SURGICALMAMBA: DUAL-PATH SSD WITH STATE REGRAMMING FOR ONLINE SURGICAL PHASE RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Online surgical phase recognition labels each frame of an operation from past frames alone. Because procedures run for hours, the cost of each prediction cannot grow with elapsed time. Structured state-space duality (SSD) meets that constraint, but is not well matched to surgical video in two ways. The same views recur through an operation, so a scene is written over its earlier occurrence and afterwards only age separates the two. Phase durations differ by an order of magnitude, so a decay fast enough to clear a phase that passes in seconds erodes one that runs for minutes. The rate does vary from frame to frame, but on contexts this long it is not reliably learned into a forgetting schedule. *State regramming* applies an input-conditioned orthogonal rotation to the state handed across chunk boundaries, so where a frame is written also depends on what has passed since, and two occurrences of the same view are held apart when different phases intervene. *Intensity-modulated stepping* supervises a time-warp of the discretization step toward the annotated phase boundaries, clearing the state where a phase ends so that the decay within phases can lean toward retention. Both preserve SSD's N-semiseparable structure and $O(d)$ per-frame cost. On seven public benchmarks *SurgicalMamba* matches or exceeds the state of the art in online accuracy and phase-level Jaccard ($94.6\%/82.7\%$ on Cholec80, $89.5\%/68.9\%$ on AutoLaparo) at 312.88 fps on a single GPU. Code :`https://anonymous.4open.science/r/Surgical-Mamba-8930`

1 INTRODUCTION

Online surgical phase recognition (SPR) identifies the current phase of an ongoing procedure from streaming endoscopic video, and is the temporal foundation for context-aware operating-room systems Maier-Hein et al. (2017); Garrow et al. (2021); Hashimoto et al. (2018); Mascagni et al. (2022). Unlike offline segmentation the setting is causal: the model commits to a prediction at frame t from frames $1{:}t$ alone, at bounded per-frame latency Twinanda et al. (2016a); Jin et al. (2017). A procedure runs 30–90 minutes Twinanda et al. (2016a); Wagner et al. (2023), so a recognizer must reason over an hour-long horizon while holding per-frame cost fixed. Transformer- and memory-based recognizers meet the first demand at the expense of the second, since attention scope or an external feature cache grows with elapsed length.

We therefore build on Mamba2's structured state-space duality (SSD) Gu & Dao (2024); Dao & Gu (2024), whose chunked training and recurrent inference compute the same function, so one set of weights trains on tensor cores and streams at $O(d)$ per frame regardless of elapsed length. In SSD, computing a chunk as one matrix multiply requires a per-head scalar transition, which rescales the state but never moves content between directions. Where a frame is written is therefore fixed by $\bar{B}$, and how fast the state empties by the step Δ, both computed from that frame alone. Surgical video makes each of these a problem.

The same instruments and views return throughout an operation, and frames from phases minutes apart can be near indistinguishable (Fig. 1B). Such frames select nearly the same $\bar{B}$, so the later occurrence is written on top of the earlier one, and the transition that follows only rescales what is there. The two are then separated by their difference in age alone, which vanishes once both are old.

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

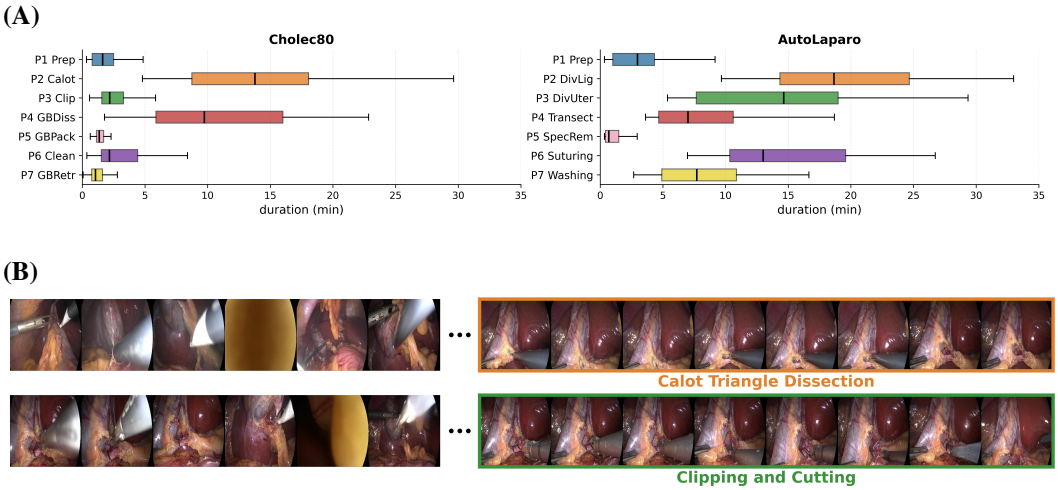


Figure 1. Two properties of surgical video that the SSD recurrence does not accommodate. *(A)* Distribution of phase duration over videos. Boxes span the interquartile range and whiskers the full range. Within a single procedure the longest phase runs roughly ten times the shortest (15.6 against 1.4 minutes on Cholec80, 20.9 against 2.0 on AutoLaparo), so a decay rate fast enough to clear a short phase erodes a long one from within, while one slow enough to hold a long phase does not clear a short one. λ raises the decay at the boundaries, which eases this trade-off (§3.3). *(B)* Frames from two phases of one Cholec80 procedure, minutes apart. The same instruments, organs and viewpoint recur throughout, so a write direction chosen from the current frame alone places a recurring scene on top of its earlier occurrence, after which only age separates the two. Z makes that placement depend on what has intervened (§3.5).

State regramming (Z) rotates the carried state at each chunk boundary, by an amount that chunk's content decides. Where a frame sits then depends on what has passed since it was written. The same clipping scene lands in a different part of the state when a dissection has intervened, and a read-out can pick out one occurrence without the other. The rotation leaves SSD's N-semiseparable structure and $O(d)$ per-frame cost intact (§3.5).

Phase lengths span an order of magnitude (Fig. 1A). Dissection runs for minutes where clipping passes in seconds, so a single decay rate fast enough to clear a short phase erodes a long one from within. Δ is recomputed at every frame, so the decay rate can vary through the procedure. Prior work reports that this variation is not reliably learned into a forgetting schedule on long contexts Chen et al. (2025); Wang et al. (2025). An hour-long procedure is such a context. *Intensity-modulated stepping* (λ) increases the decay at the phase boundaries. Those boundaries are annotated, so we supervise λ toward them and let it empty the state quickly where a phase ends and slowly in between. The decay within phases can then lean toward retention, which eases the trade-off between clearing short phases and holding long ones (Appendix B).

Both mechanisms sit in a dual-path SSD block: a slow path carries state across clips under λ-warped stepping, while a fast path resets at clip boundaries.

Our contributions are (i) **state regramming**, a chunk-granular, exactly norm-preserving, input-conditioned rotation of the SSD carried state, so that what a frame reads from a past one depends on what lay between them rather than on the two frames and their separation alone, while provably preserving N-semiseparability and $O(d)$ per-frame inference; (ii) **intensity-modulated stepping**, a supervised continuous-time time-warp of the discretization step that separates rate control from Mamba2's unsupervised content selectivity; (iii) a **streaming-compatible design** in which both mechanisms and the dual-path carry preserve SSD's chunk structure end to end, holding per-frame inference at $O(d)$; and (iv) **state-of-the-art online SPR** on seven benchmarks under strict causal evaluation at 312.88 fps.

2 Related Work

2.1 Surgical phase recognition

Early recognizers paired CNN backbones with recurrent or convolutional temporal models Jin et al. (2017; 2020); Czempiel et al. (2020), and transformer-, memory-, and retrieval-based models then extended temporal scope Jin et al. (2021); Gao et al. (2021); Liu et al. (2025; 2023); Yang et al. (2024); Yue et al. (2023) at a streaming cost: per-frame attention grows with elapsed length or is capped by a sliding window Wang et al. (2021); Xu et al. (2021). Three recent recognizers frame the setting we target. Rivoir et al. (2024) show that BatchNorm is the obstacle to end-to-end training in this setting, and that with a backbone without it, such as ConvNeXt Liu et al. (2022), single-stage training is competitive. DACAT Yang et al. (2025a) retrieves the most correlated past clip from an unbounded cache, growing as $O(Td)$ and assuming the most informative past is the most visually similar. MTTR-Net Huang et al. (2025) closes the train/test feature gap with multi-teacher cross-mimicking, at the cost of $K+1$ encoders in a two-stage pipeline. SSMs have entered the field in several forms, as temporal modules Cao et al. (2024); Zhang et al. (2026), within hybrid temporal stacks Fan et al. (2026), and as an endoscopic foundation model Tian et al. (2025), though the bidirectional temporal scan of Cao et al. (2024) is non-causal and therefore $O(Td)$ online. In all of them the recurrence advances at a uniform rate, and where a frame is written into the state is fixed by that frame alone. Neither of the two handles we contribute, λ (§3.3) and Z (§3.5), is available in any of them.

2.2 Rotations in recurrent and state-space models

Orthogonal and unitary RNNs constrain the recurrence *weight* matrix against exploding and vanishing gradients Arjovsky et al. (2016); Mhammedi et al. (2017); Jing et al. (2017); Helfrich et al. (2018); Lezcano-Casado & Martınez-Rubio (2019), with a single learned matrix applied at every step. The Cayley parameterization we use comes from that line, though the rotated object and its purpose differ. In state-space models, rotations arise from complex-diagonal state transitions, which give each channel pair a fixed rotation in its own plane Gu et al. (2022); Orvieto et al. (2023). Mamba-3 Lahoti et al. (2026) makes these transitions data-dependent within SSD, rotating the state at every step on fixed coordinate pairs by angles computed from the input, and applies the rotation as a rotary embedding on the B, C projections. State regramming instead rotates the carried state at chunk boundaries, densely across all N dimensions, in planes chosen by that chunk's content. Appendix C.6 relates the two.

2.3 Forgetting in recurrent sequence models

Selective SSMs already decide how much to forget at every frame. In Mamba the step Δ is computed from the current token and fixes how much of the state survives it Gu & Dao (2024); Dao & Gu (2024). Recent work has found two things wrong with how that decision is made. Gated DeltaNet Yang et al. (2025b) takes it to be too coarse, since a single Δ decays every key–value association by the same factor, and adds a delta-rule term so that a write can target one association without disturbing the rest. Chen et al. (2025) find instead that the decision is barely learned at all, Mamba-2 failing to forget earlier content because training on contexts short relative to the state size lets it do well without ever needing to. Wang et al. (2025) attribute the same difficulty to exponential decay itself. Both answer with the architecture, by enlarging what the recurrence can express or by changing how it is trained.

In all of these the model must infer for itself when the past stops being useful. Where a task marks that point, the inference is unnecessary. The supervision target is the transition itself, read off the frame-level phase labels, and λ is supervised toward it (§3.6), so what the state releases and when follows the structure of the procedure rather than the frame-level loss. The price is that the task must have such structure.

