $\mathbb{E}[w]$, and the calibration sample, from which β follows; Table 2 gives their values and how they were chosen. The standard objective is the case $\beta = 0$.

4 EXPERIMENTAL SETUP

Model and data. We continue training public Mamba-2 checkpoints (Dao & Gu, 2024) at 1.3B (primary) and 370M (scale) for one epoch of the PG-19 training split (Rae et al., 2020)(≈3.07B tokens) in sequences of L=8,192 tokens. Mamba-2 suits the test for three reasons: its fixed-size recurrent state makes the carried state the model's entire memory of the prefix, so the premise of §1 holds literally; state-space models recall from long contexts less reliably than Transformers (§2), so there is headroom; and its implementation provides both a chunked path with a carried state and a fused unchunked path, which the two gradient arms require. Optimization: AdamW (Loshchilov & Hutter, 2019) with $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.1, constant learning rate 10^{-5}, no warmup, gradient clipping at 1.0, 128 sequences per optimizer step (2,933 steps), fp32 master weights with bf16 autocast, 8×A6000 (48 GB). Every arm within a scale consumes the identical token stream in the identical order; seeds 42, 43, 44 set the permutation of training sequences and the framework's random state.

Hyperparameters and constants. Table 2 lists the constants. All of them were fixed before any training run and were not revised afterwards (Reproducibility statement). Appendix Table 6 reports a one-seed sensitivity sweep over $\mathbb{E}[w]$, W, and a gap-weighted variant of the mask, run after the

Table 1: Results for the 2×2 comparison of objective and gradient path after one epoch of continued training on PG-19. Each row corresponds to one combination of objective (standard cross-entropy or the rare $\wedge$ non-recent reweighting of Eq. (1)) and gradient path (chunked or full backpropagation). Accuracies reported in % as mean $\pm$ std over three seeds. Perplexity is evaluated on a fixed held-out subset of PG-19. Results for the remaining RULER tasks at 8K are reported in Appendix Table 4.

arm	VT@8K ↑	NIAH-s1@32K ↑	FWE@8K ↑	perplexity ↓
Mamba-2 1.3B				
standard + chunked	18.3 ± 1.2	67.7 ± 7.4	50.1 ± 1.0	10.34
standard + full backprop	21.1 ± 3.1	56.0 ± 1.7	44.6 ± 1.2	10.32
rare $\wedge$ non-recent + chunked	33.1 ± 1.0	94.0 ± 1.7	40.0 ± 0.9	12.18
rare $\wedge$ non-recent + full backprop	33.0 ± 1.7	64.7 ± 0.6	34.9 ± 0.5	12.15
Mamba-2 370M				
standard + chunked	8.9 ± 0.6	44.3 ± 0.6	42.6 ± 1.0	12.87
standard + full backprop	6.7 ± 0.1	52.0 ± 2.0	44.2 ± 0.7	12.86
rare $\wedge$ non-recent + chunked	11.1 ± 0.6	99.0 ± 0.0	28.6 ± 2.0	15.38
rare $\wedge$ non-recent + full backprop	9.3 ± 1.1	100.0 ± 0.0	30.9 ± 0.2	15.35

first main runs on a separate tuning set (1.3B, 0.2 epoch, its own evaluation instances); on that sweep $\mathbb{E}[w]{=}2$ and $W{=}256$ are not the best values, and we kept them because the specification had fixed them and a one-seed, 0.2-epoch sweep on a separate instance set is not a basis for revising it. τ was not swept. With the GPT-NeoX tokenizer (Black et al., 2022), the unigram counts are estimated on a fixed 180-book, 20.1M-token prefix of the PG-19 training split (held-out books excluded); on that sample $\mathcal{V}_\tau$ has 1,877 types and rare tokens occupy 0.200 of positions. ρ is measured on the first 50 training sequences of each run and logged at launch; the resulting multiplier $1 + \beta$ is 6.2–6.5 for the mask of Eq. (1), and the ablation masks of §6 span 5.6 to 21 at the same budget (Table 3).

Arms. Objective {standard, reweighted (Eq. (1))} $\times$ gradient path {chunked, full backpropagation}, at both scales; chunked training with the standard objective is the baseline and the primary control condition. Ablation arms at the same budget (1.3B, chunked): rare-only (rarity without the recency condition), random placebo (density-matched deterministic random mask), rare$\wedge$in-past (rare tokens with an occurrence in a previous chunk of the sequence).

Implementation. The two gradient paths run through different code: with a carried state the Mamba-2 implementation takes an unfused cache path, whereas the unchunked forward uses the fused kernel, and the full-backpropagation arms use activation checkpointing. A gradient-level diagnostic on one fixed window found the kernel-only discrepancy orders of magnitude smaller than the difference caused by detaching the carried state (Appendix A.5); a training-level comparison that holds kernel and checkpointing fixed was not run (§8). Evaluation uses the chunked path for every arm. In this implementation the chunked path does not realize the memory saving that motivates chunked training in general (Appendix A.5); our comparison is about learning, not cost.

Normalization of the full-backpropagation arms. At 1.3B both compared arms minimize Eq. (2): the reweighted full-backpropagation arm was trained with per-chunk normalization, and for the standard objective ($\beta{=}0$) whole-sequence normalization coincides with Eq. (2). At 370M the full-backpropagation arms retain whole-sequence normalization, so the 370M comparison under the reweighted objective is confounded by normalization scope as well (§7; the history and the size of the difference are in Appendix A.5).

Evaluation. We use a frozen RULER (Hsieh et al., 2024) instance set shared by every arm and seed, $n{=}100$ per task and length; the primary metric and the mask variant were chosen on it (Appendix A.5).

