AUDITING TEST-TIME TRAINING IN STREAMING 3D RECONSTRUCTION: SCALE-INVARIANT UPDATES, LARGE WEIGHT CHANGES, SMALL POSE EFFECTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming 3D reconstruction transformers increasingly incorporate test-time training (TTT) branches whose fast weights are updated online and are intended to retain long-horizon scene information. We audit this mechanism in the released Scal3R model under its published evaluation configuration on TUM RGB-D and 7-Scenes, explicitly tracing the implemented update rule and loss and evaluating interventions as paired differences relative to variation induced by window placement. We find that the deployed TTT update, inherited from large-chunk TTT and based on Muon-style orthogonalisation, is effectively scale-invariant: changing the base learning rate by $8\times$ alters absolute trajectory error (ATE) by at most 0.023%, while the approximately $4\times$ variation in per-token learning rates primarily reweights update directions. Although block gradients are highly redundant, orthogonalisation substantially magnifies their differences. Replacing the full gradient with that of a single block yields gradient cosine similarities of 0.77–0.99 but orthogonalised-update similarities of only 0.54–0.81, producing final fast weights that differ by 5–16%. Despite these large parameter-space differences, ATE remains within 0.14% of the full update across all tested 300-frame windows. More broadly, the fast weights undergo substantial adaptation, corresponding to an estimated 20°–34° parameter rotation, while their effect in trajectory space is small: removing the TTT branch or freezing its fast weights changes ATE by approximately 1% or less, and blockwise adaptation without retraining degrades ATE on two of three sequences, by as much as 26%. In contrast, a larger share of test-time variability originates elsewhere in the inference pipeline. Averaging predictions across two window offsets improves ATE on every sequence of both benchmarks, while ground-truth-free selection of the alignment back-end through self-consistency provides additional, though less universal, gains, reducing mean ATE by 9.9% on TUM and 8.3% on held-out 7-Scenes. These results reveal a pronounced mismatch between adaptation in parameter space and adaptation in reconstruction accuracy, and show that inference-window and alignment choices can dominate the observable contribution of the deployed TTT memory.

1 INTRODUCTION

Test-time training replaces a fixed layer with a small model whose parameters ("fast weights") are optimised on the current input during inference (Sun et al., 2020; 2024). Large-chunk variants accumulate gradients over large token blocks and orthogonalise the update with Muon-style Newton–Schulz iterations (Zhang et al., 2025; Jordan et al., 2024). Scal3R (Xie et al., 2026) brings this design to large-scale streaming 3D reconstruction on a VGGT-style backbone (Wang et al., 2025a), where the TTT layers are intended to carry scene context across blocks of frames.

Ablations of such layers usually compare a trained model with and without the layer, conflating two questions: does the extra branch help, and does updating it at test time help? For orthogonalised updates there is a third: which properties of the accumulated gradient can influence the update at all? We answer these for one released model with a measurement-first protocol (Section 4) whose noise floor is set by the placement of the evaluation window, the dominant source of variation in an otherwise deterministic pipeline.

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.

Thesis. We trace one chain in a deployed TTT memory: a known optimiser invariance makes the nominal learning-rate magnitude unidentifiable; block gradients are redundant, although the orthogonalised updates they produce are not; the resulting fast-weight change is large in parameter space but weak in trajectory space; and, consequently, most observable test-time variation lives elsewhere in the pipeline. The novelty is not the invariance, which is inherited from LaCT and Muon, but its consequences for a released 3D model and the intervention evidence that establishes them.

Pre-specified mechanistic prediction. If the update depends on the accumulated gradient only through its direction, subsampling block gradients should be harmless where they are concentrated and matter where they are spread. *The prediction fails if keeping only the first block's gradient deviates no more from the full accumulation on the sequences with the most spread gradients (`fr1_room`, `fr2_pioneer_slam`; effective rank up to* 4.9*) than on the most concentrated ones, or if changing the base learning rate produces an effect separable from the noise floor on any sequence.* Section 7 reports the outcome.

Contributions. (i) *Identifiability.* The released update and its loss in mathematical form, and what its inherited scale invariance (Remark 1) implies: the base rate, the block count and the gradient magnitude cannot act, and the per-token rates only reweight the update's direction (Corollary 1). (ii) *Redundancy.* Block gradients never disagree in sign on any tested sequence, but gradient redundancy does not imply update redundancy: the orthogonalised updates from the first block and from all blocks have cosine only 0.54–0.81. Gradient accumulation can nevertheless be aggressively subsampled with negligible trajectory change in the tested settings, because the output is insensitive to the fast weights (Fig. 2). (iii) *Parameter-space versus trajectory-space change.* Disable modes that separate the branch from its adaptation, per-sequence paired statistics, per-layer additivity, update variants and a direct estimate of the weight change show that a large change of the fast weights moves the trajectory by about one percent or less. (iv) *Where the variation lives.* As a system-level consequence, window averaging and ground-truth-free selection of the alignment back-end exploit the test-time variation that the fast weights do not capture (Section 9). All results rest on a measurement protocol for a deterministic streaming pipeline (window-offset noise floor, paired comparisons, separability bar, metric validation, execution checks), and we report which of our own earlier claims did not survive measurement (Appendix D).

2 Related work

TTT was introduced as self-supervised adaptation at inference (Sun et al., 2020) and recast as sequence layers whose hidden state is a model trained online (Sun et al., 2024), related to fast-weight programmers (Schlag et al., 2021) and learned memories (Behrouz et al., 2024). Large-chunk TTT (LaCT) (Zhang et al., 2025) updates fast weights once per chunk with the negative dot-product loss, per-token rates inside the chunk gradient, Muon-style orthogonalisation (Jordan et al., 2024) and L2 weight normalisation, with no learning rate after orthogonalisation. Scal3R inherits this update. The scale invariance that follows is a known property of Frobenius-pre-normalised Newton–Schulz iterations (Jordan et al., 2024; Bernstein & Newhouse, 2024), whose step magnitude is otherwise set by an explicit rescaling (Liu et al., 2025), and of weight-normalised parameters (Salimans & Kingma, 2016; van Laarhoven, 2017; Arora et al., 2019); we do not claim it as new, but measure its consequences in a released 3D model. Feed-forward multi-view models such as VGGT (Wang et al., 2025a) and recurrent models with persistent state such as CUT3R (Wang et al., 2025b) predict geometry directly from images; Scal3R adds TTT layers and a loop-closure back-end with a SALAD descriptor (Izquierdo & Civera, 2024). TTT3R (Chen et al., 2025) and ZipMap (Jin et al., 2026) also cast streaming 3D reconstruction as test-time training of a persistent state. We audit only Scal3R's TTT component; whether the updates of these related models are likewise magnitude-free is a direct next test of our analysis.

3 The system under study

Scal3R processes a sequence in overlapping blocks (60 frames, overlap 30; 9 blocks for 300 frames). At four global-attention layers ($\ell \in \{14, 17, 20, 23\}$, width 1024) a SwiGLU fast-weight module

$f_W(q) = (\mathrm{silu}(qW_0) \odot qW_2)W_1$ is attached. Algorithm 1 states the released inference path in mathematical form; the corresponding implementation locations are listed in Appendix A.

Algorithm 1 Released TTT update at one layer (inference)

1: $\eta_t \leftarrow \mathrm{softplus}(w^\top x_t + b + c)$, $c = \mathrm{softplus}^{-1}(\eta_{\text{base}})$ ▷ per-token rate, $\eta_{\text{base}} = 0.01$
2: **for** each block $b = 1, \ldots, B$ **do**
3: $\quad g_b \leftarrow \sum_{t \in b} \eta_t \nabla_W \mathcal{L}_t(W^{\text{pre}})$ ▷ rates weight tokens inside the gradient
4: **end for**
5: $G \leftarrow \sum_{b=1}^{B} g_b$ ▷ sum over blocks
6: $X_0 \leftarrow G/(\|G\|_F + 10^{-7})$; $X_{i+1} \leftarrow aX_i + (bX_iX_i^\top + c'(X_iX_i^\top)^2)X_i$, $i < 5$ ▷ Newton–Schulz, bf16
7: $W' \leftarrow W^{\text{pre}} - X_5$ ▷ no step size after orthogonalisation
8: $W'_{:,j} \leftarrow W'_{:,j}\, \|W^{\text{pre}}_{:,j}\|/(\|W'_{:,j}\| + 10^{-5})$ ▷ restore per-column norms
9: apply $f_{W'}$ to every block

Inner loss and inputs. The fast weights are fitted to the negative dot-product loss of LaCT (Zhang et al., 2025), $\mathcal{L}_t(W) = -f_W(k_t)^\top v_t$ with $f_W(k) = (\mathrm{silu}(kW_0) \odot kW_2)W_1$. The query, key and value q_t, k_t, v_t come from the attention block's own qkv projection of token x_t, followed by per-head query/key normalisation and rotary position embedding; $f_{W'}$ is applied to q_t. The rate η_t is computed from the input token x_t (not from q_t or k_t), with one rate per head and per fast-weight matrix. Algorithm 1 is written in descent form, $W' = \mathrm{normalise}(W - \mathrm{NS}(G))$ column-wise; the implementation adds the orthogonalised ascent direction, which is identical because the Newton–Schulz polynomial is odd, $\mathrm{NS}(-G) = -\mathrm{NS}(G)$.