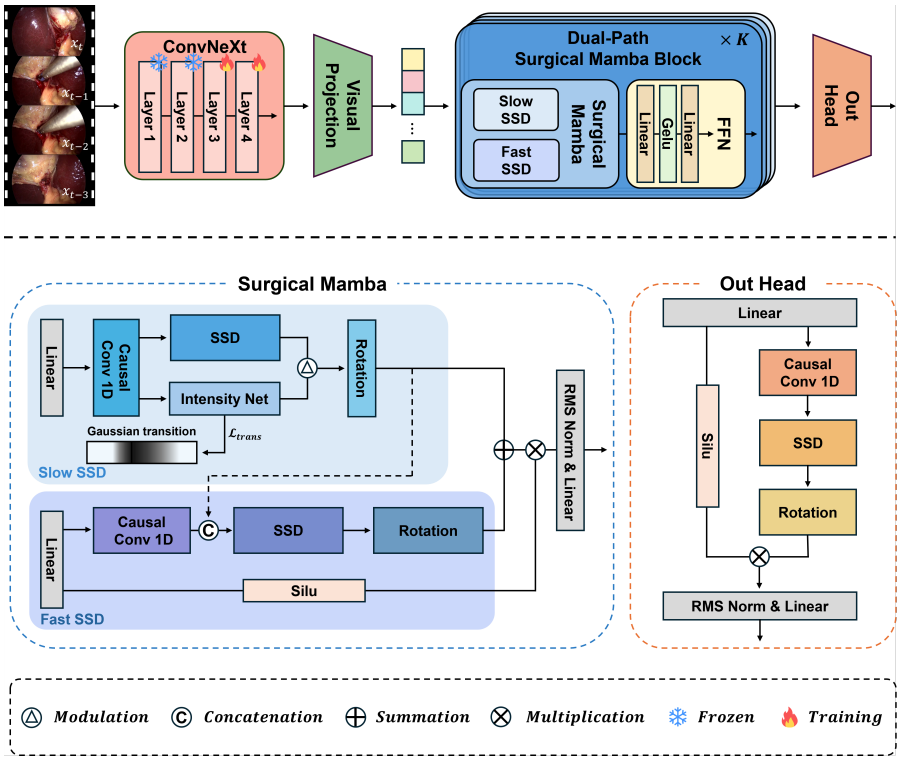


Figure 2. Overview of SurgicalMamba. Frames are encoded by a partially frozen ConvNeXt and processed by K stacked Dual-Path blocks, each running a Slow and a Fast SSD path plus a GELU FFN, before an Out Head produces per-frame phase predictions. *Bottom left:* within a block, both paths apply a chunk-boundary rotation (§3.5). The Slow path additionally carries state across clips, and an Intensity Net reads each frame to set λ, which lengthens the discretization step where a phase is ending so the state clears there rather than throughout (§3.3). The Slow path's output conditions the Fast path's selectivity. *Bottom right:* the Out Head mirrors the Slow path without cross-clip carry.

3 METHOD

3.1 PRELIMINARIES AND NOTATION

Mamba2 Gu & Dao (2024); Dao & Gu (2024) discretizes a continuous-time selective SSM with input-dependent step $\Delta_n = \Delta(x_n)$, giving the per-head recurrence

$$h_n = \bar{A}_n\, h_{n-1} + \bar{B}_n\, x_n, \qquad y_n = C_n\, h_n, \qquad \bar{A}_n = \exp(\Delta_n A),\ \ \bar{B}_n \approx \Delta_n B_n, \tag{1}$$

with $A < 0$ a *per-head scalar*, so $\bar{A}_n$ acts on the state $h \in \mathbb{R}^{H\times P\times N}$ by element-wise decay. The projections B_n and C_n are likewise computed from the frame, and fix the two things the recurrence does with a direction of the state: B_n says where frame n is written, and C_n which directions the output at n reads. The SSD form computes (1) chunk-wise: a clip of L frames is split into $n_c = L/C_{\text{chunk}}$ chunks, each computed as one fused matrix multiply, with exactly one state tensor passed across each chunk boundary. Dually, the same map is an N-semiseparable matrix Dao & Gu (2024), the property that yields $O(d)$ per-frame inference and that our structural result (Proposition 2) preserves. The zero-order-hold derivation and the notation used throughout are in Appendix A.

3.2 OVERVIEW

Online phase recognition produces, at each time t, a distribution $\hat{p}_t \in \Delta^{C-1}$ over C phases from frames $I_{1:t}$ only. For training, videos are split into non-overlapping clips of length L, with slow-path state carried clip-to-clip under truncated back-propagation through time (TBPTT). We write $\Delta_n = \Delta(x_n)$ for the input-dependent step and subscript quantities *slow*/*fast* by path.

SurgicalMamba keeps the standard encoder–temporal-model–classifier recipe and redesigns the temporal model (Fig. 2). Two demands pull the recurrence in opposite directions. Placing the current frame within the workflow needs context from minutes earlier, hence a state that survives clip boundaries. Reacting to a brief event needs a state not already committed to that history. The block therefore runs two SSD paths on the same input with independent projections and scans, one carrying its state across clips (§3.3) and one resetting at every clip boundary (§3.4), with the first conditioning the second. Within each path, the discretization step of the slow one is warped by a learned intensity (§3.3) and the carried state is rotated at every chunk boundary (§3.5). The surrounding pipeline and all hyperparameters are standard and given in §4.1.

3.3 Slow path and intensity-modulated stepping

Carrying state across clips makes the forward pass behave as if clips were concatenated while gradients stay bounded by the TBPTT window (Appendix D). The second departure from a standard Mamba2 block is that this path's discretization step is warped by a learned intensity.

We treat λ as a change of time variable, not an ad-hoc multiplier on Δ. Positing an intrinsic event time that runs faster where the workflow is changing and at wall-clock rate where it is not,

$$\tau(t) := \int_0^t \big(1 + \lambda(s)\big)\,\mathrm{d}s, \qquad \alpha(t) := \frac{\mathrm{d}\tau}{\mathrm{d}t} = 1 + \lambda(t) \in [1, 2], \tag{2}$$

and assuming the dynamics are stationary in τ, the chain rule re-expresses them in wall-clock time as $\mathrm{d}h/\mathrm{d}t = \alpha(t)[A\,h(t) + B(t)x(t)]$. Since λ is constant on each frame interval ($\alpha_n := 1 + \lambda_n$), exact ZOH integration gives

$$\bar{A}_n = \exp(\alpha_n \Delta A), \qquad \bar{B}_n = (\alpha_n \Delta A)^{-1}\big(\exp(\alpha_n \Delta A) - I\big)\,\alpha_n \Delta B_n \;\approx\; \alpha_n \Delta B_n, \tag{3}$$

which is *identical* to the standard Mamba2 discretization with Δ replaced by $\alpha_n \Delta$: the SSD kernel, the structured matrices, and the state shape are unchanged, and only the scalar step is reweighted. We instantiate

$$\lambda(t) = \sigma\big(\mathrm{MLP}_\lambda(\mathrm{sg}[\,x^{\mathrm{conv}}_{\mathrm{slow}}(t)\,])\big) \in [0, 1] \tag{4}$$

with a small MLP (the *Intensity Net* of Fig. 2). Its only input is the depthwise-convolved frame feature, so λ has no access to elapsed time, to the phase index, or to previous predictions: it must read a transition from local visual evidence rather than from where in a procedure the model happens to be. The input is detached ($\mathrm{sg}[\cdot]$), so $\mathcal{L}_{\mathrm{trans}}$ trains the Intensity Net alone and λ reaches the rest of the model only through $\alpha_n \Delta$. The ablation of §4.3 therefore removes the time-warp itself, and not a representation that boundary supervision has shaped.

A single decay rate must both retain information within a phase and forget it at a boundary. λ takes over part of the second role. It raises the decay only near a boundary, so the base rate no longer carries the forgetting alone and can lean toward retention (Appendix B). A single frame at most doubles its discretization step, so forgetting cannot be done in one frame and must be spread over a short window.

λ is supervised toward a window centred on each phase boundary (Fig. 2, *Gaussian transition target*), wider after the boundary than before. Forgetting costs less on a segment that has just begun than on one that has accumulated its full context, and λ is predicted causally, so the evidence for a transition arrives only after it. Appendix B derives both. The warp is confined to the slow path, whose cross-clip carry leaves a transient over-acceleration recoverable.

3.4 Fast path and fusion

The fast path is a clip-local observer: its SSM state and conv buffer reset at every clip boundary, so it reacts to within-clip events without diluting its short-term selectivity or duplicating the slow path's long-memory role. Within a clip it still propagates and rotates state across chunk boundaries (§3.5). Only clip-level carry is severed. Its selectivity is conditioned on the slow path: the fast (Δ, B, C) are computed from the local frame together with the slow path's long-context summary, so whether a grasper signals dissection or retraction can depend on instruments seen minutes earlier. Conditioning is one-way, letting the slow summary shape the fast path's selectivity without writing transient events back into the carried state. The fast scan is otherwise the standard Mamba2 form, with no λ modulation. The two outputs are summed channel-wise and gated, which keeps the SSM inner dimension at the canonical SSD value (Appendix D).

3.5 State regramming via Cayley rotation

State regramming rotates each head's carried state at chunk boundaries by an input-conditioned orthogonal $Z \in \mathbb{R}^{N\times N}$, so the direction a frame occupies depends on the chunks that have gone by and not on that frame alone. Appendix C works this out in the SSD matrix form.

At the end of chunk c the per-head SSD output $y^{(c)} \in \mathbb{R}^{B\times H\times P\times C_{\text{chunk}}}$, which includes the contribution of the state carried into the chunk, is mean-pooled over the chunk and LayerNorm-ed per head into a descriptor

$$\phi^{(c)} = \mathrm{LN}_P\Big(\tfrac{1}{C_{\text{chunk}}}\textstyle\sum_{t=1}^{C_{\text{chunk}}} y^{(c)}[:,:,:,t]\Big) \in \mathbb{R}^{B\times H\times P}, \tag{5}$$

which two per-head MLPs map to r rotation planes and their angles,

$$[U\|V] = \mathrm{MLP}_{UV}(\phi^{(c)}), \qquad \theta = \mathrm{softplus}\big(\mathrm{MLP}_\theta(\phi^{(c)})\big) \in \mathbb{R}^{B\times H\times r}_{\geq 0}, \tag{6}$$

with $U, V \in \mathbb{R}^{B\times H\times N\times r}$ column-wise L_2-normalized so that θ alone carries magnitude, letting the chunk's content choose *where* to rotate and *by how much* independently. The low-rank product is antisymmetrized,

$$\widetilde{S} = U\,\mathrm{diag}(\theta)\,V^\top, \qquad S^{(c)} = \widetilde{S} - \widetilde{S}^\top, \tag{7}$$

and the Cayley transform of the resulting skew-symmetric $S^{(c)}$ gives the rotation in closed form,

$$Z^{(c)} = \big(I - \tfrac{1}{2}S^{(c)}\big)^{-1}\big(I + \tfrac{1}{2}S^{(c)}\big) \in SO(N), \qquad h^{(c)} \leftarrow h^{(c)}\,Z^{(c)}. \tag{8}$$

We use the Cayley form rather than a matrix exponential because it is differentiable everywhere, $I - \frac{1}{2}S$ being invertible for any real skew-symmetric S, and because it is orthogonal by construction rather than approximately so, which matters when the slow path composes these rotations over tens of chunk boundaries. Initializing $\theta \approx 0$ gives $Z^{(c)} \approx I$, so training starts from vanilla SSD.

Rotating the state alone is what has an effect. The per-head scalar A commutes with any orthogonal Q, so the scan is rotation-equivariant: turning h, B and C together leaves the output unchanged. Turning only h therefore leaves the next chunk's fresh $B^{(c+1)}, C^{(c+1)}$ reading the carried state in a re-oriented basis, and each chunk summarizes itself and chooses the basis in which that summary is passed on. Nothing is lost in the process: $\|h^{(c)}Z^{(c)}\|_F = \|h^{(c)}\|_F$, so the content of the carry is redirected but not attenuated. By the same commutation the accumulated rotation can be moved onto B and C, so the boundary rotation preserves the N-semiseparable structure of SSD and its $O(d)$ per-frame inference cost, which Appendix C states as Proposition 2. Per-head batching and the cost of the rotation, which is amortized over the frames of a chunk, are in Appendix D.