The primary metric is variable-tracking accuracy at the training length (VT@8K). Variable tracking asks which variables hold a given value after a chain of assignments spread across the context. Its answers are random five-letter variable names that occur only in those assignments, so the task probes targets that are typically rare and absent from the recent context, the population Eq. (1) upweights, at the training length (token census in Appendix A.5). Single-needle retrieval leaves no

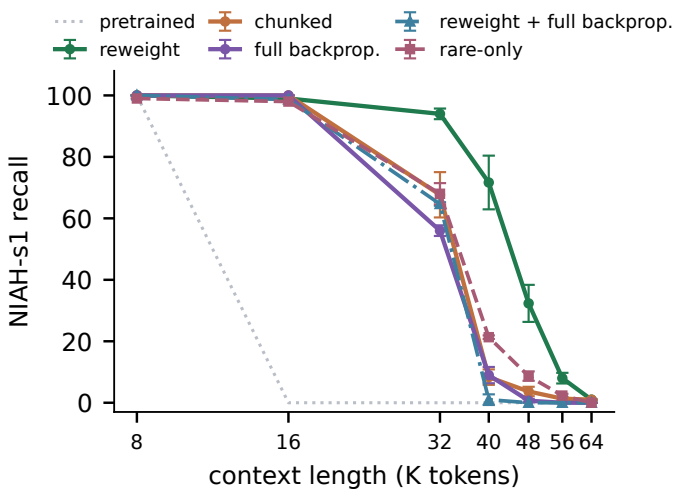

Figure 2: Single-needle recall (NIAH-s1) against context length (1.3B, means over three seeds; values in Table 10). The training length is 8K.

headroom at 8K (the pretrained model scores 100.0; Table 10). Costs: held-out perplexity on 200 fixed PG-19 sequences and frequent-word extraction (FWE); perplexity is reported as a cost, not a proxy (§2). Recall beyond the training length is measured by single-needle retrieval at 32K, whose seven-digit answer has the same profile, and at 40K–64K in Appendix A.3. Decoding is greedy with chunked prefill for every arm. Values are means over three seeds $\pm$ seed standard deviation unless stated otherwise.

5 Results

5.1 Reweighting with the gradient window fixed

Holding the gradient path fixed at one chunk and changing only the objective raises VT@8K from 18.3 ± 1.2 to 33.1 ± 1.0 and NIAH@32K from 67.7 ± 7.4 to 94.0 ± 1.7 (Table 1). The paired improvements are $+14.7$ points for VT (95% seed-level interval $[+11.2, +18.2]$) and $+26.3$ points for NIAH ($[+10.6, +42.1]$). These intervals are t intervals over the three paired seed differences (Appendix A.5).

The effect is task-dependent rather than uniform. At 8K, reweighting improves VT; single-needle retrieval remains at ceiling, while NIAH-s2, NIAH-mk1, and NIAH-mk2 change by at most 1.3 points. CWE and FWE decrease by 0.9 and 10.1 points, respectively (Appendix Table 4; §5.4).

Beyond the 8K training length, the difference in single-needle recall becomes more pronounced (Fig. 2, Table 10). Under standard chunked training, recall falls from 67.7 at 32K to 8.3 at 40K. With reweighting, it remains at 94.0 at 32K and 71.7 at 40K, then decreases to 32.3 at 48K and 8.0 at 56K.

5.2 Extending the gradient path

Under the standard objective, extending the gradient path does not yield a statistically detectable gain on the primary metric. Full backpropagation gives VT@8K of 21.1 ± 3.1, compared with 18.3 ± 1.2 under chunked training (Table 1), a mean paired difference of $+2.8$ points with a 95% seed-level interval of $[-5.0, +10.6]$. The interval does not bound the effect within ± 3 points.

Across the seven 8K RULER tasks at 1.3B, we detect no statistically significant improvement from full backpropagation (Appendix Table 4). Single-needle retrieval and NIAH-mk2 are identical between the two arms, while the full-backpropagation arm has lower mean accuracy on NIAH-s2,

Table 2: Experimental settings. Both model sizes (1.3B and 370M) share the same tokenizer, so ρ and β are identical at equal seed.

constant	value	how chosen
L, C, K	8,192; 2,048; 4	fixed before training
τ	0.8 ($\lvert\mathcal{V}_\tau\rvert = 1{,}877$ types)	fixed before training; not tuned
W	256	fixed before training; not tuned; sensitivity in Appendix Table 6
$\mathbb{E}[w]$	2 ($\beta = 1/\rho$)	fixed before training; not tuned; sensitivity in Appendix Table 6
calibration sample	first 50 training sequences of each run	fixed before training
ρ / β	$0.181/5.5$, $0.192/5.2$, $0.192/5.2$ (seeds 42/43/44)	derived from training data

Table 3: Ablation arms (Mamba-2 1.3B, chunked gradients). All reweighted arms have $\mathbb{E}[w]{=}2$ with $\beta = 1/\rho$ from each arm's measured mask density. The in-past arm is reported in Appendix A.1.

arm	β	VT@8K ↑	NIAH-s1@32K ↑	perplexity ↓
standard + chunked	—	18.3 ± 1.2	67.7 ± 7.4	10.34
density-matched random + chunked	5.6	17.5 ± 0.5	73.7 ± 5.8	10.40
rare-only + chunked	4.6	23.4 ± 1.2	68.0 ± 3.5	12.00
rare ∧ non-recent + chunked	5.5	33.1 ± 1.0	94.0 ± 1.7	12.18

NIAH-mk1, CWE, and FWE; for FWE, the paired difference is -5.5 points ($[-10.1, -1.0]$). At 32K, the point estimate is also lower (56.0 vs. 67.7), although the interval includes zero ($[-29.1, +5.8]$).

The broader grid is mixed: exploratory full-versus-chunked comparisons include positive as well as negative differences, particularly at shorter context lengths and at 370M (Appendix A.5). We therefore do not conclude that the gradient path is irrelevant. Rather, at 1.3B and the 8K training length, extending it under the standard objective provides no statistically detectable improvement on the evaluated tasks.