The rates enter only as token weights inside each block gradient; no rate multiplies the orthogonalised update; and the iteration starts from the Frobenius-normalised input (which bounds, but is not, the spectral normalisation). Let $U(G; W)$ denote lines 6–8 in exact arithmetic with the stabilising constants set to zero.

Remark 1 (Scale invariance, inherited from LaCT/Muon)**.** *For any $\alpha > 0$, $U(\alpha G; W) = U(G; W)$. This is the scale invariance of the Frobenius-pre-normalised Newton–Schulz iteration used by Muon and LaCT (Jordan et al., 2024; Zhang et al., 2025), combined with the scale invariance of weight normalisation (Salimans & Kingma, 2016); we state it only to fix notation.*

Proof. $X_0(\alpha G) = \alpha G/\|\alpha G\|_F = X_0(G)$; the recursion is a deterministic function of X_0, and lines 7–8 depend on G only through X_5. □

Assumptions. Remark 1 and Proposition 1 hold in exact arithmetic with the stabilising constants set to zero; the implementation runs Newton–Schulz in bf16 with constants 10^{-7} and 10^{-5}, so measured invariances are approximate, not bitwise. All assumptions and the proof of Corollary 1 are in Appendix B. Remark 1 removes three candidate step sizes: the gradient magnitude, a uniform factor on all rates, and the block count B when block gradients are parallel. Non-uniform rates are not cancelled: they change the relative weight of tokens in G, hence its direction, but not the length of the step, which is fixed by the Newton–Schulz output and the norm restoration.

Corollary 1 (Base learning rate)**.** *With $\eta_t = \mathrm{softplus}(u_t + c)$, changing η_{base} (i.e. $c \to c'$) multiplies each rate by $r_t = e^{c'-c}\,\phi(u_t + c')/\phi(u_t + c)$, $\phi(z) = \mathrm{softplus}(z)e^{-z} = 1 - \frac{1}{2}e^z + O(e^{2z})$. The uniform factor $e^{c'-c}$ cancels by Remark 1; only the token-dependent residual $\approx \frac{1}{2}(e^{u_t+c} - e^{u_t+c'})$, of the order of half the rates themselves ($c \approx -4.6$ at $\eta_{base} = 0.01$), can change the update. No logged token leaves this regime: $u_t + c < 0$ for every token at all tested base rates (Appendix B).*

Proposition 1 (Aligned block gradients)**.** *If $g_b = a_b u + e_b$ with $\|u\|_F = 1$, $a_b > 0$, $\|e_b\|_F \le \varepsilon a_b$ for all b and some $\varepsilon < 1$, then for every non-empty subset S, $\sin\angle(\sum_{b \in S} g_b, u) \le \varepsilon$.*

Proof. Let $A = \sum_S a_b > 0$. Then $\sum_S g_b = A(u + e)$ with $e = \sum_S e_b/A$ and $\|e\|_F \le \varepsilon < 1$. Every vector in the ball of radius ε around the unit vector u makes an angle of at most $\arcsin\varepsilon$ with u (the tangent from the origin to that ball), so $\sin\angle(\sum_S g_b, u) \le \varepsilon$. □

Proposition 1 bounds the angle between block-gradient *sums*, not between the Newton–Schulz outputs they produce. The orthogonalisation is not Lipschitz near rank deficiency: it amplifies small

singular directions, so a small residual e can become an $O(1)$ component of X_5. Aligned block gradients therefore do not guarantee that a subset of blocks yields nearly the same update, and measurement shows that it does not (Fig. 2C); the first-block result rests on the measurement in Appendix Table 10, not on Proposition 1. The alignment account motivates the pre-specified mechanistic prediction stated in the introduction.

4 MEASUREMENT PROTOCOL

What is measured. The released Scal3R code and checkpoint with all inference defaults (base rate 0.01, five Newton–Schulz steps, 518-pixel processing; identifiers in Appendix A) on TUM RGB-D (Sturm et al., 2012). The in-depth study uses the first 300 frames of `fr3_long_office_household`; replication uses four further sequences (Section 7). Inference is deterministic within a container; repeating the released configuration on a different GPU container changed ATE by at most 1.4×10^{-5} relative (`fr2_desk`; $\leq 1.3 \times 10^{-6}$ on `fr3_long_office`), which bounds hardware-level variation far below every effect we report.

Metric and its blind spots. ATE is the RMSE of camera positions after Sim(3) alignment (Umeyama, 1991), computed on the frames retained after Scal3R's overlap de-duplication (we record the retained indices, which the release does not export). By construction ATE is blind to global scale (removed by the alignment), to local drift that averages out over the trajectory, to depth and point-map quality (measured separately only in Appendix G), and to discarded frames. Our conclusions concern camera trajectories only.

Metric validation. Two configurations known to damage the backbone rank worst: skipping global attention in layers 10–15 ($+62.2\%$) and processing at 630 pixels ($+112.1\%$).

Nuisance factors and noise floor. No random seed exists, but the measurement moves with the placement of the evaluation window. Shifting the 300-frame window by $0, 5, 10, 15, 20$ frames gives released-model ATE $0.03416, 0.03325, 0.03659, 0.03998, 0.03075$ m: mean 0.03495, sd 0.00351 (relative sd 10.0%, the dispersion statistic we use throughout). The second nuisance factor is the sequence itself.

Comparison, separability and pre-registration. Because the floor dwarfs the effects of interest, every intervention is compared with the released model *on the same window* (paired), and we report the mean paired difference, its sd and $|d|/\mathrm{sd}$ over offsets, requiring $|d|/\mathrm{sd} \geq 2$ before an effect is called separable. This bar and the offset set were fixed by our audit protocol before interventions were scored; the mechanistic prediction was fixed before its runs completed. The sequence set and the number of offsets per sequence were not pre-registered, and all completed runs are reported.

Interventions and execution check. *Branch*: the TTT branch is skipped entirely. *Zero*: the full pathway runs (compute-matched) but its residual is multiplied by zero. *Frozen*: the branch runs with the pretrained fast weights and no test-time step. *Block subsampling*: only every s-th block gradient (or the first only) enters G. *Configuration*: base rate, Newton–Schulz steps and precision. All switches are off by default, and the patched pipeline then reproduces the released output. The execution check is judged against the container-to-container variation above: every branch, frozen-weight, block-subsampling and residual-gain cell exceeds that bound, while 4 of the 18 cross-sequence base-rate cells fall inside it, which is the outcome the theory predicts for a switch that cannot act; in addition, the update-variant switches of Section 8 write a per-run marker from inside the modified code path. Runtimes are single-A100 wall-clock seconds and indicative (single runs).

Scope. One model and checkpoint; two indoor benchmarks (five TUM RGB-D sequences, one also at 1,200 frames, and four 7-Scenes scenes (Shotton et al., 2013)); no retraining. The full setup is listed in Appendix C.

Evaluation versus training configuration. Scal3R's own evaluation uses chunks of 60 frames with overlap 30 on all datasets and includes TUM RGB-D as a dense long-video benchmark (Xie et al., 2026); our inference setting is identical. Training used variable effective sequence lengths of 1 to 32 chunks: our 300-frame windows (9 chunks) lie inside that range, whereas the 1,200-frame runs (41 chunks) exceed it. The audit is therefore carried out at the authors' released evaluation configuration, which answers part of the concern that the branch is evaluated outside its operating regime. A mismatch in unreported training details, such as the training resolution, cannot be excluded.