Conditioning Z on content is a requirement of the mechanism rather than one of its settings. Were Z the same at every boundary, the product accumulated between chunks c' and c would reduce to $Z^{\,c-c'}$, which factors as $Z^{-c'}\cdot Z^{c}$. The same equivariance then absorbs the two factors into the chunk-c' writes and the chunk-c reads. What is left is vanilla SSD with a chunk-level relative position encoding on B and C, whose absorbed factor depends only on the chunk distance between the two frames. Only a Z that varies with content makes the accumulated product depend on what lay between two frames rather than on how far apart they are.

3.6 Output head and training objectives

A lightweight SSD-augmented head maps the last block's output to per-frame logits (Fig. 2, bottom right), mirroring the slow path without cross-clip carry: state resets each clip, rotations are applied within it, and the gated result is projected to C phase classes. The clip loss is $\mathcal{L} = \mathcal{L}_{\text{CE}} + w_{\text{sm}}\mathcal{L}_{\text{smooth}} + w_{\text{trans}}\mathcal{L}_{\text{trans}}$, over valid (non-padded) positions.

$\mathcal{L}_{\text{CE}}$ is cross-entropy with label smoothing. $\mathcal{L}_{\text{smooth}}$ penalizes frame-to-frame change in the prediction, but weighted by the confidence at both endpoints ($c_{b,t} = 1 - \mathcal{H}(\hat{p}_{b,t})/\log C$), so that it acts within a phase, where the model is confident, and relaxes at transitions, where predictions *should* change abruptly. $\mathcal{L}_{\text{trans}}$ is a per-layer BCE between each layer's intensity $\lambda^{(\ell)}$ and the transition target of §3.3, an asymmetric Gaussian peaking at 1 on each boundary with standard deviations $\sigma_L < \sigma_R$ before and after (values in Appendix D). Targets of this kind are also used for boundary supervision by LoViT Liu et al. (2025).

4 EXPERIMENTS

4.1 SETUP

We evaluate on seven public surgical-video datasets. The main text reports **Cholec80** Twinanda et al. (2016a) and **M2CAI16** Twinanda et al. (2016b), both laparoscopic cholecystectomy, and **AutoLaparo** Wang et al. (2022), laparoscopic hysterectomy. The remaining four span cataract, multi-procedure, and prostatectomy surgery. All are sub-sampled to 1 fps Twinanda et al. (2016a); Jin et al. (2017) and evaluated causally: at each frame the model predicts from $I_{1:t}$ only, with per-frame work independent of look-back length. We report video-wise accuracy and phase-wise precision, recall and Jaccard Jin et al. (2017); Funke et al. (2023); Huang et al. (2025), emphasizing Jaccard as the metric most sensitive to hard-to-recognize phases, and for Cholec80 report both with and without the 10-second relaxed boundary Twinanda et al. (2016b). Dataset splits and metric definitions are in Appendix F. The backbone, hyperparameters, and their sensitivity are in Appendices D and E.

4.2 COMPARISON WITH STATE-OF-THE-ART METHODS

We compare against results reported in the original papers, all under the protocol of §4.1 (Table 1). On Cholec80 SurgicalMamba is state-of-the-art on every metric under both protocols, with the gain concentrated on Jaccard, which improves by $+1.1$ pp relaxed and $+1.3$ pp strict over the strongest prior method. The same pattern holds on the other two main-text benchmarks, with $+2.6$ pp Jaccard over DACAT on M2CAI16 and the best accuracy and Jaccard on AutoLaparo. The four remaining benchmarks are reported in Appendix F. Our per-video accuracy deviations are among the lowest reported. To assess the temporal model on its own terms, we also train it with a ResNet-50 He et al. (2016) (GroupNorm Wu & He (2018)) as the visual backbone. Cholec80 and M2CAI16 are essentially unchanged. AutoLaparo, with only ten training videos, is not, and there the result appears to depend on how well the pre-trained backbone already generalizes.

4.3 ABLATION STUDIES

On Cholec80 we remove one component at a time from the full model, and separately build up from a vanilla Mamba2 baseline (Table 2). The two controls bracket the joint effect: a vanilla single-path Mamba2 reaches 77.7% strict Jaccard against the full model's 82.7%, and state regramming alone recovers $+1.5$ pp of that gap.

Each component contributes under both protocols, and the three main ones do so on the same order. Removing Z costs 2.0 pp strict Jaccard, the intensity 1.6, and the fast path 1.5, with the smoothness loss adding 0.9. Every variant is evaluated on the same test set, so the comparisons are paired, and we test them across the seven phases. The full model improves strict Jaccard over removing Z by $+2.01$ pp (95% CI $[+0.10, +3.91]$, paired t-test, $p = 0.042$) and over removing λ by $+1.62$ pp ($[+0.29, +2.95]$, $p = 0.025$). Seven phases give the test little power, and both differences are significant at the 5% level nonetheless. λ also acts on consistency. Removing it inflates the per-video accuracy standard deviation from 3.71 to 6.05, where removing Z or the fast path leaves it near 4.2.

4.4 STREAMING EFFICIENCY

The model holds 198.56 M parameters and spends 4.55 GFLOPs per frame, in under 1 GB of peak memory. Measuring the forward pass at batch 1 on one RTX A6000, averaged over 500 frames, it reaches 88.42 fps, and 312.88 fps with CUDA graph capture. Both are above the 25–60 fps at which endoscopic video is captured. What matters for streaming is how the rate behaves in T. Per-frame work is $O(d)$ and the carried state has fixed size, so these are steady-state rates that hold however long an operation runs. The same property is what makes graph capture applicable at all, since the computation issued per frame does not change from one frame to the next and can be captured once and replayed.

Table 1. Comparison with state-of-the-art methods on the three main-text benchmarks, each under its standard protocol. Best per block in bold, second best underlined. "–" marks values not reported by the source. The two SurgicalMamba rows are the same model differing only in the visual backbone.

Method	Backbone	Accuracy	Precision	Recall	Jaccard
Relaxed (10-second boundary)					
PhaseNet Twinanda et al. (2016a)	AlexNet	78.8 ±4.7	71.3 ±15.6	76.6 ±16.6	–
EndoNet Twinanda et al. (2016a)	AlexNet	81.7 ±4.2	73.7 ±16.1	79.6 ±7.9	–
SV-RCNet Jin et al. (2017)	ResNet-50 (BN)	85.3 ±7.3	80.7 ±7.0	83.5 ±7.5	–
MTRCNet-CL Jin et al. (2020)	ResNet-50 (BN)	89.2 ±7.6	86.9 ±4.3	88.0 ±6.9	–
TeCNO Czempiel et al. (2020)	ResNet-50 (BN)	88.6 ±7.8	81.6 ±7.0	85.2 ±6.7	75.1 ±6.9
Opera Czempiel et al. (2021)	ResNet-50 (BN)	91.2 ±6.4	82.2 ±7.0	86.9 ±8.6	–
TMRNet Jin et al. (2021)	ResNet-50 (BN)	90.1 ±7.6	90.3 ±3.3	89.5 ±5.0	79.1 ±5.7
Trans-SVNet Gao et al. (2021)	ResNet-50 (BN)	90.3 ±7.1	90.7 ±5.0	88.8 ±7.4	79.3 ±6.6
Not E2E Yi et al. (2022)	ResNet-50 (BN)	91.5 ±7.1	–	86.8 ±8.5	77.2 ±11.2
UATD Ding et al. (2023)	Inception	91.9 ±5.6	89.5 ±4.4	90.5 ±5.9	79.9 ±8.5
CMTNet Yue et al. (2023)	ResNet-50 (BN)	92.9 ±5.9	90.1 ±7.1	92.0 ±4.4	81.5 ±10.4
LoViT Liu et al. (2025)	ViT	92.4 ±6.3	89.9 ±6.1	90.6 ±4.4	81.2 ±9.1
SKiT Liu et al. (2023)	ViT	93.4 ±5.2	90.9	91.8	82.6
SR-Mamba Cao et al. (2024)	ResNet-34 (BN)	92.6 ±8.6	90.3 ±5.2	90.6 ±7.2	81.5 ±8.6
Surgformer Yang et al. (2024)	ViT	93.4 ±6.4	91.9 ±4.7	92.1 ±5.8	84.1 ±8.0
DACAT Yang et al. (2025a)	ConvNeXt-T	95.5 ±4.3	93.6 ±4.1	93.4 ±5.3	87.4 ±8.1
MTTR-Net Huang et al. (2025)	ResNet-50 (BN)	95.0 ±4.9	92.7 ±5.1	91.2 ±10.3	84.7 ±11.0
SPRMamba Zhang et al. (2026)	ResNet-50 (BN)	93.6 ±5.3	90.6 ±5.1	92.4 ±5.1	83.4 ±6.4
SurgicalMamba (Ours)	ConvNeXt-T	**96.1** ±3.6	**94.9** ±4.2	**94.4** ±6.2	**88.5** ±8.1
	ResNet-50 (GN)	95.5 ±3.7	94.1 ±4.5	93.0 ±11.1	87.1 ±11.5
Strict (unrelaxed)					
Trans-SVNet Gao et al. (2021)	ResNet-50 (BN)	89.1 ±7.0	84.7	83.6	72.5
LoViT Liu et al. (2025)	ViT	91.5 ±6.1	83.1	86.5	74.2
SKiT Liu et al. (2023)	ViT	92.5 ±5.1	84.6	88.5	76.7
Surgformer Yang et al. (2024)	ViT	92.4 ±6.4	87.9 ±6.9	89.3 ±7.8	79.9 ±10.2
DACAT Yang et al. (2025a)	ConvNeXt-T	93.5 ±5.3	87.5 ±6.1	88.5 ±8.3	79.1 ±11.7
MTTR-Net Huang et al. (2025)	ResNet-50 (BN)	93.9 ±5.0	88.8 ±6.9	88.2 ±10.8	80.5 ±11.8
SPRMamba Zhang et al. (2026)	ResNet-50 (BN)	93.1 ±4.6	89.3 ±6.7	90.1 ±5.6	81.4 ±6.9
SurgicalMamba (Ours)	ConvNeXt-T	**94.6** ±3.7	**89.6** ±8.7	**90.5** ±8.1	**82.7** ±11.5
	ResNet-50 (GN)	94.2 ±3.9	88.3 ±9.6	89.3 ±12.7	81.8 ±13.7
M2CAI16 (relaxed, 10-second boundary)					
TMRNet Jin et al. (2021)	ResNet-50 (BN)	87.0 ±8.6	87.8 ±6.9	88.4 ±5.3	75.1 ±6.9
Trans-SVNet Gao et al. (2021)	ResNet-50 (BN)	87.2 ±9.3	88.0 ±6.7	87.5 ±5.5	74.7 ±7.7
UATD Ding et al. (2023)	Inception	87.6 ±8.7	88.2 ±7.4	87.9 ±9.6	75.7 ±9.5
CMTNet Yue et al. (2023)	ResNet-50 (BN)	88.2 ±9.2	88.3 ±7.8	88.7 ±6.2	76.1 ±9.2
DACAT Yang et al. (2025a)	ConvNeXt-T	91.3 ±9.3	90.8 ±7.6	90.6 ±6.7	80.7 ±8.8
SurgicalMamba (Ours)	ConvNeXt-T	92.2 ±8.8	91.8 ±7.2	91.4 ±7.7	83.3 ±9.7
	ResNet-50 (GN)	**92.7** ±7.4	**92.3** ±4.7	**92.3** ±6.2	**83.7** ±9.2
AutoLaparo (strict)					
LoViT Liu et al. (2025)	ViT	81.4 ±7.6	85.1	65.9	55.9
SKiT Liu et al. (2023)	ViT	82.9 ±6.8	81.8	70.1	59.9
Surgformer Yang et al. (2024)	ViT	85.7 ±6.9	82.3	75.7	66.7
MTTR-Net Huang et al. (2025)	ResNet-50 (BN)	85.4 ±9.2	78.8	**76.8**	65.6
DACAT Yang et al. (2025a)	ConvNeXt-T	87.8 ±7.6	78.5	75.0	66.9
SurgicalMamba (Ours)	ConvNeXt-T	**89.5** ±6.8	**90.6**	76.2	**68.9**
	ResNet-50 (GN)	85.7 ±8.8	76.8	76.3	66.0

4.5 Qualitative analysis

Figure 3 shows predicted phase timelines for four test videos, comparing SurgicalMamba against the ground truth, the no-λ and no-rotation ablations, and the strongest baselines. Both baselines over-segment, fragmenting the opening *Prep* phase and the long AutoLaparo stretches, while our predictions stay contiguous. The two ablations fail in different places: removing λ reproduces the early-onset instability in (a), the transition it is supervised to sharpen, whereas removing Z fragments the *ClipCut*/*GBDiss* transition, where two visually similar phases meet.

