TopoContour-Mamba: Topology-Guided Visual Serialization with Augmented Contour Trees

Dian Yu Haoyu Yang

Abstract

Visual state-space models need a sequence before selective scanning, so the route decides which evidence is read early and which remains close to the final output. Most visual routes are fixed across images. We propose TopoContour-Mamba, a fully specified architecture that derives an image-specific route from a deterministic saliency field and reads raw image samples along that route. A full contour tree supplies valid continuation, split, join, source, and terminal events; only useful regular contours are inserted into a sparse execution tree. Each retained contour receives a perimeter-based point budget. Uniform pilot points sample one Sobel field: the mean signed normal response selects one low-saliency read side for the whole contour, the smoothed response magnitude redistributes the fixed point budget, and its maximum fixes the common start/end point p_0. Normal Mamba reads RGB samples from the first collision back to a tagged contour point. Two shared-weight Contour Mamba scans start at p_0, traverse the same closed samples in opposite directions, and end on a tagged return to p_0. A terminal contour reads both normal sides and uses a second pair of loops to combine them. All contours are encoded independently before a Tree Mamba aggregates their tagged outputs from low to high saliency. Splits copy one tagged summary, joins use permutation-invariant support-aware fusion, and terminal leaves are sorted by saliency measured on their actual reads for the final Mamba. The route is deterministic and uses no CNN feature stem. We give a complete algorithmic specification of its routes, token interfaces, boundary cases, defaults, and structural properties without claiming empirical performance.

1 Introduction

Mamba uses input-dependent state updates but still needs a sequence order before scanning [5]. Vision variants commonly use raster, directional, windowed, or otherwise predesigned traversals [17, 13, 16, 10]. These routes are efficient, but they need not follow the structure of the current image.

Our starting intuition is simple. Human inspection does not assign equal attention to every region: broad context may be registered first, while distinctive details receive later and denser inspection. We do not claim a biological model. We use this idea only to schedule evidence for a selective state-space model. Low saliency enters the global history earlier, high saliency enters later, and rapid changes in the route field receive denser samples. The full order is computed before each Mamba scan.

TopoContour-Mamba builds this order from a scalar saliency field. Closed level-set components become local routes, while a contour tree records how they appear, continue, split, join, and terminate as saliency rises [4]. The tree replaces nearest-contour matching, which can create false long-range links around basins. Fine branches can remain in the topology without receiving scan tokens.

The initial model uses a traditional saliency operator for routing and samples visual content directly from the RGB image. It has no CNN feature stem. A single Sobel field makes three contour-level decisions. Its signed normal response selects one low-saliency read side for the whole loop; its magnitude sets the within-loop sampling density; and its strongest smoothed response selects the common anchor p_0. This unification avoids separate heuristics for side selection, density, and sequence endpoints.

Local reading and global aggregation are strictly separated. Every retained contour is encoded independently. A Normal Mamba reads each ray from its collision back to a tagged contour point. Two Contour Mamba scans start at the same p_0, traverse the loop in opposite directions, and end on a tagged return to p_0. A terminal contour reads both normal sides before a second loop-level fusion. Only after all local encodings are ready does a Tree Mamba follow the sparse augmented contour tree. Figure 1 shows one complete route generated by these rules.

The contributions are:

- an input-specific serialization that combines closed level-set routes with a low-to-high augmented-contour-tree schedule;
- one Sobel-guided rule for loop-wide side selection, fixed-budget density redistribution, and the shared contour anchor; and
- raw-pixel normal and contour encoders with tagged endpoints, followed by Tree Mamba aggregation with cheap splits, order-invariant joins, tagged terminal leaves, and an explicit empty-route fallback.

This manuscript presents a complete architectural and algorithmic specification. Its contribution is the deterministic route construction, token interfaces, topological aggregation rule, and structural analysis; it makes no empirical performance claim.