5 RESULTS: WHAT THE TTT LAYER DOES

Step-size knobs. An eightfold change of the base rate moves ATE by at most 0.023% (Appendix Table 11), of the same order as switching the iteration to fp32 (0.009%) and more than two orders of magnitude below the window-placement sd. All changes are $\leq 0.023\%$, consistent with the small token-dependent residual of Corollary 1; the corollary predicts a weight-space residual, not the sign of the ATE change. The number of Newton–Schulz steps, which changes the shape rather than the scale of the update, has a larger but still sub-percent effect.

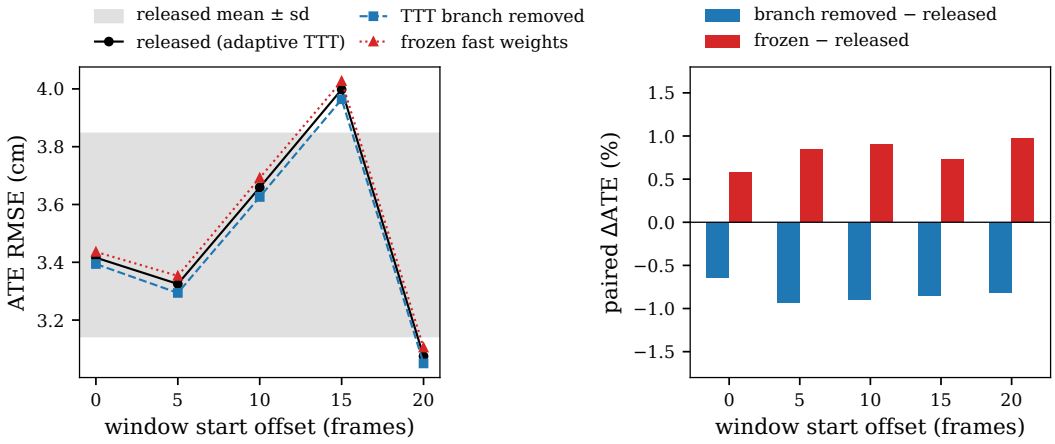


Figure 1: Window-offset effect versus TTT interventions on `fr3_long_office` (300 frames). Left: ATE (cm) at five window start offsets for the released model, the model without the TTT branch, and the model with frozen fast weights; the grey band is the released model's mean $\pm$ one sd over the five offsets (3.50 ± 0.35 cm). Right: paired difference (%) of each intervention against the released model at the same offset.

Table 1: Disable modes at five window offsets (`fr3_long_office`). ATE in metres; paired Δ relative to the released model at the same offset. *Zero* (compute-matched) is identical to *branch* in every cell where both were run (Table 13). Released ATE over the five offsets: mean 0.03495 m, sd 0.00351 m; the last row gives mean $\pm$ sd of the paired differences over offsets.

Offset	Released	Branch removed	Frozen fast weights	Adaptation gain (frozen − released)
0	0.03416	0.03394 (-0.64%)	0.03436 ($+0.59\%$)	$+0.59\%$
5	0.03325	0.03294 (-0.93%)	0.03353 ($+0.84\%$)	$+0.84\%$
10	0.03659	0.03626 (-0.90%)	0.03692 ($+0.90\%$)	$+0.90\%$
15	0.03998	0.03964 (-0.85%)	0.04027 ($+0.73\%$)	$+0.73\%$
20	0.03075	0.03050 (-0.81%)	0.03105 ($+0.98\%$)	$+0.98\%$
mean $\pm$ sd		$-0.83 \pm 0.11\%$, $\lvert d\rvert/\mathrm{sd} = 7.34$	$+0.81 \pm 0.15\%$, $\lvert d\rvert/\mathrm{sd} = 5.25$	

Branch versus adaptation. At every offset of this sequence the adapted fast weights beat the frozen ones ($+0.81 \pm 0.15\%$ for frozen, $|d|/\mathrm{sd} = 5.25$): here the test-time step carries information. At every offset of this sequence, however, removing the branch lowers ATE further ($-0.83 \pm 0.11\%$, $|d|/\mathrm{sd} = 7.34$; Table 1, Figure 1). Both effects clear the separability bar in paired form while being about ten times smaller than the window-placement sd. The compute-matched *zero* control is bitwise identical to *branch*, so the difference is due to the branch's residual contribution, not to changes in numerical scheduling. On this sequence the adaptation thus recovers part of a small cost introduced by the branch; whether the branch's net effect keeps its sign on other sequences is examined in Section 7.

6 MECHANISM: WHAT THE UPDATE CAN AND CANNOT CHANGE

6.1 GEOMETRY OF THE PER-BLOCK GRADIENTS

On `fr3_long_office`, all 4×36 block pairs have positive cosine similarity; alignment is strongest at layers 14 and 23 (minimum cosine $\geq$ 0.84) and weakest at 17 and 20 (0.60–0.64; Appendix Figure 3). Across sequences (Fig. 2A, Appendix Table 7) two properties hold everywhere: every pairwise cosine is positive (minimum 0.07, `fr1_room` layer 17), and the most concentrated layer is always 14 or 23. Beyond that, concentration is scene- and layer-dependent: effective rank reaches 4.89, and on the 1,200-frame window it rises at layers 17–23 relative to 300 frames of the same scene. The premise of Proposition 1 thus holds with small ε on `fr3_long_office` and `fr2_desk` and only loosely elsewhere. The effective rank describes the set of block gradients, not the rank of the fast weights.

6.2 BLOCK SUBSAMPLING

One block's gradient reproduces the full update's ATE: on `fr3_long_office` (300 frames, offset 0), strides 2 and 3 leave ATE unchanged and keeping one of nine block gradients changes it by 0.06% (Appendix Table 10). Runtimes are single timed runs; stride 3 and first-block-only both took 354 s, so these timings do not resolve the per-block cost, and the 12% (first block) and 16% (branch removed) differences are indicative (single runs).

Per-layer contributions are additive. Disabling TTT at one layer at a time on `fr3_long_office` (Appendix Table 18), the four single-layer skips sum to -0.67%, against -0.64% for removing the whole branch. Predicting each "only layer ℓ active" configuration as the sum of the skips of the other three layers gives -0.62 vs. observed -0.59% (only 14), -0.41 vs. -0.40% (only 17), -0.31 vs. -0.31% (only 20) and -0.67 vs. -0.65% (only 23). Layer 23 is inert (skipping it changes ATE by 0.00%), and layer 20 carries the largest share of the branch's effect.

Gradient redundancy is not update redundancy. A read-only probe that ran the released update unchanged (ATE identical to a released rerun) shows that Newton–Schulz amplifies the small differences between the first-block and full gradients. Over nine sequences and four layers (averaged over W_0, W_1, W_2), the gradient cosine is 0.77–0.99, the cosine between the orthogonalised updates only 0.54–0.81, lower in every cell, and the final fast weights after norm restoration differ by 5–16% in relative Frobenius norm (Fig. 2B–C; `fr1_room`, layer 20: 0.77, 0.54, 16%). Yet the trajectory changes by at most 0.11% at 300 frames, and gradient concentration does not predict that change (Fig. 2D; Spearman -0.26 over ten settings). The first-block result therefore reflects the output's insensitivity to the fast weights (Section 6.3), not nearly identical updates.

6.3 HOW LARGE IS THE WEIGHT CHANGE?

Because Newton–Schulz fixes the scale of its output regardless of G, the size of the released weight change can be estimated without the scene: we apply the released update, $W' = \mathrm{normalise}(W + s\,\mathrm{NS}(G))$ with column-norm restoration and the released five-step bf16 iteration, to the pretrained fast weights of all four TTT layers ($W_0, W_2 \in \mathbb{R}^{1024\times4096}$, $W_1 \in \mathbb{R}^{4096\times1024}$), using a random-direction G. The magnitude is representative; the direction is not, so this estimates only the size of the change. Results are identical across layers to two decimals. Pretrained column norms are about 1.34–1.44; the Newton–Schulz output has column norm 0.48 (W_0, W_2) and 0.97 (W_1), i.e. $0.36\times$ and 0.67–$0.70\times$ the pretrained norms. At the released step ($s = 1$) the updated weights have cosine 0.94 (W_0, W_2) and 0.82–0.83 (W_1) to the pretrained ones, a rotation of roughly 20° and 34°; at $s = 0.25$ the cosines are 0.996 and 0.986, at $s = 2$ they are 0.81 and 0.58–0.60.