Table 2. Component ablation on Cholec80 (%, mean±std over 40 videos), under both protocols. Shaded rows are controls: *Vanilla Mamba2* is a single-path baseline and *+ Z only* adds state regramming alone. The middle rows remove one component from the *Full* model. Best in bold.

	Relaxed (10-s boundary)				Strict			
Variant	Accuracy	Precision	Recall	Jaccard	Accuracy	Precision	Recall	Jaccard
Vanilla Mamba2	93.82 ±5.09	92.02 ±6.30	90.47 ±10.35	83.37 ±10.62	92.45 ±5.23	86.97 ±11.50	86.16 ±12.38	77.69 ±13.57
+ Z only	94.70 ±4.29	92.46 ±5.10	93.05 ±6.11	85.08 ±9.25	93.16 ±4.48	86.97 ±9.78	88.84 ±7.44	79.21 ±12.11
w/o rotation Z	95.60 ±4.20	92.97 ±4.25	93.71 ±7.95	87.05 ±9.05	94.01 ±4.28	88.07 ±7.68	89.77 ±9.82	80.71 ±12.07
w/o intensity λ	95.08 ±6.07	93.37 ±4.47	93.49 ±5.04	86.70 ±8.75	93.69 ±6.05	88.72 ±8.06	89.26 ±7.30	81.12 ±11.99
w/o fast path	95.85 ±4.03	94.28 ±4.03	93.47 ±8.65	88.03 ±9.44	94.27 ±4.14	88.92 ±8.91	88.67 ±10.84	81.24 ±12.69
w/o smooth loss	95.84 ±3.69	93.54 ±4.71	94.43 ±5.41	87.70 ±8.48	94.38 ±3.82	88.55 ±8.59	90.36 ±7.80	81.80 ±11.81
Full	**96.05** ±3.55	**94.91** ±4.24	**94.38** ±6.16	**88.48** ±8.14	**94.61** ±3.71	**89.59** ±8.70	**90.48** ±8.11	**82.73** ±11.50



Figure 3. Predicted phase timelines on four test videos (frame index on the horizontal axis, phases colour-coded). Ablation rows are shown for Cholec80, where both mechanisms are trained. The same comparison over the whole test set is in Appendix G.

Appendix G reports phase-level F_1 and the confusion matrix over the whole test set, and there the two ablations differ in where the loss lands. Removing Z concentrates it on one phase, *Clean* dropping 4.23 F_1 points through being read as *GBDiss*, a phase that has already occurred, while confusions with phases still to come fall. Removing λ spreads it instead over the closing phases, which follow one another within minutes. The per-video figures divide them the same way, the spread of macro-F_1 rising to 6.81 without Z against 7.81 without λ.

5 Conclusion

In this work we addressed two properties of surgical video that the SSD recurrence does not accommodate. *State regramming* applies a content-conditioned rotation to the carried state at each chunk boundary, so that visually similar frames from separate phases occupy different directions of the state. *Intensity-modulated stepping* increases the decay at the annotated phase boundaries, so that a finished phase is cleared where it ends and the decay within phases can lean toward retention. Both preserve N-semiseparability and $O(d)$ per-frame cost, so the adaptation keeps the streaming profile that motivated using SSD in the first place. Across seven online SPR benchmarks the model matches or exceeds the state of the art in accuracy and Jaccard at 312.88 fps, and the ablations separate what each mechanism contributes.

REFERENCES

Andrei team. Surgical workflow analysis in the SensorOR (EndoVis 2017 sub-challenge participant). https://endovissub2017-workflow.grand-challenge.org/, 2017.

Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *ICML*, 2016.

Nicolás Ayobi, Santiago Rodríguez, Alejandra Pérez, Isabela Hernández, Nicolás Aparicio, Eugénie Dessevres, Sebastián Peña, Jessica Santander, Juan Ignacio Caicedo, Nicolás Fernández, et al. Pixel-wise recognition for holistic surgical scene understanding. *Medical Image Analysis*, pp. 103726, 2025.

Sebastian Bodenstedt, Martin Wagner, Darko Katić, Patrick Mietkowski, Benjamin Mayer, Hannes Kenngott, Beat Müller-Stich, Rüdiger Dillmann, and Stefanie Speidel. Unsupervised temporal context learning using convolutional neural networks for laparoscopic workflow analysis, 2017. URL https://arxiv.org/abs/1702.03684.

Rui Cao, Jian Wang, and Yun-Hui Liu. SR-Mamba: Effective surgical phase recognition with state space model. *arXiv preprint arXiv:2407.08333*, 2024.

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

Tobias Czempiel, Magdalini Paschali, Matthias Keicher, Walter Simson, Hubertus Feussner, Seong Tae Kim, and Nassir Navab. TeCNO: Surgical phase recognition with multi-stage temporal convolutional networks. In *MICCAI*, pp. 343–352, 2020.

Tobias Czempiel, Magdalini Paschali, Daniel Ostler, Seong Tae Kim, Benjamin Busam, and Nassir Navab. OperA: Attention-regularized transformers for surgical phase recognition. In *MICCAI*, pp. 604–614, 2021.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *ICML*, 2024.

Xinpeng Ding, Xinjian Yan, Zixun Wang, Wei Zhao, Jian Zhuang, Xiaowei Xu, and Xiaomeng Li. Less is more: Surgical phase recognition from timestamp supervision. *IEEE Transactions on Medical Imaging*, 42(6):1897–1910, 2023.

Dylan team. Surgical workflow analysis in the SensorOR (EndoVis 2017 sub-challenge participant). https://endovissub2017-workflow.grand-challenge.org/, 2017.

Wenpei Fan, Yaonan Wang, Licheng Liu, Jiayi Zeng, and Min Liu. CMANet: A TCN-RMamba-attention network for surgical phase online recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 36(2):1460–1472, 2026. doi: 10.1109/TCSVT.2025.3599391.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *ICCV*, 2019.

Isabel Funke, Dominik Rivoir, and Stefanie Speidel. Metrics matter in surgical phase recognition. *arXiv preprint arXiv:2305.13961*, 2023.

Xiaojie Gao, Yueming Jin, Yonghao Long, Qi Dou, and Pheng-Ann Heng. Trans-SVNet: Accurate phase recognition from surgical videos via hybrid embedding aggregation transformer. In *MICCAI*, pp. 593–603, 2021.

Carly R Garrow, Karl-Friedrich Kowalewski, Linhong Li, Martin Wagner, Mona W Schmidt, Sandy Engelhardt, Daniel A Hashimoto, Hannes G Kenngott, Sebastian Bodenstedt, Stefanie Speidel, et al. Machine learning for surgical phase recognition: a systematic review. *Annals of Surgery*, 273(4):684–693, 2021.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *COLM*, 2024.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *ICLR*, 2022.

Daniel A Hashimoto, Guy Rosman, Daniela Rus, and Ozanan R Meireles. Artificial intelligence in surgery: promises and perils. *Annals of Surgery*, 268(1):70–76, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Zhuohong He, Ali Mottaghi, Aidean Sharghi, Muhammad Abdullah Jamal, and Omid Mohareri. An empirical study on activity recognition in long surgical videos. In *Machine Learning for Health (ML4H)*, pp. 356–372, 2022.

Kyle Helfrich, Devin Willmott, and Qiang Ye. Orthogonal recurrent neural networks with scaled Cayley transform. In *ICML*, pp. 1969–1978, 2018.

Kaide Huang, Xiang-Lei Yuan, Rui-De Liu, Lian-Song Ye, Yao Zhou, Bing Hu, and Zhang Yi. Multi-teacher temporal regulation network for surgical workflow recognition. *IEEE Transactions on Medical Imaging*, 44(11):4690–4703, 2025.

Yueming Jin, Qi Dou, Hao Chen, Lequan Yu, Jing Qin, Chi-Wing Fu, and Pheng-Ann Heng. SV-RCNet: workflow recognition from surgical videos using recurrent convolutional network. *IEEE Transactions on Medical Imaging*, 37(5):1114–1126, 2017.

Yueming Jin, Huaxia Li, Qi Dou, Hao Chen, Jing Qin, Chi-Wing Fu, and Pheng-Ann Heng. Multi-task recurrent convolutional network with correlation loss for surgical video analysis. *Medical Image Analysis*, 59:101572, 2020.

Yueming Jin, Yonghao Long, Cheng Chen, Zixu Zhao, Qi Dou, and Pheng-Ann Heng. Temporal memory relation network for workflow recognition from surgical video. *IEEE Transactions on Medical Imaging*, 40(7):1911–1923, 2021.

Li Jing, Yichen Shen, Tena Dubcek, John Peurifoy, Scott Skirlo, Yann LeCun, Max Tegmark, and Marin Soljačić. Tunable efficient unitary neural networks (EUNN) and their application to RNNs. In *ICML*, 2017.

Aakash Sunil Lahoti, Kevin Li, Berlin Chen, Caitlin Wang, Aviv Bick, Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *ICLR*, 2026.

Mario Lezcano-Casado and David Martınez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *ICML*, pp. 3794–3803, 2019.

Yang Liu, Jiayu Huo, Jingjing Peng, Rachel Sparks, Prokar Dasgupta, Alejandro Granados, and Sebastien Ourselin. SKiT: a fast key information video transformer for online surgical phase recognition. In *ICCV*, pp. 21017–21027, 2023.

Yang Liu, Maxence Boels, Luis C Garcia-Peraza-Herrera, Tom Vercauteren, Prokar Dasgupta, Alejandro Granados, and Sebastien Ourselin. LoViT: Long video transformer for surgical phase recognition. *Medical Image Analysis*, 99:103366, 2025.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *CVPR*, 2022.

Lena Maier-Hein, Swaroop S Vedula, Stefanie Speidel, Nassir Navab, Ron Kikinis, Adrian Park, Matthias Eisenmann, Hubertus Feussner, Germain Forestier, Stamatia Giannarou, et al. Surgical data science for next-generation interventions. *Nature Biomedical Engineering*, 1(9):691–696, 2017.

Lena Maier-Hein, Martin Wagner, Tobias Ross, Annika Reinke, Sebastian Bodenstedt, Peter M Full, Hellena Hempe, Diana Mindroc-Filimon, Patrick Scholz, Thuy Nuong Tran, et al. Heidelberg colorectal data set for surgical data science in the sensor operating room. *Scientific Data*, 8(1): 101, 2021.

Pietro Mascagni, Deepak Alapatt, Luca Sestini, Maria S Altieri, Amin Madani, Yusuke Watanabe, Adnan Alseidi, Jay A Redan, Sergio Alfieri, Guido Costamagna, et al. Computer vision in surgery: from potential to clinical value. *npj Digital Medicine*, 5(1):163, 2022.

Zakaria Mhammedi, Andrew Hellicar, Ashfaqur Rahman, and James Bailey. Efficient orthogonal parametrisation of recurrent neural networks using Householder reflections. In *ICML*, 2017.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *ICML*, pp. 26670–26698, 2023.

Bo Qi, Xianpeng Qin, Jian Liu, Yu Xu, and Yan Chen. A deep architecture for surgical workflow recognition with edge information. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1358–1364, 2019.