5.3 Full backpropagation on top of the reweighting

Adding full backpropagation to the reweighted objective leaves VT@8K without a detectable difference (33.0 ± 1.7 vs. 33.1 ± 1.0; per-seed difference -0.1, $[-6.3, +6.2]$, an interval too wide to support equivalence) and lowers NIAH@32K from 94.0 ± 1.7 to 64.7 ± 0.6 (-29.3, $[-32.2, -26.5]$; Table 1); at 40K, where the reweighted chunked arm still retrieves 71.7, it is at 1.0 (Table 10). At 8K it also shows the pattern of the standard full arm on the secondary tasks (NIAH-s2 9.3, NIAH-mk1 10.0, FWE 34.9; Appendix Table 4). The reweighted chunked model achieves the in-window gain without cross-chunk gradients. Under the reweighted objective, the tested full-BPTT configuration yields substantially lower single-needle recall at 32K and 40K.

5.4 Trade-offs

The reweighting raises held-out perplexity by 1.84 points ($10.34 \rightarrow 12.18$, $+18\%$; the standard full arm is at 10.32), lowers FWE at 8K by 10.1 points ($50.1 \rightarrow 40.0$), lowers CWE by 0.9, and regresses needle retrieval from the essay haystack at 4K (NIAH-s2 $46.7 \rightarrow 29.7$, NIAH-mk1 $20.3 \rightarrow 9.3$; Table 12) while changing by at most 1.3 points at 8K. Because the method was deliberately chosen as a simple probe rather than an optimized objective, more selective or adaptive weighting schemes may reduce these trade-offs.

6 Ablations

Does the placement of the weight matter? All ablation arms use chunked gradients and the same budget $\mathbb{E}[w]{=}2$, with β set from each arm's own mask density (Table 3). The random placebo, a deterministic random mask with the density of Eq. (1) and nearly the same ($\beta{=}5.6$ vs. 5.5), leaves

VT@8K where chunked training leaves it (17.5 ± 0.5 vs. 18.3 ± 1.2; per-seed difference -0.9, $[-3.9, +2.2]$), NIAH@32K within seed variance (73.7 ± 5.8 vs. 67.7 ± 7.4), and perplexity almost unchanged (10.40 vs. 10.34). The density-matched random placebo does not reproduce the gains, indicating that the placement of the added weight matters.

Which condition carries the effect? Dropping the recency condition (rare-only: every rare target is upweighted, β=4.6) retains about a third of the tracking gain (23.4 ± 1.2; $+5.1$ over chunked training, $[+0.6, +9.6]$, against $+14.7$ for the full mask) and none of the needle gain at 32K (68.0 ± 3.5 vs. 67.7 ± 7.4), at nearly the same perplexity cost (12.00 vs. 12.18). The recency condition, which removes from the mask the rare targets that the recent context already supplies, accounts for much of the observed gain while adding little perplexity cost over rare-only weighting.

What the mask selects. On 220 held-out sequences, 79.8% of the selected targets have no earlier occurrence anywhere in the 8,192-token sequence and only 11.9% occur in a previous chunk (Appendix Table 5): the mask mostly selects rare tokens that neither the local context nor the carried state can supply by copying.

7 SCALE

The main needle-recall gain persists at 370M. With chunked gradients, the rare $\wedge$ non-recent objective raises NIAH-s1@32K from 44.3 ± 0.6 to 99.0 ± 0.0, a gain of $+54.7$ points (95% seed-level interval $[+53.2, +56.1]$), compared with $+26.3$ points at 1.3B. In contrast, VT@8K is near floor at this scale (8.9 for standard and 11.1 for the reweighted objective), limiting the informativeness of the 8K tracking comparison.

The perplexity cost is larger at 370M: reweighting increases held-out perplexity from 12.87 to 15.38 ($+2.51$), compared with $+1.84$ at 1.3B. Full-backpropagation comparisons at 370M are reported in Table 1; they are less informative because retrieval under the reweighted objective is at ceiling and the full-backpropagation arms use a different normalization scope (§4).

8 CHARACTERIZATION AND LIMITS

Sensitivity. On a separate tuning set (one seed, 0.2 epoch; Appendix Table 6) a larger weight budget raises VT@8K and raises perplexity, the gap-weighted variants are lower at every budget, and $W \in \{128, 256, 512\}$ changes VT@8K by at most 2.4 points; the main runs keep $\mathbb{E}[w]$=2 and W=256, fixed before the sweep (§4).

Limitations. One architecture family and one corpus, at two model sizes (370M and 1.3B; the needle-recall gain replicates across them, §7); transfer to LSTMs, linear attention, or other corpora is untested. Synthetic recall probes only; natural-document benchmarks are not evaluated, and the essay-haystack regression at 4K shows that the gain is not uniform. No head-to-head comparison with LongCE, RHO-1, or DeciMamba was run. The gradient-path comparison is underpowered at the seed level: with three seeds its null result does not bound the effect within ± 3 points ($[-5.0, +10.6]$ on VT@8K; §5). τ is fixed without a sensitivity study. At 370M the full-backpropagation arms were trained with whole-sequence normalization and were not re-run, so the 370M comparison under the reweighted objective differs in normalization scope as well as in gradient path (§4); at 1.3B the per-chunk re-run and the earlier whole-sequence run agree within seed variance at every seed (Appendix A.5). A training-level comparison that toggles only the detach, holding kernel and checkpointing fixed, was not run; the decomposition of Appendix A.5 is at the level of a single gradient. Only three chunk boundaries per sequence are crossed (8K/2K), and the memory saving that motivates chunked training in general is not realized by this implementation (§4). The in-past ablation is a confounded single-seed arm (Appendix A.1), and the parameter-dynamics tables of Appendix A.4 are single seed.

9 Discussion and Conclusion

What the results establish. In continued training of Mamba-2 on PG-19, with the state carried across 2K chunks of 8K sequences: (i) changing which tokens the objective emphasizes, under chunk-local gradients, raised recall of rare tokens from far back—variable tracking at the training length and needle retrieval up to six times it—and, for needle retrieval, at two scales; (ii) at 1.3B and the 8K training length, extending the gradient to the whole sequence improved no task significantly and lowered several under either objective, while at shorter lengths and at 370M its effects were mixed; (iii) with the reweighted objective it lowered needle recall beyond the training length. These statements are about the metrics and lengths measured. They do not say that the gradient path never matters (the seed-level intervals do not bound its effect within ± 3 points), that the objective is the best of its kind, or that the gain extends to natural-document tasks (the essay-haystack regression at 4K says it does not extend uniformly).