The update is therefore not small in weight space, yet the reconstruction changes by about one percent or less (Tables 1 and 13), and a sixteenfold change of the step (s from 0.25 to 2, Table 12) changes ATE by at most 0.34%. The weakness lies in the insensitivity of the output to the fast-weight values, not in a tiny update; this agrees with the residual-gain dose-response (Section 8), where the branch's effect is set by the magnitude of its residual.

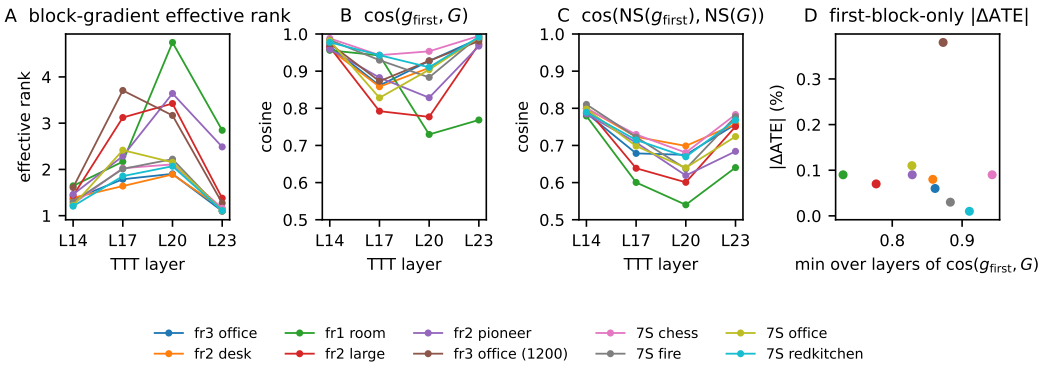


Figure 2: Gradient redundancy does not imply update redundancy (released model, offset 0, 300 frames unless marked). **A**: effective rank of the per-block gradient Gram matrix at each TTT layer. **B**: cosine between the first block's gradient and the full accumulated gradient G (all three fast-weight matrices concatenated), computed exactly from the logged block Gram matrices and norms. **C**: cosine between the Newton–Schulz outputs of the first-block and full gradients, measured by a read-only probe running the released update unchanged (averaged over W_0, W_1, W_2); it is lower than the gradient cosine in every cell. **D**: observed change of ATE when only the first block's gradient is used, against the smallest gradient cosine over layers (Spearman -0.26 over ten settings). The updates differ substantially, yet the trajectory barely changes.

6.4 The learned per-token rates

A read-only hook (which leaves ATE unchanged) records the rates η_t. On `fr3_long_office` the median coefficient of variation across tokens is 0.45 and the median p95/p5 ratio is 4.08 (`fr2_desk`: 0.51 and 4.51). The update discards only their uniform component; their variation across tokens survives as a reweighting of the update's direction.

Duplicating the frames of one half of the window $k = 1$–5 times leaves ATE and depth error on the untouched half without systematic change (Appendix G).

7 Generality

Appendix Table 13 repeats the interventions on four further sequences at two offsets; Table 2 summarises them with the sequence as the unit (a descriptive pooling over cells is in Appendix Table 14). By Corollary 1 the base-rate invariance should transfer independently of the scene. The *branch* and *zero* columns coincide in every cell where both were run. The direction of the branch-removal effect is not uniform across sequences (Table 13): the ATE reduction on `fr3_long_office` does not generalise to every scene, so we make no general claim about it.

Per-sequence statistics. With the sequence as the unit (Table 2), the branch-removal effect is negative in the mean of all nine sequences (median -0.23%, mean -0.30%; sign test over sequences $p = 0.004$), but it is below 0.3% in seven of them and not consistent across the windows of a sequence: at the level of individual windows, excluding `fr3_long_office`, removal is negative in 6 of 8 cells (median -0.24%, $p = 0.29$). The frozen-weight effect has no consistent sign across sequences (positive in 5 of 9 sequence means, $p = 1.0$): adaptation beats frozen weights on `fr3_long_office` and `fr1_room`, while on 7-Scenes freezing lowered ATE in three of four scenes. The sequence means rest on different numbers of windows (5 for `fr3_long_office`, 2 for the other TUM sequences, 1 for each 7-Scenes scene). The pooled (sequence, offset) cell statistics are descriptive only (Appendix Table 14).

Outcome of the pre-specified mechanistic prediction. The *1 block* column is the prediction stated in the introduction: it fails if its deviation on `fr1_room` ($+0.09$, $+0.11$) and `fr2_pioneer_slam` (-0.09, $+0.14$) is not larger than on `fr2_desk` ($+0.08$, $+0.07$) and `fr3_long_office` ($+0.06$), or if any base-rate cell departs from zero by more than the offset floor. *Verdict.* The mean devi-

Table 2: Per-sequence mean paired Δ ATE (%) against the released model on the same window; the sequence is the statistical unit. Offsets: number of window offsets averaged per sequence (7-Scenes: offset 0 only). Summary rows: mean (sd) over sequences.

Sequence	Offsets	Branch removed	Frozen fast weights
`fr3_long_office`	5	-0.83	$+0.81$
`fr1_room`	2	-0.59	$+0.99$
`fr2_desk`	2	-0.10	$+0.32$
`fr2_large_no_loop`	2	-0.20	$+0.01$
`fr2_pioneer_slam`	2	-0.16	-0.29
7-Scenes *chess*	1	-0.23	-0.36
7-Scenes *fire*	1	-0.26	$+0.25$
7-Scenes *office*	1	-0.29	-0.11
7-Scenes *redkitchen*	1	-0.04	-0.23
TUM (5 sequences)		-0.38 (0.32)	$+0.37$ (0.54)
7-Scenes (4 scenes)		-0.21 (0.11)	-0.11 (0.26)
All 9 sequences		-0.30 (0.25)	$+0.15$ (0.48)

ations are ordered as predicted (0.10–0.12% on the two spread sequences vs. 0.06–0.08% on the concentrated ones), but every deviation is at most 0.14%, far below the window-placement sd, so the prediction neither confirms nor refutes the alignment account; one block's gradient suffices even where gradients are least concentrated, although the orthogonalised updates differ (Fig. 2C). All base-rate cells stay within $\pm 0.023\%$.

8 Can the update be made to adapt?

We tested seven variants that re-introduce a step size or update the fast weights block by block, and a gain on the branch's residual, on three sequences (Appendix Table 12).

Rescaling the orthogonalised step moves ATE by at most 0.34%, and the residual gain moves it almost linearly, so the branch's contribution is set largely by the magnitude of its residual. Only block-by-block (online) adaptation has a large effect: the causal online update raises ATE by 2.3–7.8%, and applying the final online weights to every block gives -3.9%, $+26.0\%$ and $+13.2\%$ on `fr3_long_office`, `fr2_desk` and `fr1_room`, so without retraining such adaptation is unsafe.

9 Test-time averaging and self-consistency selection

Once the fast-weight mechanism was found to have about one percent or less influence on the trajectory, we asked where test-time variability actually resides. Two sources dominate: the placement of the evaluation window (relative sd 10.0%, Section 4) and the alignment back-end that joins blocks, whose largest levers change sign between scenes (the robust-loss rule gives -33% on `fr3_long_office` and $+18\%$ on `fr2_desk`). We test two ground-truth-free procedures: averaging the trajectories of two window offsets (the robust finding) and selecting the back-end per scene by self-consistency (a further, less universal gain).

Procedure. Each candidate back-end configuration is run on the same sequence at two window offsets (0 and 10 frames). The *self-consistency* of a configuration is the RMS distance between its two trajectories on shared frames after a Sim(3) fit of one onto the other, divided by the RMS radius of the reference trajectory, so it is scale-free and uses no ground truth. The most self-consistent configuration is selected and its two trajectories averaged; ground truth is used only for scoring, against the released model's mean two-offset ATE on shared frames. The candidates are the released model and four back-end settings from our audit: a residual-scaled robust loss (*adaptive Huber*), point-map alignment, reduced overlap, and fused heads.

Result. Averaging the released model's two offsets alone lowers ATE on every sequence of both benchmarks (-0.7 to -10.0%). Adding selection lowers ATE on every TUM sequence (select-then-average -2.3 to -26.2%, mean -9.9%); on held-out 7-Scenes it helps three of four scenes but raises *office* by 2.7%.