Dominik Rivoir, Isabel Funke, and Stefanie Speidel. On the pitfalls of batch normalization for end-to-end video learning: a study on surgical workflow analysis. *Medical Image Analysis*, 94: 103126, 2024.

Robin team. Surgical workflow analysis in the SensorOR (EndoVis 2017 sub-challenge participant). `https://endovissub2017-workflow.grand-challenge.org/`, 2017.

Klaus Schoeffmann, Mario Taschwer, Stephanie Sarny, Bernd Münzer, Manfred Jürgen Primus, and Doris Putzgruber. Cataract-101: video dataset of 101 cataract surgeries. In *ACM Multimedia Systems Conference (MMSys)*, pp. 421–425, 2018.

Qingyao Tian, Huai Liao, Xinyan Huang, Bingyu Yang, Dongdong Lei, Sebastien Ourselin, and Hongbin Liu. EndoMamba: an efficient foundation model for endoscopic videos via hierarchical pre-training. In *MICCAI*, pp. 224–234, 2025.

Andru P Twinanda, Sherif Shehata, Didier Mutter, Jacques Marescaux, Michel De Mathelin, and Nicolas Padoy. EndoNet: a deep architecture for recognition tasks on laparoscopic videos. *IEEE Transactions on Medical Imaging*, 36(1):86–97, 2016a.

Andru P Twinanda, Sherif Shehata, Didier Mutter, Jacques Marescaux, Michel De Mathelin, and Nicolas Padoy. Workshop and challenges on modeling and monitoring of computer assisted interventions (M2CAI). `http://camma.u-strasbg.fr/m2cai2016/`, 2016b.

Martin Wagner, Beat-Peter Müller-Stich, Anna Kisilenko, Duc Tran, Patrick Heger, Lars Mündermann, David M Lubotsky, Benjamin Müller, Tornike Davitashvili, Manuela Capek, et al. Comparative validation of machine learning algorithms for surgical workflow and skill analysis with the HeiChole benchmark. *Medical Image Analysis*, 86:102770, 2023.

Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Zhengrong Zuo, Changxin Gao, and Nong Sang. OadTR: Online action detection with transformers. In *ICCV*, pp. 7565–7575, 2021.

Youjin Wang, Yangjingyi Chen, Jiahao Yan, Jiaxuan Lu, and Xiao Sun. MemMamba: Rethinking memory patterns in state space model, 2025. URL `https://arxiv.org/abs/2510.03279`.

Ziyi Wang, Bo Lu, Yonghao Long, Fangxun Zhong, Tak-Hong Cheung, Qi Dou, and Yunhui Liu. AutoLaparo: A new dataset of integrated multi-tasks for image-guided surgical automation in laparoscopic hysterectomy. In *MICCAI*, pp. 486–496, 2022.

Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, pp. 3–19, 2018.

Tong Xia and Fucang Jia. Against spatial–temporal discrepancy: Contrastive learning-based network for surgical workflow recognition. *International Journal of Computer Assisted Radiology and Surgery*, 16(5):839–848, 2021.

Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Xia, Zhuowen Tu, and Stefano Soatto. Long short-term transformer for online action detection. In *NeurIPS*, volume 34, pp. 1086–1099, 2021.

Kaixiang Yang, Qiang Li, and Zhiwei Wang. DACAT: Dual-stream adaptive clip-aware time modeling for robust online surgical phase recognition. In *ICASSP*, pp. 1–5, 2025a.

Shu Yang, Luyang Luo, Qiong Wang, and Hao Chen. Surgformer: Surgical transformer with hierarchical temporal attention for surgical phase recognition. In *MICCAI*, pp. 606–616, 2024.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated Delta Networks: Improving Mamba2 with delta rule, 2025b. URL https://arxiv.org/abs/2412.06464.

Fangqiu Yi, Yanfeng Yang, and Tingting Jiang. Not end-to-end: Explore multi-stage architecture for online surgical phase recognition. In *ACCV*, pp. 417–432, 2022.

Wenxi Yue, Hongen Liao, Yong Xia, Vincent Lam, Jiebo Luo, and Zhiyong Wang. Cascade multi-level transformer network for surgical workflow analysis. *IEEE Transactions on Medical Imaging*, 42(10):2817–2831, 2023.

Xiangning Zhang, Qingwei Zhang, Jinnan Chen, Chengfeng Zhou, Yaqi Wang, Zhengjie Zhang, Xiaobo Li, and Dahong Qian. SPRMamba: surgical phase recognition for endoscopic submucosal dissection with Mamba. *IEEE Sensors Journal*, 2026.

APPENDIX CONTENTS

A Preliminaries and notation 15

B Analysis of intensity-modulated stepping 15

C Matrix view of state regramming 18

D Implementation details 21

E Hyperparameter sensitivity 21

F Additional benchmark results 21

G What each mechanism repairs 23

H Does the mechanism transfer beyond surgery? 24

A PRELIMINARIES AND NOTATION

Continuous-time SSM and zero-order hold. The slow path discretizes a continuous-time linear SSM,

$$\frac{\mathrm{d}h(t)}{\mathrm{d}t} = A\,h(t) + B(t)\,x(t), \qquad y(t) = C(t)\,h(t), \tag{9}$$

with A constrained for stability. Holding the input constant over a step of length Δ integrates (9) exactly and gives

$$h_n = \bar{A}\,h_{n-1} + \bar{B}\,x_n, \qquad \bar{A} = \exp(\Delta A), \quad \bar{B} \approx \Delta B. \tag{10}$$

Since A is a per-head scalar, $\bar{A}$ is an element-wise decay, and Δ is the only quantity setting how much of h_{n-1} survives a step. §3.3 reads λ as a change of the variable being integrated, which is why the same zero-order hold applies with Δ replaced by $\alpha_n\Delta$.

Notation. The inner dimension splits into H heads of width P, so $d_{\text{inner}} = H\,P$, and a clip of L frames into $n_c = L/C_{\text{chunk}}$ chunks of size C_{chunk}. The carried state is one tensor in $\mathbb{R}^{H\times P\times N}$ per clip boundary, and it is this tensor that state regramming rotates (§3.5).

Two properties we rely on. First, the chunked training matmul and the frame-by-frame inference recurrence compute the same function, so we train on chunked clips and deploy per-frame with identical weights. Second, a per-head scalar $\bar{A}$ commutes with any orthogonal matrix acting on the state, so decay and rotation may be applied in either order; this is what lets a boundary rotation be inserted without disturbing the exponential decay, and what keeps the result analyzable in the SSD matrix framework (Appendix C).

B ANALYSIS OF INTENSITY-MODULATED STEPPING

Phase durations within a procedure differ by an order of magnitude (Fig. 1A), so whether a past frame belongs to the current phase is not determined by how long ago it was written, and forgetting has to be placed at the boundaries rather than spread by age. This appendix states what the intensity λ does to the slow-path kernel (§B.2), how far its effect reaches and why its range is capped (§B.3), and why its target is asymmetric (§B.4). §B.5 separates what holds exactly from what rests on a model.

B.1 SETUP

By (3) the slow path discretizes with $\bar{A}_n = \exp(\alpha_n\Delta A)$ and $\bar{B}_n \approx \alpha_n\Delta B_n$, where $\alpha_n = 1+\lambda_n \in [1,2]$. Writing $a := |A|\Delta > 0$ (recall $A<0$) gives $\bar{A}_n = e^{-a\alpha_n}$, and the fraction of the input written at step s that survives at step $n > s$ is the product of the intervening decays:

$$R(s,n) := \prod_{k=s+1}^{n} \bar{A}_k = \underbrace{e^{-a(n-s)}}_{\text{decay by age}} \cdot \underbrace{e^{-a\Lambda(s,n)}}_{\text{extra, from }\lambda}, \qquad \Lambda(s,n) := \sum_{k=s+1}^{n} \lambda_k. \tag{11}$$

Two readings of (11) are used repeatedly. Since $\lambda \geq 0$, the second factor never exceeds one: λ can only add forgetting. And what it adds depends on the *total amount* of λ accumulated between s and n, never on how high λ rose at any single frame. Note also that α_n multiplies the input gain $\bar{B}_n \approx \alpha_n\Delta B_n$, not only the decay.

We write $\tau_a := 1/a$ for the memory horizon of a head, in frames: under unmodulated steps, content decays by e^{-1} over τ_a frames. Since A is learned per head, a and τ_a differ across heads, and every statement below holds head by head. For tractability we hold Δ constant, so that α_n is the only source of per-frame variation in the decay; §B.5 returns to what that means for the argument.

B.2 A BURST ATTACHES FORGETTING TO THE BOUNDARY

Each boundary crossed multiplies what remains by a fixed factor, and no fixed step can do this.

Let λ rise to a burst around each boundary $b_1 < b_2 < \dots$ and be zero in between, as its target is. Write M for the amount of λ in one burst, W for its width, and $N(s, n)$ for how many boundaries fall in $(s, n]$. For s and n lying outside the bursts, $\Lambda(s, n) = M\, N(s, n)$, and (11) becomes

$$R(s, n) = \underbrace{e^{-a(n-s)}}_{\text{age}} \cdot \underbrace{\gamma^{N(s,n)}}_{\text{boundaries}}, \qquad \gamma := e^{-aM}. \tag{12}$$

The first factor is ordinary decay by age. The second is new: each boundary crossed multiplies what remains by γ, a fixed survival fraction that does not depend on when the crossing occurred or on how long the phases on either side were. The strength of the whole mechanism is one number, the burst size M, through $\gamma = e^{-aM}$.

Two things follow immediately. A component of λ present everywhere, say $\lambda \equiv c$, only turns a into $a(1 + c)$, which is indistinguishable from running the unmodulated model with $\Delta \mapsto (1 + c)\Delta$; in particular $\lambda \equiv 1$ is *exactly* the unmodulated model at twice the step. A small nonzero floor in the learned λ therefore only rescales the step, and the mechanism lives entirely in the contrast between burst and background. Second, if two boundaries fall closer together than a burst is wide, adjacent bursts overlap and λ no longer returns to zero between them. By the same argument the contrast shrinks, and in the limit the mechanism becomes a rescaled step. The analysis therefore assumes bursts narrower than the shortest phase, $W < L_{\min}$. Where a phase is shorter, the contrast it can draw on shrinks accordingly.

This boundary factor is what separates the phases. Consider two stretches of the same length ℓ, $(s, n]$ and $(s', n']$ with $n - s = n' - s' = \ell$, both with ends outside the bursts, of which only the second contains a boundary. The first describes how an input written ℓ steps earlier survives when it belongs to the current phase, $R_{\text{within}}(\ell) := R(s, n)$, the second when it belongs to the previous one, $R_{\text{across}}(\ell) := R(s', n')$. We call their ratio $S := R_{\text{within}}(\ell)/R_{\text{across}}(\ell)$ the separation.

Proposition 1 (Selectivity)**.** *For every age ℓ,*

$$S = \frac{R_{\textit{within}}(\ell)}{R_{\textit{across}}(\ell)} = \begin{cases} 1 & \textit{under a fixed step, for every rate } a, \\ \gamma^{-1} = e^{aM} = e^{M/\tau_a} & \textit{under } (12). \end{cases}$$

Under a fixed step both survival factors are $e^{-a\ell}$; under (12) they share the age term and differ by one power of γ. The first case is an identity: a kernel that depends only on how many steps have elapsed cannot treat the two inputs differently, whatever its rate. Whether an input of age ℓ has crossed a boundary depends on how long ago that boundary was, which such a kernel does not see. The information has to enter from outside the age, and supervising λ toward the annotated boundaries (§3.6) is how it enters. The second case is what it buys: a separation that grows exponentially with the burst mass measured in memory horizons.