2 Related Work

2.1 Serialization in Visual State-Space Models

Selective state-space models operate on ordered sequences, so visual variants must choose both an adjacency pattern and a readout direction. Vim flattens image patches and uses bidirectional state-space blocks with positional information [17]. VMamba's SS2D traverses the feature grid along four routes, combining multiple directional contexts while retaining dense grid coverage [13]. PlainMamba improves continuity through a designed continuous 2D traversal and direction-aware updating [16]. LocalMamba partitions the grid into windows and searches for suitable scan choices at different layers, prioritizing short-range two-dimensional dependencies [10]. These methods make increasingly careful use of spatial adjacency, but their basic geometric primitives—rows, columns, continuous grid paths, or windows—are defined independently of the level-set structure of the current image.

Several later models make the route itself content dependent. DAMamba learns scanning orders and regions through Dynamic Adaptive Scan [12]. QuadMamba predicts token locality and recursively partitions windows into quadrants; a Gumbel-Softmax masking mechanism supplies a straight-through training path for the discrete partition [15]. For video super-resolution, MambaVSR builds semantic connectivity graphs, clusters related content, and interleaves temporal features along the resulting spatial order [7]. These systems are close to TopoContour-Mamba in rejecting a single fixed raster order. The distinction is the structured object being learned or extracted: they adapt permutations, regions, quadrants, or semantic groups, whereas TopoContour-Mamba derives closed level-set components and explicitly preserves their split/join evolution in a contour tree. Our initial route generator is deterministic rather than learned, so this difference is architectural and not a claim of greater adaptivity.

PRISMamba is the closest geometric comparison. It partitions the image into concentric rings, performs order-agnostic aggregation within a ring, and propagates context between rings with short radial SSMs [9]. TopoContour-Mamba does not assume a common center or nested circular geometry. It may contain many disconnected and nonconvex contours at one level; each is scanned in both arc-length directions, while normal rays read adjacent bands and a separate contour tree controls global low-to-high aggregation. Thus PRISMamba organizes a dense grid by image-centered rings, whereas TopoContour-Mamba organizes sparse, input-specific level sets by scalar topology.

Table 1: Representative visual serialization rules. The comparison concerns route structure, not Mamba's input-dependent gates.

Method	Route	Key difference from ours
Vim / VMamba	patch/grid directions	dense coordinates; no extracted level-set topology
Plain-/ LocalMamba	continuous paths or windows	improved continuity/locality within predefined grid primitives
DAMamba / QuadMamba	learned orders, regions, or quadrants	trainable spatial organization without an explicit contour tree
MambaVSR	semantic groups and graph order	content-aware restoration route rather than scalar level sets
PRISMamba radial SSMs	concentric rings, common-center geometry without split/join evolution	
Ours	normals, closed contours, and tree	separates band reading, loop encoding, and topological aggregation

2.2 Saliency, Contours, and Topological Structure

Classical visual-attention models combine intensity, color, and orientation cues, while frequency-tuned saliency measures contrast in Lab space [11, 1]. Such maps represent bottom-up conspicuity, not guaranteed task relevance. We use the frequency-tuned map only as a reproducible scalar field for the initial serialization experiment. A task-conditioned CNN field is discussed in Section 8; its training boundary must be stated carefully because hard contour extraction is discrete.

Contour trees encode how connected components of level sets appear, continue, split, join, and disappear as a scalar threshold changes [4]. Augmentation associates regular contours or mesh elements with tree arcs, and on-demand variants avoid materializing unnecessary geometry [14, 6]. Existing neural uses include contour-tree hierarchies and topology-guided maps for segmentation [8, 2]. In contrast, TopoContour-Mamba does not predict a contour tree or use it only as a loss. The tree is a deterministic execution graph: regular contours define independent local Mamba sequences, and only their tagged outputs travel through the tree.

Graph Mamba addresses the more general problem of ordering and encoding arbitrary graph neighborhoods [3]. Our graph is narrower and more constrained. Scalar value gives every tree edge a low-to-high orientation; split and join semantics are known; and unsampled paths can be contracted without inventing nearest-neighbor links. This structure permits specialized operations—copy at a split, permutation-invariant fusion at a join, and Mamba along nonbranching segments—instead of applying one generic graph tokenization rule.