Table 3: Ground-truth-free selection: Δ ATE (%) against the released model's expected single-window ATE. *Averaging*: released model, two offsets averaged; *Select*: most self-consistent configuration; *Select+avg*: its two offsets averaged. †candidates identified on this sequence; 7-Scenes is held out (per scene: Appendix K).

Sequence	Selected	Averaging	Select	Select+avg
`fr3_long_office`†	adaptive Huber	-6.59	-20.15	-26.22
`fr2_desk`	released	-9.96	$+0.00$	-9.96
`fr1_room`	released	-2.26	$+0.00$	-2.26
`fr2_large_no_loop`	adaptive Huber	-4.79	-1.39	-5.31
`fr2_pioneer_slam`	adaptive Huber	-2.70	-3.81	-5.54
TUM mean (worst)		-5.26 (-2.26)	-5.07 ($+0.00$)	-9.86 (-2.26)
7-Scenes mean (worst)		-2.99 (-0.72)	-5.15 ($+3.18$)	-7.57 ($+2.66$)

Cost and a held-out rule. Five candidates cost ten reconstructions; a two-candidate rule (released model and adaptive Huber, four runs), stated before any 7-Scenes candidate run, reproduces TUM exactly and gives a held-out 7-Scenes mean of -8.26%.

Averaging versus selection, at equal compute. On TUM outside the design sequence, averaging two offsets carries about 85% of the gain (-4.93 of -5.77%; Appendix Table 15). Against an equal-compute baseline that averages the released model over four offsets, the four-run select-then-average wins on four of five TUM sequences but loses on `fr2_pioneer_slam`, and excluding `fr3_long_office` its advantage is small (-5.88 vs. -5.03%; Appendix Table 16). Averaging more windows is not reliably better: four offsets equal two on average and worsen `fr1_room` ($+2.18\%$). On 7-Scenes selection contributes more than averaging (-4.58 vs. -2.99%); the equal-compute test there was not run.

10 Implications for TTT layers and their evaluation

(1) If an orthogonalised update is used, some quantity must set its magnitude after orthogonalisation (a learned or scheduled step size); otherwise learning rates, block counts and gradient norms are unidentifiable at test time (Remark 1, Table 11). Here, adding such a step or updating block by block at inference did not help consistently (Section 8). (2) Ablations should include *frozen* and compute-matched *zero* controls, not only removal of the layer. (3) Deterministic pipelines still need a noise floor: window placement alone gives a relative ATE sd of 10.0% (Section 4).

11 Threats to validity and limitations

Internal. Offsets are windows of one recording, not independent draws; cross-sequence cells have one or two offsets each. **Construct.** ATE covers camera trajectories only. **External.** One released model on two indoor benchmarks; long-horizon runs are only 1,200 frames; no claim about TTT layers in general or retrained models. Revised claims: Appendix D.

12 Conclusion

The nominal TTT learning-rate magnitude of the released Scal3R system is structurally unidentifiable, its block gradients are redundant (its orthogonalised updates are not), and its fast-weight change, though large in weight space, moves the trajectory by about one percent or less (Tables 1, 2, 17); block-by-block adaptation without retraining degrades ATE by up to 26% (Section 8). Test-time variation lives elsewhere: window averaging lowers ATE on every sequence, and back-end selection adds a smaller, less universal gain (Tables 3, 16).

REPRODUCIBILITY STATEMENT

Model, checkpoint, code revision and all intervention switches are identified in Appendix A; every reported number is mapped to its logged artefact in Appendix E. The patch is inert when its switches are unset. Proofs are in Section 3 and Appendix B, with all assumptions stated; the complete experimental setup is in Appendix C. Scripts for all interventions, the gradient probe, the frame-index bookkeeping, the statistics and the figures will be released.

ETHICS STATEMENT

This work analyses a publicly released reconstruction model on a public benchmark without human subjects or personal data. We see no specific ethical concerns.

AI USE STATEMENT

We used generative AI tools to assist with drafting and editing portions of the manuscript and with searching and summarising experiment logs. Every reported numerical result was independently checked against the original logged outputs, and all derivations and scientific claims were verified by the authors. The authors take full responsibility for the content of this paper.

REFERENCES

Sanjeev Arora, Zhiyuan Li, and Kaifeng Lyu. Theoretical analysis of auto rate-tuning by batch normalization. In *International Conference on Learning Representations (ICLR)*, 2019.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *arXiv preprint arXiv:2501.00663*, 2024.

Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. *arXiv preprint arXiv:2409.20325*, 2024.

Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. TTT3R: 3D reconstruction as test-time training. *arXiv preprint arXiv:2509.26645*, 2025.

Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Haian Jin, Rundi Wu, Tianyuan Zhang, Ruiqi Gao, Jonathan T. Barron, Noah Snavely, and Aleksander Holynski. ZipMap: Linear-time stateful 3D reconstruction via test-time training. *arXiv preprint arXiv:2603.04385*, 2026.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. `https://kellerjordan.github.io/posts/muon/`, 2024.

Jingyuan Liu et al. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025.

Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning (ICML)*, 2021.

Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning (ICML)*, 2020.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (learn at test time): RNNs with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

Twan van Laarhoven. L2 regularization versus batch and weight normalization. *arXiv preprint arXiv:1706.05350*, 2017.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3D perception model with persistent state. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b.

Tao Xie, Peishan Yang, Yudong Jin, Yingfeng Cai, Wei Yin, Weiqiang Ren, Qian Zhang, Wei Hua, Sida Peng, Xiaoyang Guo, and Xiaowei Zhou. Scal3R: Scalable test-time training for large-scale 3D reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2604.08542.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William T. Freeman, and Hao Tan. Test-time training done right. *arXiv preprint arXiv:2505.23884*, 2025.

A Reproducibility details and implementation identifiers

Released artefacts. Code: Scal3R, commit `26a2fa73898440fefe056a80dfb08763e5b81314`. Checkpoint: `xbillowy/Scal3R` (`scal3r.pt`). Configuration: `base_lr`=0.01, `muon_update_steps`=5, block size 60, overlap 30.

Implementation locations of Algorithm 1. Per-token rates: `softplus(lrs(x) + inv_base_lr)` in `scal3r/utils/ttt_utils.py`. Per-block gradients and their running sum: `apply_ttt` in `scal3r/pipelines/backend.py` (`w0_grad_sum.add_(g0)`). Orthogonalisation: `zeropower_via_newtonschulz5`, whose Frobenius normalisation is `X / (X.norm(dim=(1,2)) + 1e-7)` (`ttt_utils.py`, line 113), computed in bf16 unless fp32 is requested. The update call passes `lr=None`, so no scaling follows orthogonalisation; column norms are then restored.

Intervention switches (environment variables added by our patch; all unset reproduces the release): `SCAL3R_TTT_DISABLE=branch|zero|update` (branch / zero / frozen); `SCAL3R_TTT_CHUNK_STRIDE=`s ($s = 99$: first block only); `SCAL3R_NS_FP32=1`; configuration overrides of `base_lr` and `muon_update_steps`; `SCAL3R_SKIP_GLOBAL` (metric validation); processing size 630 (metric validation). Window offsets drop the first o frames of the sequence before taking 300. The gradient probe records each per-block gradient before summation and the retained frame order after de-duplication.

B Assumptions and proofs

Assumptions. Remark 1 and Proposition 1 are statements in exact arithmetic: the stabilising constants 10^{-7} (Newton–Schulz normalisation) and 10^{-5} (norm restoration) are set to zero and the iteration is carried out in real numbers. The implementation runs Newton–Schulz in bf16 and keeps

both constants, so the identities hold only approximately; the measured invariances below are approximate, not bitwise. Proposition 1 assumes $\varepsilon < 1$ alignment of the block gradients, which Section 6.1 measures rather than assumes. Corollary 1 assumes $u_t + c \leq 0$ and $u_t + c' \leq 0$ for all tokens, i.e. rates in the exponential regime of the softplus (true at $\eta_{\text{base}} = 0.01$, $c \approx -4.6$, unless a token's rate logit exceeds $+4.6$). *Evidence.* A dedicated read-only run logged, over every token and rate channel, the maximum logit $u_t + c$ and the fraction of positive logits at $\eta_{\text{base}} \in \{0.005, 0.01, 0.04\}$ on `fr3_long_office` and `fr2_desk`. The maximum is -2.63 and -2.40 at 0.005, -1.93 and -1.71 at 0.01 (released), and -0.53 and -0.31 at 0.04; the fraction of positive logits is exactly 0 in all six runs, and the median per-token CV is unchanged by the base rate (0.45 and 0.50–0.51). The assumption therefore holds for every token at every tested base rate. At 0.04 the most extreme tokens approach the boundary (maximum -0.31), so the corollary's residual term is largest there; this remains consistent with the measured base-rate effect of at most 0.023%.