A fixed step could reach the same suppression only by raising its rate everywhere. To suppress pre-boundary content by the factor γ at age ℓ, it needs $e^{-a'\ell} = e^{-a(\ell+M)}$, that is

$$a' \;=\; a\Big(1 + \frac{M}{\ell}\Big). \tag{13}$$

That rate applies to every input, so the current phase's own content then decays with an exponent $1 + M/\ell$ times larger, and the price is highest just after the boundary, where ℓ is small. A burst attaches the same suppression to the boundary and leaves content written outside it untouched.

B.3 The effect of a burst is bounded by its width

Since $\lambda \le 1$, a burst of width W adds at most the decay of W extra frames. This bounds both what it removes at a boundary and what it removes by mistake.

Hence $M \le W$, and the separation of Proposition 1 is bounded by

$$S = e^{aM} \le e^{W/\tau_a}. \tag{14}$$

Halving the content from before a boundary, relative to content of the same age written after it, needs $M = \tau_a \log 2$: a burst comparable to the memory horizon.

The same bound limits the damage of a false detection: a burst raised where there is no boundary keeps at least e^{-W/τ_a} of whatever it crosses, nearly all of it in heads with $\tau_a \gg W$. A false detection therefore leaves the long context of the slow heads intact, which matters over an operation of several hours. A boundary leak independent of a, such as $\log \bar{A}_n = -a - \kappa\lambda_n$, would lose this guarantee: one false detection would erase that context.

The two ends of $1 \le \alpha \le 2$ have separate roles. The upper end spreads forgetting over several frames instead of erasing the state in one step ($\alpha \to \infty$ gives $\bar{A} \to 0$). The lower end prevents freezing the state ($\bar{A} \to 1$) and makes $\lambda = 0$ recover the unmodulated model exactly. Raising the upper end to $1 + \lambda_{\max}$ would only replace W by $\lambda_{\max} W$ in (14).

B.4 Why the window is asymmetric

Forgetting at a boundary has to cost some phase, the one ending or the one beginning. The target puts the cost on the one beginning.

The target of §3.6 rises with width σ_L before the boundary and decays with width σ_R after it. With $u = t - b$,

$$\lambda(u) = \begin{cases} \exp(-u^2/2\sigma_L^2), & u < 0, \\ \exp(-u^2/2\sigma_R^2), & u \ge 0, \end{cases} \qquad M_{L,R} = \int \lambda = \sqrt{\tfrac{\pi}{2}}\,\sigma_{L,R}, \quad M = M_L + M_R. \tag{15}$$

By Proposition 1, how much is forgotten depends only on the total mass M; the split between the two sides decides which phase pays for it. Three arguments put the mass after the boundary. The first holds without assumptions; the other two use the energy model of §B.5.

A transition can only be seen after it happens. λ is predicted from past frames (§3.3), so mass before the boundary asks the model to raise λ before any evidence of the transition exists, whereas mass after it asks the model to recognize evidence already present. If the true transition is $b + \epsilon$ with $\epsilon \sim p$, the pointwise-optimal target is $\mathbb{E}_\epsilon[\delta(u - \epsilon)] = p(u)$: a soft target is the right one under boundary uncertainty, and p is skewed to the right.

The ending phase has more to lose. Just before the boundary the state is full of the ending phase, $E \approx \beta^2\tau_a/2$, while just after it the new phase has barely been written, $E(u) \approx \beta^2 u$. Mass placed where the state holds more destroys more; integrating $\mathrm{d}D \approx 2a\,E\,\mathrm{d}M$ over each side gives

$$D_L \approx \sqrt{\tfrac{\pi}{2}}\,\beta^2\sigma_L, \qquad D_R \approx \frac{2\beta^2\sigma_R^2}{\tau_a}. \tag{16}$$

Damage on the left grows with σ_L from the start, whereas damage on the right starts from zero and stays small while σ_R is small compared with τ_a. Damage to the ending phase is also permanent, since that phase is over, whereas the new phase continues and refills.

The extra write gain helps only the new phase. While λ is raised, inputs are also written harder (§B.1). After the boundary this helps establish the new phase; before it, it strengthens the last frames of a phase about to be discarded. An input written at $u \ge 0$ is amplified by α_u but then crosses the rest of the burst, which leaves it with the net factor

$$g(u) = \alpha_u\, e^{-aM_R(u)}, \qquad M_R(u) := \int_u^\infty \lambda(v)\,\mathrm{d}v. \tag{17}$$

For the input written at the peak ($u = 0$, $\alpha = 2$) to stay amplified, $2e^{-aM_R} > 1$, which bounds

$$\sigma_R < \frac{\log 2}{\sqrt{\pi/2}}\,\tau_a. \tag{18}$$

Together the three arguments give $\sigma_L \ll \sigma_R$, with σ_R on the scale of the memory horizon.

B.5 What is established, and what is not

Equation (11), the form (12) for s and n outside the bursts, Proposition 1, the degeneracy of a constant λ, the rate (13) a fixed step would need, and the bound (14), including that on a false detection, hold exactly under the constant-Δ setup of §B.1.

The comparisons of §B.4 other than causality use an energy model: mutually incoherent inputs of constant per-frame energy β^2, a state energy that accumulates as $E(u) \approx \beta^2 \min(u, \tau_a/2)$, and damage measured as energy loss, whereas the read-out C_t selects directions; (17) further reads the gain as a uniform scaling, exact only for $\bar{B}_n \approx \alpha_n \Delta B_n$. They indicate the direction and scale of the asymmetry rather than its exact value.

None of this is stated in units of task loss. That the selectivity improves predictions is measured by the ablation of §4.3 and the error analysis of Appendix G, not proved here. Nor does the appendix isolate λ as the only way to obtain this schedule. Holding Δ fixed is what lets its effect be read off, and a selective Δ could in principle vary in the same way. In the model itself λ multiplies that input-dependent step rather than replacing it, and what the mechanism contributes is the supervision that ties forgetting to event structure (§2.3).

C MATRIX VIEW OF STATE REGRAMMING

The same views recur throughout an operation (Fig. 1B), so a write direction chosen from the current frame alone places a recurring scene on top of its earlier occurrence. State regramming rotates the carried state at each chunk boundary, so that where a frame sits depends on what has passed since it was written. This appendix shows how the boundary rotations enter the SSD kernel (§C.2), how they change what a read sees (§C.3), that they preserve the structure of SSD (§C.4), and why their planes are chosen by the content (§C.5); §C.6 relates them to the rotations of Mamba-3.

C.1 SETUP

Fix a single head with per-head scalar $A \in \mathbb{R}_{<0}$ and per-head width $P = 1$; the general case is recovered by stacking. Frames carry a global index n, the discrete decay at frame n is $\bar{a}_n = \exp(\Delta_n A)$, and

$$a(t{:}s) \;:=\; \prod_{n=s+1}^{t} \bar{a}_n, \qquad a(t{:}t) = 1. \tag{19}$$

A sequence is partitioned into chunks of C_{chunk} frames, and $c(t)$ denotes the chunk containing frame t. The boundary rotation $Z^{(c)} \in SO(N)$ applied at the end of chunk c is the Cayley map of §3.5. Following the boundary update of §3.5, we use row vectors: the recurrence is

$$h_t = \bar{a}_t \, h_{t-1} + x_t \, B_t^\top, \qquad y_t = h_t \, C_t, \tag{20}$$

and $h \leftarrow h \, Z^{(c)}$ at the end of chunk c.

C.2 HOW BOUNDARY ROTATIONS ENTER THE KERNEL

Each boundary crossed adds one rotation between the write and the read, in order.

An input x_s enters the state as $x_s B_s^\top$. Every later frame multiplies it by a scalar decay, and every chunk boundary it crosses multiplies it on the right by that boundary's rotation. Since a scalar commutes with any matrix, the decays collect into $a(t{:}s)$ and the rotations into their ordered product, so the map $y = Mx$ has entries

$$M[t,s] = a(t{:}s) \; B_s^\top \, \mathcal{Z}_{c(s),c(t)} \, C_t, \qquad \mathcal{Z}_{c',c} := Z^{(c')} Z^{(c'+1)} \cdots Z^{(c-1)}, \quad \mathcal{Z}_{c,c} := I, \tag{21}$$

for $s \le t$, and $M[t,s] = 0$ otherwise. Within a chunk $\mathcal{Z} = I$ and (21) is the vanilla SSD kernel. Stacking the rows $C_t^\top$ of chunk c into $C^{(c)}$ and those of B into $B^{(c')}$, the block of M from chunk c' to chunk $c \ge c'$ is

$$M_{cc'} = L^{(c,c')} \odot \Big(C^{(c)} \, \mathcal{Z}_{c',c}^\top \, B^{(c')\top} \Big), \qquad L^{(c,c')}[t,s] = a(t{:}s). \tag{22}$$

M is block lower-triangular, and each step away from the main diagonal adds one boundary rotation (Table 3).

Written out,

$$B_s^\top \, \mathcal{Z} \, C_t = \sum_{i,j} B_{s,i} \, \mathcal{Z}_{ij} \, C_{t,j}, \tag{23}$$

Table 3. Rotation factor $\mathcal{Z}_{c',c}$ in each block $M_{cc'}$ for four chunks. Diagonal blocks reduce to vanilla SSD; each sub-diagonal step adds one boundary rotation.

$c \setminus c'$	0	1	2	3
0	I	0	0	0
1	$Z^{(0)}$	I	0	0
2	$Z^{(0)}Z^{(1)}$	$Z^{(1)}$	I	0
3	$Z^{(0)}Z^{(1)}Z^{(2)}$	$Z^{(1)}Z^{(2)}$	$Z^{(2)}$	I

against $\sum_i B_{s,i}C_{t,i}$ in vanilla SSD, which pairs each state dimension only with the read-out component of the same index. Across a boundary, a dimension written in one chunk can be read through any component of a later chunk's C_t, with weights set by the chunks in between.

C.3 Effective read-out

The direction through which a chunk reads an earlier memory is set by the chunks in between.

Grouping the rotation with the read-out in (21),

$$\widetilde{C}_t^{(c,c')} := \mathcal{Z}_{c',c}\, C_t, \qquad M[t,s] = a(t{:}s)\, B_s^\top\, \widetilde{C}_t^{(c(t),c(s))}. \tag{24}$$

Although C_t is computed from frame t alone, the direction through which it reads a memory written in chunk c' is rotated by every boundary crossed since. Componentwise, $\widetilde{C}_{t,i} = \sum_j (\mathcal{Z}_{c',c})_{ij} C_{t,j}$, so the coefficient that retrieves dimension i is a mixture of all components of C_t; this is the pairing of (23) read from the other side.

This is what separates a recurring scene. Let the same scene be written along the same direction B at frames of chunks $c_1 < c_2$, and read at a frame t of a later chunk c. The two occurrences are read through $\mathcal{Z}_{c_1,c}C_t$ and $\mathcal{Z}_{c_2,c}C_t$, which differ by the rotations $\mathcal{Z}_{c_1,c_2} = Z^{(c_1)} \cdots Z^{(c_2-1)}$ applied between them, so a read-out can weight one occurrence without the other. What separates the two is the content of the intervening chunks and, since rotations do not commute in general, its order. Were Z the same at every boundary, $\mathcal{Z}_{c',c} = Z^{c-c'}$ would depend only on the chunk distance, and the rotation would reduce to a chunk-level position encoding (§3.5).

The rotation leaves the rank of the read-out unchanged: what it changes is which stored content a read picks up, not how much it can read.

C.4 Equivalent form and structure preservation

Collected over chunks, the rotations move onto the projections, which shows that the SSD structure is untouched.

The effective read-out attaches a separate rotation to each pair of chunks. Collecting them into the rotation accumulated up to each chunk,

$$Q_0 := I, \qquad Q_{c+1} := Q_c\, Z^{(c)}, \tag{25}$$

gives a form that no longer depends on the pair.