What might explain it. We consider two possible explanations. First, the mask mostly selects tokens with no earlier occurrence in the sequence, which cannot be supplied by direct copying (§6), so the gradient it concentrates is a gradient toward predicting rare content from the state rather than from the local window; the recency condition is what makes this so, and it carries most of the effect. Second, the full-backpropagation arms move the parameters in nearly the same direction as their chunked counterparts (update cosine 0.98 for the path-only pairs, against 0.84 for the objective-only pairs and 0.99 between seeds; Table 15), and the learned decay parameters move by about 0.3% under both paths (Table 14): in this regime the gradient path changes the update little and the objective changes it more. Why the full path lowers recall beyond the training length is not identified.

Conclusion. The usual response to what gradient truncation removes is a longer gradient window. In our primary setting, Mamba-2 1.3B at the 8K training length, the objective was the lever that moved long-range recall, whereas extending the gradient window yielded no detectable benefit and, under the reweighted objective, reduced recall beyond the training length. Whether this holds for other stateful models, other corpora, other scales, and natural-document recall is the question the hypothesis now poses.

AI use statement

The research questions, the hypothesis, the experimental design, and the claims of this paper were set jointly by the three authors. One author ran the experiments, wrote the code, and verified the numbers in the tables against the result ledgers and launch banners; two authors wrote the text. A coding assistant (Claude Code, Anthropic) was used under that author's direction throughout. Language-model-based reviews were used as review aids. Every code change was reviewed and merged by an author, and the implementation was also reviewed by a co-author Generative AI tools were also used to assist with literature retrieval and discovery, to draft and polish portions of the manuscript, and to support code implementation and research execution.

Ethics statement

The work uses public corpora and public model checkpoints, targets a training-efficiency and capability-measurement question, and introduces no new data collection, human subjects, or foreseeable dual-use risk beyond general language-model capability research. Total compute for all confirmatory and exploratory runs is approximately 7,000 GPU-hours on 48 GB-class GPUs.

Reproducibility statement

All models are public Mamba-2 checkpoints; the training data is the public PG-19 corpus and the evaluation sets are generated by the public RULER harness at a pinned commit. The method is Eq. (1), fully specified in §3; the mask is computed from token ids alone. Training and evaluation hyperparameters are stated in §4 and Appendix A.5; per-seed values behind every table are in Appendix A.2. Every reported number is traceable to a specific run through a logged launch banner

and ledger row. The constants of Table 2 are recorded in a dated specification that predates the first training launch and has not changed since, and every launch banner echoes them.

REFERENCES

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Assaf Ben-Kish, Itamar Zimerman, Shady Abu-Hussein, Nadav Cohen, Amir Globerson, Lior Wolf, and Raja Giryes. DeciMamba: Exploring the length extrapolation potential of mamba. *arXiv preprint arXiv:2406.14528*, 2024.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of the ACL Workshop on Challenges & Perspectives in Creating Large Language Models*, 2022.

Yingfa Chen, Xinrong Zhang, Shengding Hu, Xu Han, Zhiyuan Liu, and Maosong Sun. Stuffed Mamba: Oversized states lead to the inability to forget. In *Conference on Language Modeling (COLM)*, 2025.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 10041–10071. PMLR, 2024.

Lizhe Fang, Yifei Wang, Zhaoyang Liu, Chenheng Zhang, Stefanie Jegelka, Jinyang Gao, Bolin Ding, and Yisen Wang. What is wrong with perplexity for long-context language modeling? In *International Conference on Learning Representations (ICLR)*, 2025.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling (COLM)*, 2024.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations (ICLR)*, 2022.

Falko Helm, Nico Daheim, and Iryna Gurevych. Token weighting for long-range language modeling. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling (COLM)*, 2024.

Zhiyuan Hu, Yuliang Liu, Jiangchao Zhao, Suyuchen Wang, Yan Wang, Wei Shen, Qing Gu, Anh Tuan Luu, See-Kiong Ng, Zhiwei Jiang, and Bryan Hooi. LongRecipe: Recipe for efficient long context generalization in large language models. *arXiv preprint arXiv:2409.00509*, 2024.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Gregory Kamradt. Needle in a haystack — pressure testing LLMs. `https://github.com/gkamradt/LLMTest_NeedleInAHaystack`, 2023.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. Not all tokens are what you need for pretraining. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing LSTM language models. In *International Conference on Learning Representations (ICLR)*, 2018.

Sören Mindermann, Jan M. Brauner, Muhammed T. Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations (ICLR)*, 2020. PG-19 corpus.

Corentin Tallec and Yann Ollivier. Unbiasing truncated backpropagation through time. *arXiv preprint arXiv:1705.08209*, 2017.

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, Garvit Kulshreshtha, Vartika Singh, Jared Casper, Jan Kautz, Mohammad Shoeybi, and Bryan Catanzaro. An empirical study of mamba-based language models. *arXiv preprint arXiv:2406.07887*, 2024.

Ronald J. Williams and Jing Peng. An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural Computation*, 2(4):490–501, 1990.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

A Appendix

A.1 Additional Tables

Table 4 gives all seven RULER tasks at the 8K training length; Table 5 is the census of the mask's targets (§6); Table 6 is the tuning-set sensitivity sweep and Table 7 the parameter interpolation (§8).