Proof of Corollary 1. For $z \leq 0$, $\text{softplus}(z) = \log(1 + e^z) = e^z \phi(z)$ with $\phi(z) = \sum_{k \geq 1} (-1)^{k+1} e^{(k-1)z}/k$, an alternating series with non-increasing terms, so $1 - \frac{1}{2}e^z \leq \phi(z) \leq 1$. Hence $\eta_t'/\eta_t = e^{c'-c}\phi(u_t + c')/\phi(u_t + c)$, and with $z = u_t + c$, $z' = u_t + c'$ the token-dependent factor satisfies $1 - \frac{1}{2}e^{z'} \leq \phi(z')/\phi(z) \leq (1 - \frac{1}{2}e^z)^{-1}$. The block gradients are linear in the rates, so replacing every η_t by $e^{c'-c}\eta_t$ scales G by $e^{c'-c}$, which Remark 1 removes; what remains is the reweighting of tokens by $\phi(z')/\phi(z)$, which deviates from 1 by at most $\frac{1}{2}\max(e^z, e^{z'})(1 + O(e^z))$. □

C Experimental setup

Table 4: Complete setup used for every run unless a table states otherwise.

Item	Value
Model	Released Scal3R code and checkpoint, unmodified weights (identifiers in Appendix A)
Window	300 frames, taking every 2nd frame of the recording (600 recorded frames spanned); long-window runs use 1,200 frames (2,400 recorded frames)
Window offsets	start shifted by $0, 5, 10, 15, 20$ subsampled frames (0–40 recorded frames); cross-sequence cells use offsets 0 and 10
Blocks	60 frames with overlap 30 (9 blocks for 300 frames, 41 for 1,200)
Processing resolution	518 pixels (released default)
TTT layers	global-attention layers $\{14, 17, 20, 23\}$, width 1024, SwiGLU fast weights W_0, W_1, W_2
Update	summed block gradients; Newton–Schulz, 5 steps, bf16; no step size after orthogonalisation; per-column norm restoration; base rate $\eta_{\text{base}} = 0.01$
Loop closure	released settings (SALAD descriptor)
Hardware	one NVIDIA A100-80GB per run
Determinism	identical output within a container; at most 1.4×10^{-5} relative ATE difference across containers
TUM RGB-D sequences (recorded frames)	`fr3_long_office_household` (2,585), `fr2_desk` (2,965), `fr1_room` (1,362), `fr2_large_no_loop` (3,359), `fr2_pioneer_slam` (2,921)
7-Scenes	first sequence of *chess*, *fire*, *office*, *redkitchen* (1,000 frames each), converted to the TUM layout: timestamp = frame index/30 s, ground truth = camera-to-world translation and rotation
Ground-truth association	nearest timestamp within 20 ms; frames without a match are not scored
ATE	RMSE of camera positions after a least-squares Sim(3) alignment (Umeyama, 1991) to ground truth, on the frames retained after Scal3R's overlap de-duplication
Paired Δ	$100\ (\text{ATE}_{\text{variant}}/\text{ATE}_{\text{released}} - 1)$ on the same window
Training	none; all interventions are at inference time

D CORRECTION LOG

Table 5: Claims revised during the audit.

Initial claim	Measurement that changed it	Revised claim
The learned rate head collapses to a global scalar (inferred from weight RMS near initialisation scale).	Direct per-token measurement: median CV 0.45 (`fr3_long_office`), 0.51 (`fr2_desk`); p95/p5 4.08, 4.51.	Rates vary substantially; only their uniform component cancels (Section 6.4).
Newton–Schulz normalises by the spectral norm.	Reading the implementation: division by the Frobenius norm.	Frobenius normalisation (Algorithm 1); scale invariance unaffected.
Base learning rates 0.005–0.04 are bitwise invariant.	ATE differs in the fifth significant digit.	Invariant to within 0.023%, consistent with bf16 and a small token-dependent residual.
Duplicating frames causes zero drift in the untouched region.	Region-B ATE varies by up to 7% between k values, non-monotonically.	No systematic change with k.
Rank-1 collapse of the fast-weight memory (from one layer, one sequence).	Gram analysis on four layers and six sequence settings: effective rank 1.10–4.89.	Block gradients always agree in sign; concentration is scene- and layer-dependent.
Removing the TTT branch improves ATE (from one sequence).	Cross-sequence cells (Table 13, Table 14).	Small reduction in 11 of 13 paired cells ($\lvert\Delta\rvert \leq 0.93\%$), not uniform in sign.
Inference is deterministic: repeated runs are identical.	Re-running the released configuration on other GPU containers.	Deterministic within a container; up to 1.4×10^{-5} relative across containers.
Step size and online updating leave the output unchanged (first run of Section 8).	The switch was defined after the script's entry point and never executed; a per-run marker now confirms activation.	First run discarded; Table 12 reports the verified re-run.
The update overwrites the pretrained fast weights (intermediate hypothesis for why the output is insensitive).	Applying the released update rule to the pretrained weights: the Newton–Schulz output is 0.36–$0.70\times$ the pretrained column norm (Section 6.3).	Refuted: the update rotates the weights by roughly 20°–34° without replacing them; the output is insensitive to that change.
Aligned block gradients imply that any subset of blocks gives nearly the same update (reading of Proposition 1).	Read-only probe of the Newton–Schulz outputs: cosine 0.54–0.81 between first-block and full updates (gradient cosine 0.77–0.99); final fast weights differ by 5–16%.	Refuted: gradient redundancy does not imply update redundancy; the first-block result reflects the output's insensitivity to the fast weights (Fig. 2C).

E DATA PROVENANCE

Table 6: Source of every reported number. All numbers come from logged runs of the patched pipeline and are regenerated by the released analysis scripts.

Numbers	Source
Table 11; Table 10; offset-0 row of Table 1; runtimes; metric validation	per-run result records and summary logs of the offset-0 intervention grid on `fr3_long_office`
Offsets 5–20 in Table 1, Figure 1, noise floor	per-run result records of the released, branch-removed and frozen runs at each offset; mean, sd and $\lvert d\rvert/\mathrm{sd}$ computed from these values
Figure 3, Table 8, Table 7	gradient-probe Gram arrays of each TTT layer and pathway, recorded during the released baseline run of each sequence
Table 9	per-run result records of the duplication experiment ($k = 1, 2, 3, 5$)
Rate-head weight RMS; per-token rate spread and hook ATE	checkpoint inspection; read-only rate-logging runs on `fr3_long_office` and `fr2_desk` (per-call spread records and their summary)
Tables 13, 2, 14	per-run result records of the cross-sequence intervention grid; statistics computed by the released statistics script
Tables 12, 3, 17 and the appendix tables	per-run result records of the update-variant runs (with activation markers), the selection runs and the 7-Scenes runs
Block size, overlap, width, defaults, Algorithm 1	released code and configuration at the commit in Appendix A

F PER-PATHWAY GRADIENT GEOMETRY

Table 7: Per-block gradient concentration across sequences (released baseline, concatenated gradients): effective rank at layers 14/17/20/23, number of blocks n, and minimum pairwise cosine over the four layers. No negative pairwise cosine occurs in any cell. Per-pathway values for `fr3_long_office` are in Table 8.

Sequence (frames)	n	L14	L17	L20	L23	min cosine
`fr3_long_office` (300)	9	1.35	1.83	1.92	1.10	0.60
`fr2_desk` (300)	10	1.39	1.78	1.92	1.16	0.57
`fr2_large_no_loop` (300)	12	1.41	3.31	3.39	1.40	0.30
`fr2_pioneer_slam` (300)	10	1.52	2.75	3.60	2.38	0.23
`fr1_room` (300)	10	1.66	3.96	4.89	2.87	0.07
`fr3_long_office` (1200)	41	1.58	3.14	3.28	1.46	0.25

Table 8: Per-block gradient geometry at the four TTT layers (`fr3_long_office`, 9 blocks): leading-eigenvalue share / effective rank / minimum pairwise cosine. "Concatenated" uses the raw Gram of the concatenated gradients; per-pathway columns use each pathway's cosine-similarity matrix.