Proposition 2 (Equivalent form and structure preservation)**.** *(i) For all $s \le t$,*

$$M[t,s] = a(t{:}s)\, \big(Q_{c(s)}B_s\big)^\top \big(Q_{c(t)}C_t\big),$$

so state regramming computes the same map as SSD without rotation, with projections $\widetilde{B}_s := Q_{c(s)}B_s$ and $\widetilde{C}_t := Q_{c(t)}C_t$. (ii) Consequently M is N-semiseparable, and the recurrence runs at a per-frame cost independent of elapsed length. (iii) The rotation preserves the norm of the carried state.

For (i), $Q_{c'}\mathcal{Z}_{c',c} = Q_c$ by definition, and $Q_{c'}$ is orthogonal, so $\mathcal{Z}_{c',c} = Q_{c'}^\top Q_c$; substituting into (21) gives the form stated. For (ii), take any submatrix of the lower triangle, with rows $t \ge j$ and columns $s \le j$ for some j. Its entries are $\big(a(t{:}j)\,\widetilde{C}_t\big)^\top \big(a(j{:}s)\,\widetilde{B}_s\big)$, a product of an N-dimensional

factor of t and one of s, so its rank is at most N. This is the defining property of an N-semiseparable matrix Dao & Gu (2024); the recurrent form itself adds one $N \times N$ product per chunk, amortized over its C_{chunk} frames. For (iii), $\|hZ\|_F = \|h\|_F$ for orthogonal Z, so the rotation redirects the carried content without amplifying or attenuating it; how fast old content fades is still set by $a(t{:}s)$ alone.

The equivalent form is a tool for analysis, not an implementation. $Z^{(c)}$ is computed from the chunk output (§3.5), which includes the contribution of the carried state, so Q_c is known only once the scan has reached the end of chunk c, and $\widetilde{B}$, $\widetilde{C}$ cannot be formed ahead of it. Applying $Z^{(c)}$ to the state costs one $N \times N$ product per chunk.

C.5 Rotation planes chosen by the content

The content of a chunk chooses not only how far to rotate but in which planes, so that a recurring scene can be moved out of the subspace of its earlier occurrence.

The planes of $Z^{(c)}$ follow from its construction (Eqs. 6–8). The generator $S^{(c)} = \widetilde{S} - \widetilde{S}^\top$ with $\widetilde{S} = U \operatorname{diag}(\theta)\, V^\top$ has its range in $\operatorname{span}(U^{(c)}, V^{(c)})$, a subspace of dimension at most $2r$ predicted from the chunk's content. On the orthogonal complement of that subspace $S^{(c)}$ vanishes, and so does the rotation: any v orthogonal to it satisfies $vZ^{(c)} = v$. Within it, the skew-symmetric $S^{(c)}$ splits into mutually orthogonal planes, on each of which it acts as a scaled planar rotation, and the Cayley map keeps these planes and turns each by an angle below π. The rotation planes are therefore functions of $U^{(c)}$, $V^{(c)}$ and $\theta^{(c)}$, and vary with the chunk's content. The rank r, chosen for efficiency (Appendix D, with its sweep in Appendix E), limits a single boundary to at most r planes; since each boundary chooses its own, the rotation accumulated over several boundaries need not be confined to them.

Because the planes are chosen rather than fixed, an earlier occurrence need not stay in the subspace it was written to. For any two unit vectors u and w with $w \neq -u$ there is a rotation in the plane they span taking u to w, so the rotations between two occurrences can, in particular, move the earlier one to a direction orthogonal to B.

The rotation does not change the magnitude $a(t{:}s)$: old content still fades at the rate set by the decay. What it changes is the direction in which content is held while it lasts, and a separation in direction does not depend on how old the two occurrences are.

C.6 Relation to Mamba-3

Mamba-3 (Lahoti et al., 2026) also rotates the state of an SSD recurrence. The two rotations are introduced for different reasons and at different points of the recurrence; this section states what each does.

Mamba-3. Mamba-3 makes the state transition complex, with both its real and imaginary parts computed from the input. Written in real form, the transition at step t is a scalar decay times a block-diagonal rotation acting on fixed pairs of state coordinates, with an angle in each pair set by the step and the input, $\Delta_t \theta_t$; the state is thus rotated at every step. To keep the SSD form, the rotation is not applied inside the scan: the accumulated rotations are applied to the B and C projections as a data-dependent rotary embedding, and only the real part of the transition is passed to the SSD kernel. The motivation is expressivity: real eigenvalues can only decay, so a state governed by them cannot represent the rotational dynamics that tasks such as parity require.

State regramming. Our rotation addresses a different need: placing content carried across a long stream according to what has happened since it was written. It is applied once per chunk, to the carried state, and both its planes and its angles are predicted from the chunk output (§3.5). Since that output includes the contribution of the carried state, the rotation cannot be computed before the scan reaches the chunk boundary, and it is applied to the state directly; the SSD kernel within each chunk is unchanged.

Formal relation. Both rotations admit the same kind of rewriting: the accumulated rotation can be moved onto the projections, as in the rotary form of Mamba-3 and Proposition 2 here. They differ in when the rotation acts (every step or once per chunk), in what sets it (the current input, or the chunk output including the carried state), and in whether the input chooses the rotation planes or only the angles within fixed ones.

D IMPLEMENTATION DETAILS

Cross-clip carry. On the slow path the SSM hidden state at the start of a clip is initialized from the previous clip's final state, and the depthwise-convolution buffer is seeded with that clip's last $d_{\text{conv}} - 1$ frames, so the forward pass is identical to running on the concatenated stream while gradients remain truncated at the clip boundary.

Fusion. The fast path's input projection acts on the channel-wise concatenation $[\, x^{\text{conv}}_{\text{fast}} \|_{\text{ch}} y_{\text{slow}} \,]$, which is what makes $(\Delta^{\text{fast}}, B^{\text{fast}}, C^{\text{fast}})$ depend on the slow summary. The two path outputs are combined as $\text{out} = \text{Linear}(\text{RMSNorm}((y_{\text{slow}} + y_{\text{fast}}) \cdot \text{SiLU}(z)))$, where the gating stream z is emitted by the fast input projection alongside x_{fast}. Summing rather than concatenating keeps the SSM inner dimension at the canonical SSD value and avoids doubling the output projection.

Rotation cost. Both MLPs of (6) hold independent weights per head and are evaluated with a single batched einsum. Per head and per chunk, the Cayley map of (8) costs one $N \times N$ solve and the state update one $N \times N$ multiply; both are amortized over C_{chunk} frames, so the per-frame cost is unchanged in order and the added parameters are $O(HPNr)$. The same operation, with separate parameters, runs on both paths.

Visual backbone. Rivoir et al. (2024) showed that BatchNorm impedes end-to-end training in surgical phase recognition and used ConvNeXt Liu et al. (2022) as the visual backbone instead, a choice DACAT Yang et al. (2025a) also follows. We train end-to-end with the same backbone, an ImageNet-pretrained ConvNeXt-Tiny with the bottom two stages frozen and the top two fine-tuned jointly with the temporal model. To separate the temporal model's contribution from the backbone's, we also train Cholec80, M2CAI16 and AutoLaparo with a ResNet-50 (GroupNorm) in its place.

Training. Frames are resized to 250×250 and randomly cropped to 224×224 with standard augmentation. We train end-to-end with AdamW on clips of L=256 frames, chunk size C_{chunk}=64, K=4 blocks, state dimension N=64, and rotation rank r=16; the loss weights of §3.6 are w_{sm}=0.15 and w_{trans}=0.5, and the transition target uses σ_L=2 and σ_R=12 frames. Sensitivity to the four core architectural choices is reported in Appendix E.

E HYPERPARAMETER SENSITIVITY

Each of the four core hyperparameters (rotation rank r, chunk size C_{chunk}, state dimension N, and block count K) is swept on Cholec80 with the other three held at their defaults (r=16, C_{chunk}=64, N=64, K=4), under both the relaxed and strict protocols (Fig. 4). Every sweep stays within 1.0 pp of the default in strict accuracy, well inside the inter-video standard deviation, so none of these choices is load-bearing for the reported results. The rank sweep flattens beyond r=16, which says the rotation needs enough planes to re-express the carried state but not a full-rank map, and this is what makes the low-rank Cayley parameterization (§3.5) affordable at chunk granularity.

F ADDITIONAL BENCHMARK RESULTS

Metrics follow standard SPR conventions Jin et al. (2017); Funke et al. (2023); Huang et al. (2025). Write TP^v_p, FP^v_p and FN^v_p for the true-positive, false-positive and false-negative frame counts of phase p in video v, with P phases and V test videos. Accuracy is computed per video and averaged over videos,

$$\text{Accuracy} = \frac{1}{V} \sum_v \frac{\sum_p TP^v_p}{T^v}, \tag{26}$$

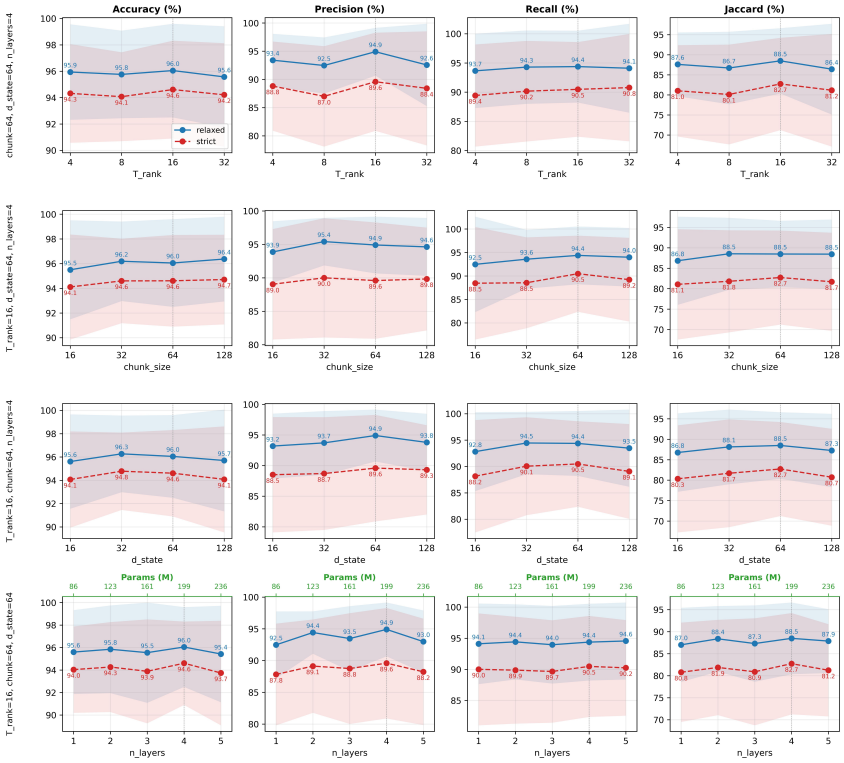


Figure 4. Hyperparameter sensitivity on Cholec80. Each of the rotation rank r, chunk size C_{chunk}, state dimension N, and block count K is swept with the other three fixed at the default (r=16, C_{chunk}=64, N=64, K=4; dotted lines); the K sweep also reports parameter count on its top axis. Solid blue/dashed red are the relaxed/strict protocols; shading is one std over the 40 test videos. Strict-accuracy means vary within 1.0 pp, well inside the inter-video std.

Table 4. Details of the datasets used in our experiments. All splits follow prior work Rivoir et al. (2024); Jin et al. (2021).