Positioning of this work. The novelty is therefore not "using a different curve" in isolation. It is the separation of three relations that previous scan designs usually conflate: first-hit normals determine local cross-level evidence, closed contours determine within-component order, and an augmented contour tree determines global information flow. Table 1 records these distinctions; the contribution is methodological, not a claim of empirical superiority.

3 Route Construction

3.1 Saliency Field and Sparse Augmented Tree

Let $I \in [0,1]^{H\times W\times 3}$ be the normalized RGB image. A deterministic operator produces a saliency field $S \in [0,1]^{H_r\times W_r}$. The reference pipeline computes frequency-tuned Lab contrast at input resolution [1], clips it to the 1st–99th percentile range, affinely normalizes it, applies a Gaussian with $\sigma = 3$ input pixels, and area-resizes it to the route grid. A zero clipped range produces the all-zero field. The model reads RGB from I; S is used only to build and describe the route.

A one-cell frame with a value below the range of S closes contours that touch the image boundary. Each route-grid cell is split by the same northwest-to-southeast diagonal, and equal scalar values use a deterministic symbolic offset. We compute the full contour tree T from this piecewise-linear field before selecting model routes. The padded rectangle is simply connected, so T has no cycle. It may contain split and join saddles, but two siblings created by one split cannot later rejoin; a join combines branches with independent lower origins. A domain with holes would require a Reeb graph.

At L retained regular levels, we extract closed contours, remove loops below simple perimeter and area thresholds, and insert the remaining loops on their tree arcs. Paths with no retained loop are contracted into a sparse execution tree T_A. The tree therefore fixes cross-level topology without forcing every fine branch to consume tokens. A normal-ray collision never creates or redirects a tree edge.

After smoothing S, we compute one 3×3 Sobel field $G = (G_x, G_y)$ on the route grid. All contours reuse this field.

3.2 One Sobel Rule for Side, Density, and Anchor

Let a retained contour C have perimeter P_C on the route grid. Its number of unique model points is fixed before adaptive placement:

$$N_C = \max(N_{\min}, \mathrm{round}(P_C/\delta_0)) . \tag{1}$$

The default is $N_{\min} = 16$ and $\delta_0 = 2$. There is no semantic upper cap: long loops use larger length buckets or exact chunked scans. Adaptive placement moves a fixed number of points; it never creates extra tokens for one region.

We orient C clockwise and place $M_C = 2N_C$ pilot points uniformly by arc length. Let u_i be a pilot and n_i its consistently oriented unit normal. Bilinear sampling of the Sobel field gives one signed normal response

$$r_i = G(u_i)^\top n_i. \tag{2}$$

Its sign says which normal side increases saliency. If the mean of r_i is nonnegative, $+n_i$ is the high side and every ordinary ray uses $-n_i$; otherwise every ordinary ray uses $+n_i$. An exact zero uses the negative-normal side. This is one decision for the whole contour, not one decision per point.

The magnitude $g_i = |r_i|$ measures how quickly the saliency field changes across the contour. We smooth g_i once around the loop with the circular kernel $(1,2,3,2,1)/9$, obtaining $\bar{g}_i$. Let m_C be the median of the positive $\bar{g}_i$. If no positive response exists, all density weights are one. Otherwise,

$$w_i = \mathrm{clip}\left(\frac{\bar{g}_i}{m_C + \epsilon}, 1, 3\right). \tag{3}$$

Thus weak regions keep the base density, strong changes receive more points, and the cap prevents a small region from consuming the loop budget. With uniformly spaced scalar levels, a larger normal derivative is also a first-order proxy for a smaller gap between neighboring level sets.

The anchor p_0 is the pilot with the largest smoothed, unclipped response $\bar{g}_i$. Ties use the lexicographically smallest image coordinate (y, x). The same Sobel measurements therefore determine the read side, density, and start/end point without any pilot-stage collision query.