Layer	Concatenated	W_0	W_1	W_2
14	93.5% / 1.35 / 0.84	97.7% / 1.15 / 0.95	93.0% / 1.37 / 0.82	97.4% / 1.16 / 0.94
17	84.6% / 1.83 / 0.60	89.9% / 1.60 / 0.77	88.8% / 1.60 / 0.70	79.3% / 2.03 / 0.55
20	85.4% / 1.92 / 0.64	85.4% / 1.92 / 0.64	86.1% / 1.86 / 0.67	87.9% / 1.75 / 0.69
23	98.6% / 1.10 / 0.96	97.9% / 1.13 / 0.95	97.6% / 1.15 / 0.94	98.8% / 1.08 / 0.97

G FRAME DUPLICATION

We split the window into region A (first 150 frames) and region B (last 150) and repeat every frame of A k times. If the update stored region-specific content, over-weighting A should alter predictions on B.

Table 9: Duplicating region A k times; metrics on the untouched region B. Gram statistics at layer 23.

k	ATE on B (m)	AbsRel on B	Leading-eigenvalue share	Effective rank
1	0.01949	0.08129	98.59%	1.097
2	0.02087	0.08128	98.44%	1.105
3	0.01932	0.08141	98.36%	1.115
5	0.01749	0.08160	98.47	1.110

Region-B depth error is unchanged to three significant digits, and region-B ATE shows no systematic change with k ($k = 2$ is 7% above $k = 1$, $k = 3$ is 1% below, both within the offset floor; Table 9). This is consistent with an update dominated by a direction shared across blocks, although few values of k on one sequence cannot exclude smaller region-specific effects.

H ADDITIONAL FIGURES AND TABLES

Table 10: Using a subset of block gradients (`fr3_long_office`, 300 frames, offset 0). ATE in metres; runtime is single-A100 wall-clock seconds for the full pipeline, one run each.

Blocks contributing to G	ATE	Δ ATE	Runtime (s)
All 9 (released)	0.03416	—	402
Stride 2 (5 of 9)	0.03416	0.00%	384
Stride 3 (3 of 9)	0.03416	0.00%	354
First block only (1 of 9)	0.03418	+0.06%	354
TTT branch removed	0.03394	−0.64%	338

Table 11: Magnitude and orthogonalisation settings on `fr3_long_office` (300 frames, offset 0). ATE in metres; Δ relative to the released configuration.

Configuration	ATE	Δ
Released ($\eta_{\text{base}} = 0.01$, 5 NS steps, bf16)	0.0341607	—
$\eta_{\text{base}} = 0.005$	0.0341583	−0.007%
$\eta_{\text{base}} = 0.02$	0.0341562	−0.013%
$\eta_{\text{base}} = 0.04$	0.0341530	−0.023%
Newton–Schulz in fp32	0.0341575	−0.009%
3 Newton–Schulz steps	0.0342067	+0.13%
8 Newton–Schulz steps	0.0342883	+0.37%

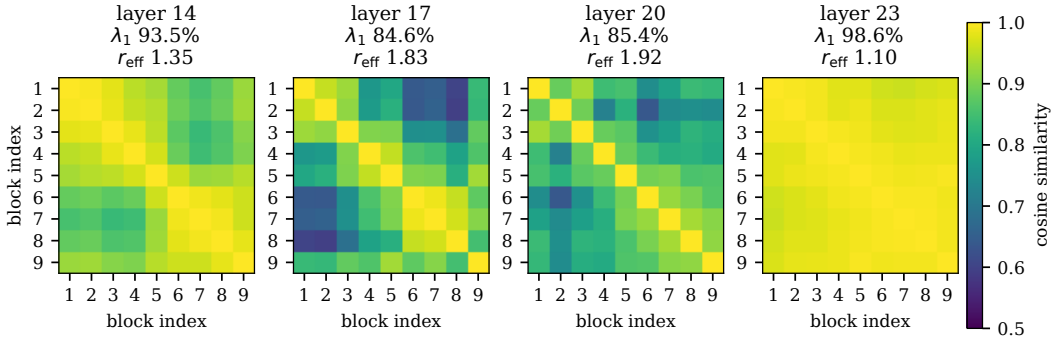


Figure 3: Pairwise cosine similarity of the nine per-block TTT gradients (concatenated over W_0, W_1, W_2) at each of the four TTT layers, `fr3_long_office`, 300 frames, offset 0. Titles: leading-eigenvalue share of the block Gram matrix and entropy effective rank $r_{\text{eff}} = \exp(-\sum_i p_i \log p_i)$.

I UPDATE VARIANTS

Table 12: Attempts to make the TTT update adapt: change of ATE (%) against the released model on the same window (offset 0). *Step size*: the orthogonalised update is scaled by s before the norm restoration. *Online*: fast weights are updated after every block, each block's gradient taken at the current weights; *causal*: block b uses the weights after blocks $\leq b$; *final*: every block uses the final weights. *Partial sum*: block b uses the update computed from the gradients of blocks $\leq b$. *Residual gain*: the branch output is multiplied by g (released: $g = 1$; $g = 0$ is branch removal).

Variant	fr3_long_office	fr2_desk	fr1_room
Step size $s = 0.25$	$+0.28$	$+0.20$	$+0.34$
Step size $s = 0.5$	$+0.12$	$+0.07$	$+0.08$
Step size $s = 2$	-0.05	$+0.07$	$+0.17$
Online, causal	$+2.35$	$+4.67$	$+7.79$
Online, causal, $s = 0.5$	$+0.54$	$+0.14$	$+1.84$
Online, final weights	-3.92	$+26.02$	$+13.18$
Partial sum, causal	$+0.02$	$+0.04$	-0.06
Residual gain $g = 0$ (branch removed)	-0.64	$+0.00$	-0.77
Residual gain $g = 0.25$	-0.50	$+0.01$	-0.58
Residual gain $g = 0.5$	-0.34	$+0.01$	-0.38
Residual gain $g = 1.5$	$+0.31$	-0.01	$+0.36$

Every run carried an activation marker written from inside the modified code path. Rescaling the orthogonalised step moves ATE by at most 0.34% and the partial sum by at most 0.06%. Block-by-block adaptation is the only variant with a large effect: the causal online update raises ATE by 2.3–7.8%, and applying the final online weights to every block lowers it by 3.9% on fr3_long_office but raises it by 26.0% and 13.2% elsewhere. Applied to the released checkpoint without retraining, such adaptation is therefore unsafe; these variants are out of distribution for a model trained with one-shot updates, and a model trained with them is untested. The residual gain, in contrast, moves ATE almost linearly (no effect on fr2_desk): the branch's contribution is set largely by the magnitude of its residual. No update variant improves ATE on all three sequences.

J CELL-LEVEL REPLICATION, POOLED STATISTICS, DECOMPOSITION OF THE SELECTION GAIN AND EQUAL-COMPUTE BASELINE

Table 13: Replication on four further TUM sequences (300 frames) at window offsets 0 and 10. Baseline ATE in metres; other columns are paired Δ ATE (%) against the baseline of the same sequence and offset. fr3_long_office rows repeat Tables 11–1 ($-$: not run).

Sequence	Off.	Baseline	Branch	Zero	Frozen	1 block	Stride 3	$\eta_{\text{base}}=.005$	$\eta_{\text{base}}=.04$
fr3_long_office	0	0.03416	-0.64	-0.64	$+0.59$	$+0.06$	0.00	-0.007	-0.023
	10	0.03659	-0.90	$-$	$+0.90$	$-$	$-$	$-$	$-$
fr2_desk	0	0.02885	$+0.00$	$+0.00$	$+0.37$	$+0.08$	$+0.01$	-0.00	$+0.02$
	10	0.03803	-0.21	-0.21	$+0.28$	$+0.07$	-0.01	$+0.00$	-0.00
fr1_room	0	0.10011	-0.77	-0.77	$+1.03$	$+0.09$	-0.06	$+0.01$	-0.01
	10	0.10602	-0.41	-0.41	$+0.94$	$+0.11$	$+0.02$	$+0.02$	-0.02
fr2_large_no_loop	0	0.09324	-0.27	-0.27	$+0.05$	$+0.07$	$+0.01$	-0.01	-0.01
	10	0.10565	-0.13	-0.13	-0.03	-0.03	-0.05	-0.00	$+0.00$
fr2_pioneer_slam	0	0.26445	$+0.16$	$+0.16$	-0.39	-0.09	-0.14	$+0.01$	-0.00
	10	0.23922	-0.47	-0.47	-0.18	$+0.14$	$+0.01$	$+0.01$	$+0.01$

Table 14: Paired effects pooled over (sequence, offset) cells from Tables 1 and 13. Δ is the paired percent change of ATE against the released model in the same cell. Sign test: exact two-sided binomial test on the number of negative cells. CI: 95% bootstrap interval of the mean Δ over cells (10,000 resamples). Cells from one sequence share a scene and are not independent; p and CI are descriptive.