Table 4: All seven tasks at the 8,192-token training length (1.3B, evaluation set). Accuracies in %, mean ± seed sd over three seeds; sd omitted where zero. NIAH-s2 and NIAH-mk1 use the essay haystack. Rare-only removes the non-recent condition from the rare ∧ non-recent objective; the density-matched random arm replaces the proposed mask with a random mask of the same density.

arm	NIAH-s1	NIAH-s2	NIAH-mk1	NIAH-mk2	VT	CWE	FWE
standard + chunked	100.0	22.7 ± 0.6	16.0 ± 0.0	0.0	18.3 ± 1.2	2.3 ± 0.1	50.1 ± 1.0
standard + full backprop.	100.0	6.7 ± 0.6	10.3 ± 0.6	0.0	21.1 ± 3.1	0.3 ± 0.1	44.6 ± 1.2
rare ∧ non-recent + chunked	100.0	22.3 ± 0.6	17.3 ± 0.6	0.0	33.1 ± 1.0	1.4 ± 0.2	40.0 ± 0.9
rare ∧ non-recent + full backprop.	100.0	9.3 ± 0.6	10.0 ± 0.0	0.0	33.0 ± 1.7	1.3 ± 0.2	34.9 ± 0.5
rare-only + chunked	99.0	22.0 ± 1.0	17.0 ± 0.0	0.0	23.4 ± 1.2	2.9 ± 0.3	31.4 ± 0.5
density-matched random + chunked	100.0	22.0 ± 0.0	16.0 ± 0.0	0.0	17.5 ± 0.5	1.8 ± 0.2	50.6 ± 0.2

Table 5: Composition of the mask's targets on 220 held-out PG-19 sequences (mask density 0.176 held-out; 0.181 on the training stream).

selected target ($\mathrm{rare} \wedge \neg\mathrm{recent}$)	share
no earlier occurrence in the 8,192-token sequence	79.8%
earlier occurrence > 256 tokens back, same chunk	8.3%
earlier occurrence in a previous chunk	11.9%

The in-past arm. The ablation arm that upweights only rare tokens with an occurrence in a previous chunk (rare $\wedge$ in-past + chunked; one seed) was designed to test cross-chunk copying directly. It is harmful (VT@8K 15.4, NIAH@32K 0.0, perplexity 12.55; per-length values in Appendix A.3), but it is confounded: its mask is sparse, so the budget rule gives β=20.1, four times the multiplier of the other arms; the mask is empty in the first chunk of every sequence, so its weight lies in chunks 2–4 (26/34/40%); its gradient norm before clipping is more than twice that of every other arm (Table 13); and it was run with one seed. We report it and do not use it to rule the cross-chunk-copy mechanism in or out.

Table 6: Budget $\mathbb{E}[w]$, gap-weighted shape exponent γ, and recency window W on the tuning set (1.3B, 0.2 epoch, one seed, a separate evaluation root; values are not comparable to the evaluation set). γ=0 is the flat mask of Eq. (1); the gap-weighted family is defined in Appendix A.5.

	VT@8K			ppl
$\mathbb{E}[w]$	γ=0	γ=1	γ=2	(γ=0)
1.5	18.8	15.6	13.2	11.35
2	19.0	18.6	15.4	12.39
3	22.2	19.4	18.0	14.17
5	28.8	23.2	22.0	17.17
W=128 / 256 / 512 ($\mathbb{E}[w]$=2, γ=0)	21.4 / 19.0 / 21.4			12.32 / 12.39 / 12.46

Table 7: Weight interpolation $\theta_\alpha = \alpha\,\theta_{\text{reweight}} + (1-\alpha)\,\theta_{\text{chunked}}$ (no training), on the seed-42 path at both scales: the endpoints are the seed-42 chunked arms (standard and rare $\wedge$ non-recent) and every α is a single interpolation. At 1.3B, α=0.5 was also built from the seed-43 and seed-44 parents (VT@8K 24.4 and 26.6; NIAH@32K 94 and 86); Appendix A.5 records their provenance. Rows are the columns of Table 1.

α	0 (standard)	0.25	0.5	0.75	1 (rare $\wedge$ non-recent)
1.3B (single seed)					
VT@8K	19.4	22.2	26.0	26.8	34.0
NIAH-s1@32K	65.0	80.0	92.0	94.0	95.0
FWE@8K	51.0	48.7	45.0	42.0	40.7
held-out ppl	10.34	10.44	10.77	11.36	12.26
370M (single seed)					
VT@8K	8.8	8.2	9.2	10.8	10.4
NIAH-s1@32K	45.0	51.0	70.0	92.0	99.0
FWE@8K	42.0	38.0	38.0	32.3	26.3
held-out ppl	12.87	13.02	13.45	14.24	15.48

A.2 PER-SEED RESULTS

Table 8: Per-seed values behind Table 1 (1.3B, Set A). The last block is the earlier whole-sequence-normalized run of the reweighted full-backpropagation arm (§4), shown for comparison with the per-chunk run above it.

arm	seed	VT@8K	NIAH@32K	ppl
standard + chunked	42	19.4	65.0	10.34
	43	17.0	76.0	10.34
	44	18.6	62.0	10.34
standard + full backprop.	42	23.8	54.0	10.32
	43	21.8	57.0	10.32
	44	17.8	57.0	10.31
rare $\wedge$ non-recent + chunked	42	34.0	95.0	12.26
	43	33.2	95.0	12.14
	44	32.0	92.0	12.14
rare $\wedge$ non-recent + full backprop.	42	32.8	65.0	12.24
	43	31.4	65.0	12.11
	44	34.8	64.0	12.12
rare $\wedge$ non-recent + full backprop. (whole-sequence norm.; not in Table 1)	42	32.8	65.0	12.25
	43	32.2	65.0	12.11
	44	33.6	65.0	12.14

Table 9: 370M per-seed values (Set A); the 370M full-backpropagation arms were trained with whole-sequence normalization (§4).

scale/arm	seed	VT@8K	NIAH@32K	ppl
370M rare $\wedge$ non-recent + chunked	42	10.4	99.0	15.48
	43	11.6	99.0	15.33
	44	11.4	99.0	15.33
370M standard + chunked	42	8.8	45.0	12.87
	43	8.4	44.0	12.87
	44	9.6	44.0	12.87
370M standard + full backprop.	42	6.8	52.0	12.86
	43	6.8	54.0	12.86
	44	6.6	50.0	12.86
370M rare $\wedge$ non-recent + full backprop.	42	8.2	100.0	15.46
	43	9.2	100.0	15.30
	44	10.4	100.0	15.31