For a pilot interval of physical arc length Δs_i, assign mass

$$a_i = \Delta s_i \frac{w_i + w_{i+1}}{2}. \tag{4}$$

Starting at p_0, we place exactly N_C final points at equal increments of cumulative mass. The anchor is kept as the first point. High-weight intervals receive closer samples, but the count never changes. No sequence is stretched, truncated, or padded with invented geometry. Clockwise and counterclockwise scans use the same physical points in reverse order.

Only after redistribution do we cast geometric rays. A final point sends its ordinary ray on the selected loop-wide side until the first positive collision with a retained contour polyline or the padded frame. The starting contact and the two incident segments of C are ignored. A terminal contour also casts the opposite-side ray. Route coordinates are mapped back to the original image, where RGB and saliency values are bilinearly sampled. Pilot points are never model tokens.

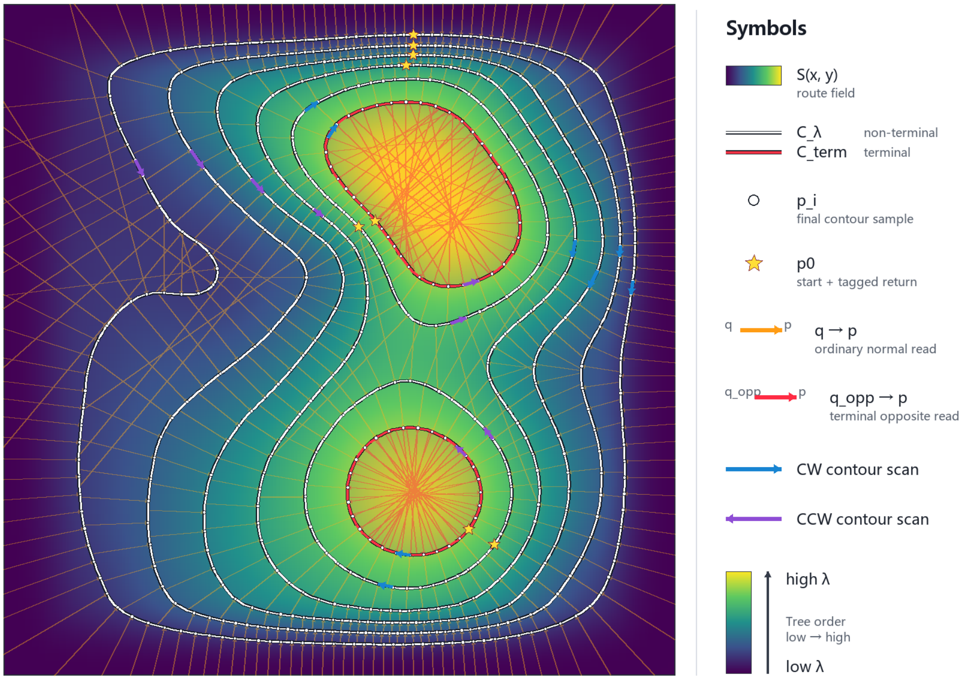

Figure 1: One route generated by the specified rules. White curves are retained nonterminal contours; red curves and tinted interiors mark the last retained contour of each visible branch. Orange arrows show ordinary Normal Mamba reads from the first collision q back to the contour point p; red arrows add the opposite-side reads used only at terminal contours. Stars mark the shared start and tagged return p_0, while blue and purple arrows show the two loop directions.

The route does not declare low-saliency regions useless or remove them. It lets broad context enter the history early, with sparse placement where the Sobel field changes weakly, while sharper and terminal evidence is read later and remains closer to tagged outputs. Selective forgetting is therefore used as an attention schedule: weak context may fade but is still available, whereas distinctive evidence is less exposed to subsequent updates. This is an inductive bias rather than a memory guarantee. Because the geometry is fixed before scanning and every contour is locally independent, the image-specific order still permits parallel execution across rays and contours.