Intervention	n cells	median Δ%	max$\lvert\Delta\rvert$%	#negative/n	sign-test p	95% CI of mean Δ
Remove TTT branch	13	-0.47	0.93	11/13	0.022	$[-0.67, -0.29]$
Zero TTT residual (compute-matched)	9	-0.27	0.77	7/9	0.180	$[-0.49, -0.12]$
Frozen fast weights (no test-time step)	13	$+0.58$	1.03	3/13	0.092	$[+0.21, +0.72]$
First block gradient only	9	$+0.07$	0.14	2/9	0.180	$[+0.01, +0.09]$
Every 3rd block gradient	9	-0.00	0.14	5/9	1.000	$[-0.06, +0.00]$
Base LR 0.005	9	$+0.00$	0.02	4/9	1.000	$[-0.00, +0.01]$
Base LR 0.040	9	-0.00	0.02	6/9	0.508	$[-0.01, +0.00]$

Table 15: Decomposition of select-then-average into averaging and selection: change of ATE (%) against the released model's expected single-window ATE. *Selection adds* is select-then-average minus averaging alone (five candidates).

Sequence	Averaging	Selection adds	Select+avg
`fr3_long_office`†	-6.59	-19.64	-26.22
`fr2_desk`	-9.96	0.00	-9.96
`fr1_room`	-2.26	0.00	-2.26
`fr2_large_no_loop`	-4.79	-0.52	-5.31
`fr2_pioneer_slam`	-2.70	-2.83	-5.54
TUM mean excluding †	-4.93	-0.84	-5.77
7-Scenes *chess*	-1.66	-1.79	-3.45
7-Scenes *fire*	-5.69	-3.12	-8.81
7-Scenes *office*	-0.72	$+3.38$	$+2.66$
7-Scenes *redkitchen*	-3.87	-16.81	-20.67
7-Scenes mean	-2.99	-4.58	-7.57

Table 16: Equal-compute comparison on TUM. All methods are scored on the same 285 frames against the released model's mean offset-0/offset-10 ATE on those frames (Δ ATE, %). *Avg2*: released model averaged over offsets 0 and 10 (2 runs). *Avg4*: released model averaged over offsets 0, 5, 10, 15 (4 runs, equal compute). *Select+avg*: two-candidate rule (released model and adaptive Huber at offsets 0 and 10; 4 runs). †candidates identified on this sequence. The equal-compute test on 7-Scenes would need further released-model offsets and was not run.

Sequence	Avg2	Avg4 (equal compute)	Select+avg (4 runs)
`fr3_long_office`†	-6.64	-6.36	-26.16
`fr2_desk`	-9.97	-7.04	-9.97
`fr1_room`	-2.23	$+2.18$	-2.23
`fr2_large_no_loop`	-5.13	-4.53	-5.74
`fr2_pioneer_slam`	-2.79	-10.73	-5.57
Mean	-5.35	-5.29	-9.93
Mean excluding †	-5.03	-5.03	-5.88

K SECOND DATASET: 7-SCENES

We repeat the TTT interventions on the first sequence of four 7-Scenes (Shotton et al., 2013) scenes, converted to the TUM layout (timestamp = frame index / 30, camera-to-world poses as ground truth), with the same 300-frame, stride-2 window and no change to any setting. No configuration was tuned on 7-Scenes.

Table 17: TTT interventions on 7-Scenes (offset 0). Baseline ATE in metres; other columns are Δ ATE (%) against the released model on the same window.

Scene	Baseline	Branch	Frozen	1 block	Stride 3	η_{base}=.005	η_{base}=.04
chess	0.05067	−0.23	−0.36	+0.09	+0.03	−0.00	+0.01
fire	0.03526	−0.26	+0.25	+0.03	+0.02	−0.00	+0.01
office	0.10251	−0.29	−0.11	+0.11	+0.00	−0.00	+0.00
redkitchen	0.05404	−0.04	−0.23	+0.01	+0.02	+0.00	+0.00

Every intervention stays within $\pm 0.36\%$ on all four scenes. Removing the branch lowers ATE slightly on every scene (-0.04 to -0.29%); freezing the fast weights lowers it on three scenes and raises it on *fire*; block subsampling and the base rate stay within 0.11%. The released TTT pathway is thus also small on a second dataset, but the sign of the frozen-weight effect differs from `fr3_long_office`: here freezing lowered ATE in three of four scenes.

K.1 Which layers carry the effect

Table 7 shows that block gradients are most concentrated at layers 14 and 23 and most spread at 17 and 20. To locate the branch's contribution, we disable test-time training at one layer while keeping the other three (*skip*), and conversely keep it at one layer only (*only*), on the two most concentrated sequences.

Table 18: Per-layer TTT ablation (offset 0, 300 frames). Paired Δ ATE (%) against the released model of the same sequence. *Skip* ℓ: TTT disabled at layer ℓ only; *only* ℓ: TTT active at layer ℓ only. For reference, removing the branch at all four layers gives -0.64 (`fr3_long_office`) and $+0.00$ (`fr2_desk`).

	Skip layer				Only layer			
Sequence	14	17	20	23	14	17	20	23
`fr3_long_office`	−0.05	−0.26	−0.36	+0.00	−0.59	−0.40	−0.31	−0.65
`fr2_desk`	+0.04	−0.03	+0.04	−0.00	−0.01	+0.06	−0.04	+0.00

If the per-layer effects were additive, the four *skip* entries of a sequence would sum approximately to the full-removal effect. On `fr3_long_office` the four *skip* entries sum to -0.67%, close to the full-removal effect (-0.64%, Table 1), so the per-layer contributions are approximately additive. Layer 23 contributes nothing measurable (skipping it changes ATE by 0.00%, and keeping only it reproduces full removal); the small total effect is carried by layers 14–20. On `fr2_desk`, where full removal is itself neutral, every per-layer configuration stays within $\pm 0.06\%$. Each cell is a single window offset and is read against the $|\Delta|$ scale of Table 1, not declared separable on its own.

K.2 A longer window

On 1,200 frames of `fr3_long_office` (41 blocks) the released model's ATE is 0.26061 m, $7.6\times$ its 300-frame value, and per-block gradients are less concentrated at layers 17–23 than on 300 frames (Table 7). This tests whether the short-window findings persist when the fast weights must summarise four times as many frames.

Table 19: TTT interventions on 1,200 frames of `fr3_long_office` (offset 0). Δ ATE (%) against the released model on the same window; runtime in seconds.

Configuration	Δ ATE (%)	Runtime (s)
Released (ATE 0.26061 m)	—	1600
TTT branch removed	-0.96	1487
Frozen fast weights	$+0.65$	1458
First block gradient only (1 of 41)	$+0.38$	1472
Every 3rd block gradient	-0.00	1617
$\eta_{\text{base}} = 0.005$	$+0.00$	1688
$\eta_{\text{base}} = 0.04$	-0.01	1595

Remark 1 predicts that the base-rate rows stay near zero regardless of length. Because gradients are more spread on the long window, the pre-specified mechanistic prediction of Section 7 also predicts a larger *first-block-only* deviation here than on 300 frames ($+0.06\%$). The base-rate rows stay within 0.01%, as predicted. The first-block-only deviation is $+0.38\%$ with one of 41 block gradients, larger than the $+0.06\%$ observed with one of nine at 300 frames, which is the predicted direction. Removing the branch (-0.96%) and freezing the fast weights ($+0.65\%$) reproduce the 300-frame pattern. Over this window the released model's ATE grows $7.6\times$ relative to the 300-frame window, while no TTT intervention moves it by more than 1%: on this sequence the TTT pathway does not counteract long-horizon drift. Wall-clock times at this length vary between 1458 and 1688 s across containers, so we draw no runtime conclusion from them. This is a single window with no offset replicate and is reported descriptively.