A.3 LENGTH CURVES

Table 10: NIAH-s1 recall against context length (1.3B, Set A; means over three seeds; — not evaluated). The training length is 8K; $5\times$ is the 40K column and $7\times$ the 56K column. Per-seed values at 40K: rare $\wedge$ non-recent + chunked $79/74/62$, standard + chunked $8/11/6$.

arm	8K	16K	32K	40K	48K	56K	64K
standard + chunked	100.0	100.0	67.7	8.3	3.7	1.3	1.0
standard + full backprop.	100.0	100.0	56.0	9.0	0.7	0.0	0.0
rare $\wedge$ non-recent + chunked	100.0	99.0	94.0	71.7	32.3	8.0	1.0
rare $\wedge$ non-recent + full backprop.	100.0	98.7	64.7	1.0	0.0	0.0	0.0
rare-only + chunked	99.0	98.0	68.0	21.3	8.7	2.3	0.0
density-matched random + chunked	100.0	100.0	73.7	12.0	5.3	2.0	1.3
rare $\wedge$ in-past + chunked (n=1)	31.0	0.0	0.0	0.0	0.0	0.0	0.0
pretrained (no training)	100.0	0.0	0.0	0.0	0.0	—	0.0

Table 11: VT accuracy against context length (1.3B, Set A, mean $\pm$ seed sd over three seeds).

arm	1K	2K	4K	8K
standard + chunked	46.5 ± 0.8	49.9 ± 1.0	39.1 ± 0.6	18.3 ± 1.2
standard + full backprop.	48.4 ± 1.2	50.1 ± 0.8	40.4 ± 1.7	21.1 ± 3.1
rare $\wedge$ non-recent + chunked	56.9 ± 3.9	58.8 ± 1.3	51.8 ± 2.2	33.1 ± 1.0
rare $\wedge$ non-recent + full backprop.	54.7 ± 1.5	55.5 ± 0.5	52.2 ± 1.6	33.0 ± 1.7
rare-only + chunked	50.2 ± 0.9	53.7 ± 1.3	43.9 ± 0.9	23.4 ± 1.2
density-matched random + chunked	43.7 ± 0.6	47.2 ± 0.9	36.5 ± 1.5	17.5 ± 0.5
rare $\wedge$ in-past + chunked (n=1)	18.8	14.0	14.0	15.4
pretrained (no training)	25.8	29.6	19.8	13.4

Table 12: Secondary tasks within the training length, against context length (1.3B, Set A, means over three seeds; seed sd ≤ 3.5). NIAH-s2 and NIAH-mk1 use the essay haystack.

	NIAH-s2				NIAH-mk1				FWE			
arm	1K	2K	4K	8K	1K	2K	4K	8K	1K	2K	4K	8K
standard + chunked	100.0	89.0	46.7	22.7	53.0	25.0	20.3	16.0	70.7	64.0	56.3	50.1
standard + full backprop.	100.0	92.0	47.3	6.7	55.3	22.7	21.3	10.3	69.4	62.9	55.8	44.6
rare $\wedge$ non-recent + chunked	97.7	84.3	29.7	22.3	51.7	22.3	9.3	17.3	62.1	56.1	47.7	40.0
rare $\wedge$ non-recent + full backprop.	97.0	87.3	32.0	9.3	51.0	24.0	10.0	10.0	58.0	54.0	46.7	34.9
rare-only + chunked	96.7	83.3	30.3	22.0	48.0	20.0	9.0	17.0	51.3	45.9	44.3	31.4
density-matched random + chunked	100.0	90.7	47.0	22.0	54.7	25.0	20.7	16.0	71.1	64.4	57.2	50.6
pretrained (no training)	100.0	76.0	0.0	0.0	52.0	26.0	4.0	0.0	61.0	55.0	43.0	1.0

A.4 TRAINING DYNAMICS

Table 13: Gradient norm before clipping, logged every 10 optimizer steps (293 points per run; 1.3B, one epoch). Means over three seeds where $n{=}3$ (per-seed means differ by at most 0.014); $n{=}1$ marks arms whose norm log was recorded on one seed. Clipping at 1.0 was triggered only at the first logged step of the full-backprop. $s42$ run and of the reweighted runs. *The whole-sequence-normalized run (§4), here and in Tables 14 and 15.

arm	mean	median	p95	first 100 steps	last 500 steps
standard + chunked	0.130	0.121	0.188	0.168	0.139
standard + full backprop.	0.125	0.113	0.162	0.251	0.120
rare ∧ non-recent + chunked	0.162	0.155	0.192	0.295	0.159
rare ∧ non-recent + full backprop.*	0.162	0.154	0.191	0.292	0.157
rare-only + chunked ($n{=}1$)	0.160	0.150	0.190	0.302	0.158
density-matched random + chunked ($n{=}1$)	0.160	0.151	0.224	0.204	0.159
rare ∧ in-past + chunked ($n{=}1$)	0.356	0.317	0.612	0.398	0.419

Table 14: Learned decay parameters and weight movement of the trained 1.3B checkpoints (seed 42) relative to the pretrained checkpoint. The nominal per-head memory length is $1/(\mathrm{softplus}(\Delta t \text{ bias}) \cdot e^{A_{\log}})$ tokens (48 layers × 64 heads; pretrained median 23.5 tokens); $\Delta \log_{10}$ is the mean over heads of its change and "heads shorter" the fraction of heads whose nominal memory shortened. Relative change is $\|\theta - \theta_{\mathrm{pre}}\|/\|\theta_{\mathrm{pre}}\|$ over all tensors of the group.

	nominal memory length		relative change of parameter group (%)			
arm	$\Delta \log_{10}$	heads shorter	A	Δt bias	in/out proj.	embedding
standard + chunked	−0.0043	0.98	0.34	0.30	1.47 / 1.49	0.77
standard + full backprop.	−0.0044	0.98	0.32	0.30	1.46 / 1.47	0.76
rare ∧ non-recent + chunked	−0.0043	0.98	0.34	0.30	1.37 / 1.39	0.90
rare ∧ non-recent + full backprop.*	−0.0043	0.98	0.32	0.30	1.36 / 1.38	0.90
rare-only + chunked	−0.0043	0.98	0.34	0.30	1.37 / 1.39	0.89