4 Architecture

4.1 Raw-Pixel Normal and Contour Encoding

All retained contours are encoded independently. No parent output enters a normal ray or a contour loop. This is the boundary between local visual reading and global tree aggregation.

Consider a final contour point p at retained level λ_C. Route construction has already fixed the ordinary side σ_C. Casting from p on that side finds the first collision q_p. Geometry is found from p to q_p, but Normal Mamba reads the samples in reverse, from q_p back to p.

For a ray sample r, let $c(r)$ be bilinearly sampled normalized RGB. Its input also contains normalized image coordinates, the pair $(\lambda_C, S(r) - \lambda_C)$, its progress from collision to contour, and the physical ray length. A shared pointwise projection E_N maps these values to the model width. This projection is not a CNN and does not mix neighboring image locations. The last sample p receives a normal-end tag:

$$n_p = \mathcal{M}_{\mathrm{N}}\big(E_N[c, S - \lambda_C, \lambda_C, \hat{x}, \hat{y}, t, d]_{q_p \to p} + e_{\mathrm{N,end}}@p\big)_p . \tag{5}$$

The tagged output n_p summarizes the RGB sequence along that ray.

The contour-point token keeps a direct path for the endpoint RGB instead of asking n_p to preserve it perfectly. A shared projection E_C combines n_p, $c(p)$, normalized coordinates, λ_C, the local arc-length support, and the ray direction and length into x_p. Coordinates are needed because sequence order alone does not reveal two-dimensional position, and local arc length is needed because adaptive neighbors are not equally spaced.

Let the final points be $(p_0, \ldots, p_{N_C - 1})$, where p_0 is the Sobel anchor retained during resampling. A closed curve has no natural first token, so we cut only its serialized order at p_0. Both loop directions begin with an untagged copy of p_0. The clockwise scan visits p_1 through p_{N_C-1}; the counterclockwise scan visits the same points in reverse. Each returns to a second copy of p_0, and only that return carries the contour-end tag:

$$X_C^{\circlearrowright} = (x_{p_0}, x_{p_1}, \ldots, x_{p_{N_C-1}}, x_{p_0} + e_{\mathrm{C,end}}), \tag{6}$$
$$X_C^{\circlearrowleft} = (x_{p_0}, x_{p_{N_C-1}}, \ldots, x_{p_1}, x_{p_0} + e_{\mathrm{C,end}}). \tag{7}$$

Every nonanchor point appears once in each direction. The anchor has a distinct opening position and tagged closing position. The two outputs at the tagged returns are concatenated, projected, and normalized into one local contour token y_C. The tag fixes the readout position; it does not guarantee lossless memory. We also retain the aligned per-point outputs from both directions for terminal contours.

Both-side terminal encoding. If C has no active higher successor, it is the last retained contour on that tree branch. Its ordinary-side ray and first pair of contour loops have already been computed. Each point now casts a second ray on the side opposite σ_C. This ray also starts at its first collision and ends at the same tagged contour point p.

At each p, we fuse the opposite-side Normal Mamba output with the aligned clockwise and counterclockwise outputs from the first contour pass. Those first-pass outputs already contain the ordinary-side ray information. The fused points are scanned by a second pair of Contour Mamba loops using the same p_0, the same sample set, opposite traversal orders, and tagged returns. Their final fusion gives y_C^{term}. The terminal result therefore contains both normal sides, not one side replacing the other. The same rule covers a peak point, a broad peak ring, and a basin boundary; no interior-fill case is needed.

Normal rays, ordinary loops, and terminal loops are batched by true length. Padding is placed after the tagged endpoint and masked. Readout always indexes the true endpoint.