Dataset	# classes	# videos	train : val : test
Cholec80 Twinanda et al. (2016a)	7	80	32 : 8 : 40
M2CAI16 Twinanda et al. (2016b)	8	41	20 : 7 : 14
Cataract-101 Schoeffmann et al. (2018)	10	101	63 : 10 : 28
AutoLaparo Wang et al. (2022)	7	21	10 : 4 : 7
HeiChole Wagner et al. (2023)	7	24	12 : 6 : 6
Heidelberg Maier-Hein et al. (2021)	14	30	18 : 6 : 6
GraSP Ayobi et al. (2025)	11	13	6 : 2 : 5

with T^v the number of frames in video v. Precision, recall and Jaccard are computed per phase and then averaged across phases within a video and across videos,

$$\text{Precision} = \frac{1}{PV}\sum_v\sum_p \frac{TP_p^v}{TP_p^v + FP_p^v}, \qquad \text{Recall} = \frac{1}{PV}\sum_v\sum_p \frac{TP_p^v}{TP_p^v + FN_p^v}, \tag{27}$$

$$\text{Jaccard} = \frac{1}{PV}\sum_v\sum_p \frac{TP_p^v}{TP_p^v + FP_p^v + FN_p^v}. \tag{28}$$

This appendix reports the four benchmarks not covered in §4.2. Table 4 first lists all seven datasets used in the paper with their class and video counts and train/val/test splits.

Cataract-101 and HeiChole. Table 5 compares both under their standard protocols. On Cataract-101 the top methods sit within a point of each other in accuracy, so the gain shows up in Jaccard

Table 5. State-of-the-art comparison on Cataract-101 and HeiChole, each model trained per dataset. Best per block in bold; "–" not reported. Both follow their standard protocols.

Method	Accuracy	Precision	Recall	Jaccard
Cataract-101				
Qi et al. Qi et al. (2019)	88.1	–	–	–
He et al. He et al. (2022)	94.5	93.1	91.6	–
MTRCNet-CL Jin et al. (2020)	94.7	92.3	91.8	85.7
RCNeSt Xia & Jia (2021)	95.4	92.6	92.3	85.8
CB-RCNeSt Xia & Jia (2021)	96.4	94.9	94.7	90.2
MTTR-Net Huang et al. (2025)	96.9 ± 2.7	95.7 ± 3.2	96.2 ± 2.2	92.1 ± 4.8
SurgicalMamba (Ours)	$\mathbf{96.9 \pm 3.1}$	$\mathbf{96.2 \pm 2.5}$	$\mathbf{96.7 \pm 2.2}$	$\mathbf{93.2 \pm 3.7}$
HeiChole				
ResNet50 He et al. (2016)	68.5 ± 12.3	66.0	58.0	44.7
SV-RCNet Jin et al. (2017)	70.2 ± 11.6	67.5	59.6	45.2
TeCNO Czempiel et al. (2020)	78.3 ± 8.8	79.7	69.9	58.3
Trans-SVNet Gao et al. (2021)	78.0 ± 9.6	77.9	68.0	56.4
MTTR-Net Huang et al. (2025)	80.1 ± 10.9	80.3	75.8	64.5
SurgicalMamba (Ours)	$\mathbf{86.4 \pm 8.7}$	**82.0**	**78.0**	**70.3**

Table 6. Comparison on the Heidelberg (HeiCo) dataset. Best in bold.

Method	Accuracy	Jaccard
Dylan et al. Dylan team (2017)	21	8
Andrei et al. Andrei team (2017)	57	25
Robin et al. Robin team (2017)	60	38
Sebastian et al. Bodenstedt et al. (2017)	61	40
MTTR-Net Huang et al. (2025)	70 ± 14	44 ± 20
SurgicalMamba (Ours)	$\mathbf{72.1 \pm 15.1}$	$\mathbf{47.3 \pm 18.1}$

instead. HeiChole has 24 videos from a different institution and considerable domain shift, and improves Jaccard by $+5.8$ pp over the strongest baseline.

Heidelberg (HeiCo). Table 6 reports the 14-phase, 30-video multi-procedure benchmark. It has twice the phase count of Cholec80 and fewer videos per phase, so the segments are short and easily confused with one another. SurgicalMamba improves Jaccard by $+3.3$ pp.

GraSP. Table 7 reports the prostatectomy benchmark under its official phase-level mean Average Precision. SurgicalMamba matches MTTR-Net at 77.7% and reduces the per-video standard deviation from 13.8 to 12.4. Five test videos make this a weak signal, though it points the same way as the per-video spreads on Cholec80 (Table 1).

G What each mechanism repairs

§4.5 reads the two mechanisms off one video. This appendix repeats that reading over the whole Cholec80 test set, on phase-level F_1. Removing either mechanism costs a similar amount, 1.48 macro points for Z and 1.70 for λ (Table 8), and Figure 5 shows that the two losses are not made of the same errors.

Z separates a phase from one already seen. The deficit is concentrated on *Clean* (-4.23), and one cell accounts for most of it: *Clean* read as *GBDiss* runs at 10.8% without Z against 6.2% with it. Among the cells that move by more than a point, the ones that fall are confusions with a phase that has already occurred (*Clean*/*GBDiss* above, *GBPack*/*GBDiss* $4.4 \to 3.0$, *GBRetr*/*Clean* $13.2 \to 11.9$), and the ones that rise are confusions with a phase still to come (*Clip*/*GBDiss* $11.4 \to 14.4$, *Prep*/*Calot* $3.4 \to 5.0$). Z acts on what the carried state already holds, so this is the split §3.5 leads one to expect.

Table 7. Comparison on the GraSP dataset using the official phase-level mean Average Precision (mAP). Best in bold.

Method	mAP
SlowFast Feichtenhofer et al. (2019)	70.7
TAPIR Ayobi et al. (2025)	74.6
TAPIS-VST Ayobi et al. (2025)	70.9
TAPIS Ayobi et al. (2025)	76.1
MTTR-Net Huang et al. (2025)	77.7 ± 13.8
SurgicalMamba (Ours)	**77.7 ± 12.4**

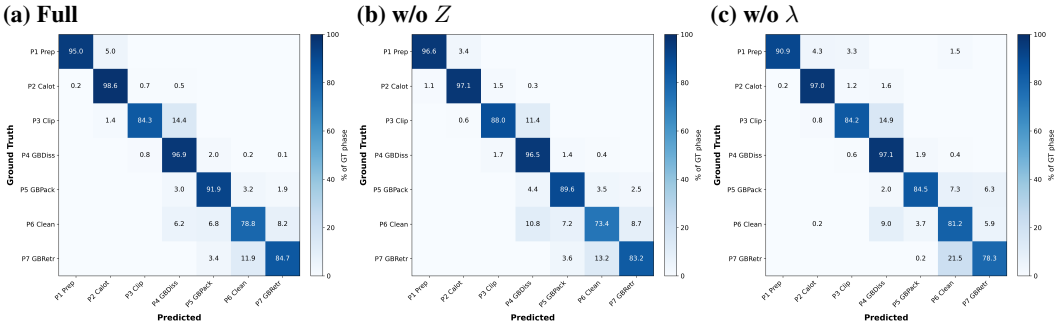


Figure 5. Row-normalized confusion over the Cholec80 test set, as a percentage of each ground-truth phase.

λ keeps the closing phases apart. The deficit is spread instead: λ is ahead on every phase, by at most 3.61. The gain is small on the two long phases (*Calot* 0.79, *GBDiss* 0.78) and on *GBPack* (0.37), and largest on *GBRetr*, *Clean* and *Prep*. The errors behind them are neighbouring phases running past their end: *GBRetr* read as *Clean* at 21.5% without λ against 11.9% with it, *GBPack* as *Clean* at 7.3% against 3.2%, and *GBPack* as *GBRetr* at 6.3% against 1.9%. All three sit in the closing minutes of the procedure, where *GBPack*, *Clean* and *GBRetr* follow one another within a few minutes, and a decay slow enough to hold the long phases does not by itself clear one before the next begins.

The per-video figures separate the two the same way. Removing Z leaves the spread of macro-F_1 across videos at 6.81 against 5.81; removing λ raises it to 7.81. Z costs a phase across videos, λ costs particular videos across phases.

H Does the mechanism transfer beyond surgery?

Multi-query associative recall (MQAR) scatters key–value pairs through a long sequence and asks for them back one at a time, so a run turns on whether the recurrent state kept the pairs apart. It is the diagnostic Dao & Gu (2024) used to argue for a larger recurrent state: if similar keys are written on top of one another, one answer is to keep enlarging the state until they stop colliding. Adding only the boundary rotation asks whether the same state can be put to more efficient use rather than enlarged.

We reuse their setting without modification. At sequence length T the sequence carries $T/4$ key–value pairs, so 256 pairs at $T{=}1024$, and the remaining positions hold random distractors drawn from a vocabulary of 8192, the harder variant in which non-key/value positions are randomized rather than left empty. Training cycles through datasets of $(T/32, T/16, T/8, T/4)$ pairs as a curriculum. Models are two layers deep. The only change we make is the boundary rotation of §3.5, with the chunk size set to $T/4$, so every run crosses three boundaries and applies three rotations whatever the sequence length.

Figure 6 gives all three sequence lengths. The gain is largest where the pairs to be stored outnumber the directions available to hold them, reaching $+27.8$ pp at $N{=}16$, $d_{\text{model}}{=}64$ for $T{=}256$, and

Table 8. Phase-level F_1 on the Cholec80 test set, pooled over all frames, with the per-video macro-F_1 mean and spread below.

Phase	Full	w/o Z	w/o λ
P1 Prep	**96.71**	94.07	94.60
P2 Calot	**98.88**	98.26	98.09
P3 Clip	**87.50**	85.82	85.60
P4 GBDiss	**95.99**	95.62	95.21
P5 GBPack	81.63	**82.11**	81.26
P6 Clean	**84.22**	79.99	81.86
P7 GBRetr	**82.69**	81.40	79.08
Macro	**89.66**	88.18	87.96
Per-video mean	**89.02**	87.57	87.85
Per-video std	**5.81**	6.81	7.81

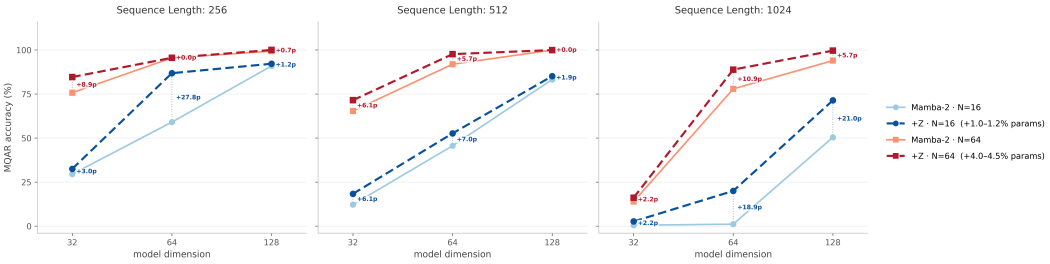


Figure 6. MQAR at sequence lengths 256, 512 and 1024 ($T/4$ key–value pairs each), for state sizes $N \in \{16, 64\}$ and model dimensions $\{32, 64, 128\}$. Labels give the gain from adding Z at matched N. Every configuration improves or is unchanged, the gain going to zero only where the baseline is already saturated (N=64 at d_{model}=128 for $T \in \{256, 512\}$).

+21.0 pp at d_{model}=128 once T=1024 makes the state scarce again at the larger width. It falls to zero where the baseline has already saturated, at N=64 with d_{model}=128 for $T \in \{256, 512\}$.

That shape follows from what the rotation does. By Appendix C the effective write direction of a token becomes $B_s^\top \mathcal{Z}_{c(s),c(t)}$, so two pairs written in different chunks are held in bases separated by the rotations accumulated between them, even when their keys coincide. The N directions are re-used according to where in the sequence a pair was written rather than multiplied, which is worth something only while pairs are competing for them. Added capacity behaves differently, helping wherever it is added, and the rotation costs 1–4.5% more parameters. Enlarging the state remains the more direct remedy here, and the two compose rather than compete.

USE OF AI ASSISTANCE

Large Language Models (LLMs) were used to edit and polish the writing of this paper, tightening prose and checking wording. The authors are responsible for all technical content and for the final text.