Table 15: Direction of the weight update $\theta - \theta_{\mathrm{pre}}$ over all parameters (1.3B): cosine similarity between the two updates, the norm of their difference relative to the first, and the ratio of their norms.

update pair (seed 42 unless noted)	cosine	rel. difference	norm ratio
standard + chunked vs. same, seed 43 (seed only)	0.991	0.13	1.00
standard: chunked vs. full backprop. (path only)	0.980	0.20	0.99
rare ∧ non-recent: chunked vs. full backprop.* (path only)	0.981	0.19	0.99
chunked: standard vs. rare ∧ non-recent (objective only)	0.836	0.57	1.00
full backprop.: standard vs. rare ∧ non-recent* (objective only)	0.837	0.57	1.00

Table 16: Peak GPU memory of one optimizer step (Mamba-2 1.3B, one 8,192-token sequence in 2,048-token chunks, micro-batch 1, fp32 master weights with bf16 autocast, AdamW) on one 48 GB A6000, steady state over steps 1–2 of 3. "Resident" is the allocation after the step; "activations" is peak minus resident.

configuration	peak (GB)	reserved (GB)	resident (GB)	activations (GB)	step (s)
chunked, per-chunk backward	37.5	38.2	15.5	22.0	2.3–3.5
full backprop. + checkpointing	26.6	27.8	16.6	10.0	1.6
full backprop., no checkpointing	40.6	41.8	16.6	24.0	1.3

A.5 Additional Details

Tasks. All probes are generated by the official RULER harness (Hsieh et al., 2024) at a pinned commit with its default task configurations and the GPT-NeoX-20B tokenizer; the haystacks are RULER's own, not PG-19. NIAH-s1: one needle (a word key with a seven-digit number value) at a random depth in a haystack of repeated noise sentences; the query asks for the number. NIAH-mk2: one needle among distractor needles (the haystack consists of other key–value pairs). VT: variable tracking with one chain of four hops ($X_1{=}v$, $X_2{=}X_1, \dots$) in a noise haystack; the query asks for all variables holding the value. CWE: ten common words, each appearing 30 times among uncommon words appearing 3 times; the query asks for the ten. FWE: words drawn from a Zipf-like distribution (exponent 2.0); the query asks for the three most frequent. The essay sibling root holds NIAH-s2 (one needle in Paul Graham essays) and NIAH-mk1 (four keys, essay haystack). Decoding is greedy with chunked prefill (128 new tokens for needle tasks, 30 for VT, 120 for CWE, 50 for FWE); scoring uses RULER's official evaluator, the fraction of expected answer strings present in the output.

Instruments. **Set A**: $n{=}100$ per task and length; lengths 1K–8K plus extrapolation extensions to 64K (NIAH-s1) and 16K (essay tasks). **Set B**: independently generated with a different seed, $n{=}200$, frozen before use. The primary metric is VT@8192; FWE is the declared cost metric; held-out perplexity uses 200 fixed PG-19 sequences.

Table 17: Training hyperparameters shared by every arm.

model	Mamba-2 1.3B (primary), 370M (scale); public checkpoints, fp32 master weights
data	PG-19 train split, one epoch, ≈3.07B tokens, sequences of 8,192 tokens
optimizer	AdamW, $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.1, constant lr 10^{-5}, no warmup
batch	128 sequences per step (2,933 steps per epoch), micro-batch 1 (370M: 2)
precision	fp32 parameters and optimizer state, bf16 autocast for activations
clipping	global gradient norm 1.0
chunking	chunk $C{=}2048$, state carried across chunks, detached between them
loss	$\sum_t w_t \, \mathrm{CE}_t / \sum_t w_t$ per 2,048-token chunk, averaged over the four chunks (the 370M full-backpropagation arms: over the whole sequence, §4)
mask	rare = complement of the 1,877-type prefix covering 80% of the token mass of a fixed 180-book (20.1M-token) prefix of the PG-19 training split; $W{=}256$
weight	$\beta = 1/\rho$ with ρ measured on the first 50 training sequences ($\mathbb{E}[w]{=}2$)
seeds	42, 43, 44 (data-order permutation and framework state)
hardware	8× NVIDIA RTX A6000 (48 GB), data-parallel

Gap-weighted family (Table 6). For a selected position t with a previous input occurrence of y_t at distance g_t (measured over the whole sequence), the weight is $w_t = 1 + \beta\,(g_t/W)^\gamma / N_\gamma$ with $N_\gamma = \mathbb{E}[(g_t/W)^\gamma \mid \text{selected, seen}]$ measured on the same 50 sequences as ρ; selected positions whose target has no previous occurrence in the sequence keep the flat weight $1 + \beta$. $\gamma{=}0$ recovers Eq. (1).

Code-path equivalence. With a carried state the Mamba-2 implementation takes an unfused code path, whereas the unchunked forward uses the fused kernel; the chunked and full arms were therefore trained through different kernels. Measured with the pretrained 1.3B weights: at every launch a probe compared the two forwards on a 4,096-token sequence (two chunks) in fp32 and aborted the run if the final-position logits differed by more than 0.1; the recorded maximum was $1.2\text{–}1.5 \times 10^{-2}$ for every 1.3B run and 7.6×10^{-3} for every 370M run. On one 8,192-token sequence (four chunks) in fp32 the two forwards agree to 1.0×10^{-2} at the final position and 0.29 at the worst position (logit magnitude ≈ 12), with 100% argmax agreement and identical cross-entropy (0.3631); on one 2,048-token chunk without carry the gradients computed through the two paths have cosine 1.000000 and relative difference 6×10^{-4}. Under the bf16 autocast used in training the forwards differ by up to 8.6 at the worst position with 99.3% argmax agreement (cross-entropy 0.3631 vs. 0.3642) and the one-chunk gradients have cosine 0.9992 and relative difference 5%. On the same sequence the four-chunk truncated gradient and the full-sequence gradient have cosine 0.85. Evaluation uses the chunked path for every arm. Throughput was 30k tokens/s (chunked) and 43.5k tokens/s (full with checkpointing); the chunked path uses more activation memory than full backpropagation with

checkpointing (Table 16), so the memory saving that motivates chunked training in general is not realized in this implementation.