4.2 Tree Mamba Aggregation

Only after every local token is available do we process T_A from low to high saliency. A maximal nonbranching tree segment B contains retained contours $(C_1, \ldots, C_r)$ in scalar order. A shared Tree Mamba $\mathcal{M}_{\mathrm{G}}$ receives an incoming summary followed by the local contour tokens:

$$b_B = \mathcal{M}_{\mathrm{G}}\big(y_B^{\mathrm{in}}, y_{C_1}, \ldots, y_{C_r} + e_{\mathrm{seg}}\big)_r. \tag{8}$$

For a terminal contour, y_C^{term} replaces its first-pass y_C. At a source, y_B^{in} is a learned source token. The last real contour carries the segment-end tag; a topological leaf also carries a leaf tag. Only tagged output vectors y cross segment boundaries. Hidden SSM states and convolution caches never cross a junction.

At a split saddle, b_B is copied to each outgoing segment. In a contour tree, the resulting siblings cannot later rejoin each other, so no recurrent reconciliation rule is needed.

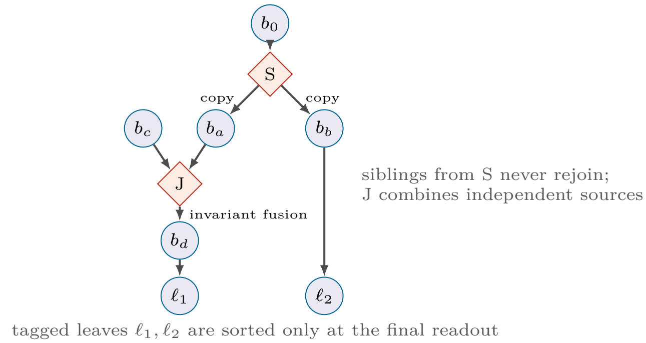

Figure 2: Tree-level aggregation. A split copies one tagged summary. A join fuses independently born branches without imposing an arbitrary order; siblings of the same earlier split cannot rejoin in a tree.

At a join saddle, incoming branches have independent lower origins and no natural order. Let b_i summarize branch i, and let Q_i be the number of unique final contour points represented by that branch history. We use a support-aware permutation-invariant gate:

$$s_i = \tfrac{1}{2}\log(1 + Q_i) + \mathcal{G}(\mathrm{LN}(b_i)), \tag{9}$$
$$\alpha_i = \frac{\exp(s_i)}{\sum_j \exp(s_j)}, \qquad b_{\mathrm{join}} = \mathrm{LN}\left(W_J \sum_i \alpha_i b_i + e_{\mathrm{join}}\right). \tag{10}$$

The final layer of $\mathcal{G}$ is initialized to zero. Fusion starts from a square-root-like support prior and then learns content corrections without using the unstable norm of y. The same operator handles any join degree.

4.3 Tagged Terminal Readout

Every terminal leaf produces a tagged summary ℓ_k. For each point on its terminal contour, we average saliency over both normal rays actually used by the terminal encoder. We then combine point scores with their physical arc-length supports. This gives an ordering score ρ_k without letting a dense sampling region vote more often merely because it contains more points.

Terminal summaries are sorted by increasing ρ_k. Ties use the terminal extremum value and then centroid coordinates. Every token receives e_{leaf}; the last also receives e_{final}:

$$g = \mathcal{M}_{\mathrm{G}}\big(\ell_{\pi(1)} + e_{\mathrm{leaf}}, \ldots, \ell_{\pi(K)} + e_{\mathrm{leaf}} + e_{\mathrm{final}}\big)_K. \tag{11}$$

Even one leaf uses both tags. If no loop survives, we project the per-channel mean and variance of the raw RGB image:

$$g = \mathrm{LN}(W_0[\mathrm{Mean}(I), \mathrm{Var}(I)] + e_{\mathrm{empty}}). \tag{12}$$

A linear head predicts the class from g.