Gradient decomposition. To separate the three implementation differences between the arms—the kernel path (cache path with a carried state vs. fused unchunked forward), the detach, and activation checkpointing—we computed the parameter gradient of one fixed 2,048-token window (four 512-token chunks, pretrained 1.3B weights, standard objective, per-chunk normalization) under four configurations: the fused unchunked forward; the cache path with the carried state left connected; the cache path with the state detached (the chunked arm); and the fused forward with and without checkpointing. Table 18 reports parameter-gradient cosines and relative differences over all parameters, on the training configuration and, as a cross-check, on the pure-PyTorch Mamba-2 path of a second machine. Connecting the carry without detaching reproduces the full gradient; checkpointing changes nothing; the detach alone accounts for the whole chunked-versus-full difference. The decomposition compares computation graphs, so a fixed random window suffices; the window (2,048 tokens in 512-token chunks) is shorter than the training configuration, so the result bounds the kernel-only discrepancy on that window and not at the training length or over a training run.

Table 18: Gradient decomposition (pretrained 1.3B, one fixed 2,048-token window in four 512-token chunks, standard objective). Cosine and relative difference of the parameter gradient over all parameters. "Training kernels": the cache path for the carried state and the fused kernel for the unchunked forward, as in training, on an A6000; "naive kernels": the pure-PyTorch Mamba-2 path on a second machine. Relative difference is $\|g_a - g_b\|/\|g_a\|$ with g_a the first-named gradient, the convention of our decomposition script; norm ratio is $\|g_b\|/\|g_a\|$.

	training kernels		naive kernels
comparison (what differs)	fp32: cos / rel. diff	bf16 autocast: cos / rel. diff	fp32: cos
kernel path only: connected carry vs. unchunked	1.000000 / 9×10^{-5}	0.99999 / 4×10^{-3}	1.000000
detach only: detached vs. connected carry	0.851 / 0.68	0.852 / 0.68	0.850
both: chunked arm vs. full arm	0.851 / 0.68	0.851 / 0.68	0.850
checkpointing on vs. off (unchunked, fp32)	1.000000 / 4×10^{-8}	—	1.000000 / 0

Normalization scope. The two gradient paths were implemented as separate code branches; the reweighted full-backpropagation arm at 1.3B was first trained with whole-sequence normalization, the difference was identified in a code review after the runs were complete, and the arm was retrained with per-chunk normalization, which is the run reported in the main tables. Per-chunk normalization (Eq. (2)) divides a chunk's weighted loss by the chunk's total weight $S_c = \sum_{t \in \mathcal{C}_c} w_t$; whole-sequence normalization divides by $\sum_c S_c$. The gradient of chunk c under the two differs by the factor $r_c = \bar{S}/S_c$, with $\bar{S}$ the mean chunk weight of the sequence. On the first 200 training sequences of each seed's order, with the mask of Eq. (1) and that seed's β, $|r_c - 1|$ has mean $0.036/0.031/0.030$ and 95th percentile $0.101/0.078/0.080$ for seeds 42/43/44 (range of r_c 0.83–1.24; 77–84% of chunks within 5%). The reweighted full-backpropagation arm re-trained with per-chunk normalization, otherwise identical, gives VT@8K $32.8/31.4/34.8$, NIAH@32K $65/65/64$, and held-out perplexity $12.24/12.11/12.12$ for seeds 42/43/44, against $32.8/32.2/33.6$, $65/65/65$, and $12.25/12.11/12.14$ for the whole-sequence runs (Table 8); the per-chunk runs are the ones reported in the main tables.

Seed-level intervals. Arms that share a seed consume the identical token stream in the identical order, so differences between arms are paired by seed. The intervals quoted in §5 are t intervals on the three per-seed differences (df $= 2$); with three seeds they are wide by construction and are quoted as bounds, not as evidence of equivalence. Pairing is a design choice that matches arms on the training seed and token order; it does not guarantee a narrower interval than an unpaired one, and the intervals characterize training-seed variability conditional on the fixed evaluation instances. Over every cell of the evaluation set in which a full-backpropagation arm and its chunked counterpart share three seeds (130 cells: both scales, both objectives, every task and length), the 95% interval lies above zero in 12 cells and below zero in 31; two of the 12 and four of the 31 have identical per-seed differences, where the interval is degenerate. The interpolation ledger rows for the seed-43 and seed-44 parents carry the literal seed 42 written by the interpolation script of the time; their

launch banners record parent checkpoints whose hashes differ from the seed-42 parents' and from each other.

Answer-token census. For the Set A instances of VT at 8K and NIAH-S1 at 32K, each reference answer was tokenized as a continuation of its prompt (input and answer prefix, as the evaluator builds it) and every answer token was classified by the two conditions of Eq. (1): outside $\mathcal{V}_\tau$, and absent from the $W{=}256$ input tokens preceding the answer. VT (100 instances, 500 answers, 1,641 tokens, 3.3 per answer): 58% of the answer tokens are outside $\mathcal{V}_\tau$, 94% are absent from the preceding 256 tokens, 54% satisfy both, and every answer token occurs somewhere earlier in the context; 99% of the answers contain at least one rare token and 93% contain no recent token. The non-rare answer tokens are single capital letters. NIAH-S1 at 32K (100 answers, seven-digit numbers, 3.2 tokens each): 78% of the tokens are outside $\mathcal{V}_\tau$ and none occurs in the preceding 256 tokens.

fp16 hazard. The public 370M–1.3B checkpoints ship in fp16; loading them as-is makes AdamW's ϵ and second-moment terms underflow and destroys training in one step. Our loader upcasts to fp32 master precision.

Provenance. Every run prints a launch banner echoing live configuration (form, mask, mode, β, data hash, first-sequence ids, git commit; runs launched after the normalization fix of §3 also echo the normalization scope); scored ledger rows carry the banner's git hash and the frozen evaluation-set id, so every number in the paper is traceable to a specific run.