An internal join and the final leaf order serve different purposes. A join is an unordered topological event and

Algorithm 1: Forward pass of TopoContour-Mamba

1. Normalize the RGB image, compute the deterministic saliency field, smooth it, and compute one Sobel field.
2. Build the full contour tree on the padded piecewise-linear saliency field. Extract retained contours, filter tiny loops, insert valid loops on tree arcs, and contract paths with no sampled contour.
3. For each contour, orient it clockwise and set its point budget from perimeter. Uniform pilots sample the signed normal Sobel response: its mean selects one low-saliency ordinary side, its smoothed magnitude sets density, and its maximum selects p_0.
4. Starting at p_0, redistribute exactly the fixed point budget by weighted arc length. At final points only, find the first collision on the chosen ordinary side.
5. Sample raw RGB and route metadata along every ordinary ray. Scan from collision to the tagged contour point, build contour-point tokens, and scan both loop directions from p_0 to a tagged return, obtaining y_C.
6. For every terminal contour, also scan the opposite-side rays. Fuse them pointwise with the first-pass contour outputs, then repeat the two p_0-to-tagged-return loops to obtain y_C^{term}.
7. Process the sparse tree from low to high. Tree Mamba scans nonbranching segments; splits copy tagged y, and joins use support-aware invariant fusion.
8. Score terminal leaves from saliency on both normal sides actually read, with arc-length support weights. Sort leaves from low to high and scan them with the final Merge Mamba.
9. If no contour survives, project raw RGB channel means and variances. Return the classifier output.

uses invariant fusion. Terminal leaves are serialized once by measured read saliency so that weaker terminal evidence enters before stronger evidence at the global readout.

5 Forward Pass and Parallel Execution

Route planning is deterministic in the initial model and finishes before any Mamba scan. The complete forward pass is:

Variable-length rays, loops, and tree segments are grouped into dynamic length buckets. Padding follows the true tagged endpoint and is masked. All normal rays and local contour encoders are independent and can be batched across levels. Tree segments at the same ready frontier can also run in parallel. The only sequential dependencies are within one ray or loop, along low-to-high tree edges, and in the final leaf sequence. Long contours use larger buckets or exact chunked scans rather than truncation or geometric stretching.

Only tagged y vectors cross tree-segment boundaries. Splits copy a vector, not parameters, hidden states, or recurrent caches. The raw-pixel projections and all Mamba modules are trainable in the initial model; the traditional saliency operator, Sobel decisions, contour extraction, and hard tree remain detached route construction. A future saliency CNN would therefore require a separate soft training path.

6 Structural Analysis

Image-dependent pseudo-time. Fixed raster or cross scans serialize pixels by coordinates. TopoContour-Mamba lets a saliency field determine closed local routes and their low-to-high tree order. Weak regions enter the global history earlier and salient regions later, placing stronger evidence closer to the final tagged readout. Selective state updates may retain broad early context while emphasizing later evidence, but this is an inductive bias, not a guarantee about Mamba's memory.

Three separate relations. Normal rays read the band next to a contour, bidirectional loops integrate one connected component, and the augmented contour tree aggregates across scalar levels. A ray collision never defines ancestry, and a parent summary never enters a local contour encoder. The tree may contain splits and joins, but its acyclicity means that siblings created by one split cannot later rejoin; a join combines branches with independent lower origins.

One Sobel signal, three decisions. For each uniform pilot, the signed normal Sobel response tells which side increases saliency. Averaging its sign gives one ordinary side for the whole loop. Its smoothed magnitude controls density, and the strongest position fixes p_0. These roles use the same local measurement but different parts of it. The rule is simple and avoids pilot-stage collision searches. It also couples two biases: a strong transition receives more samples and lies near the tagged loop endpoint. This intended emphasis must be tested against uniform sampling and independent anchors.

Fixed-budget raw-pixel sampling. Perimeter fixes how many contour points are available; Sobel weights only move them. Weighted cumulative arc length concentrates a fixed budget near strong cross-contour changes without allowing one region to create extra tokens. RGB is then sampled from the original image, so the model does not inherit locality from a CNN stem. Coordinates, level, ray progress, ray length, and local arc support tell Mamba what the irregular raw-pixel sequence means geometrically. Arc-length weighting also prevents dense regions from dominating terminal saliency scores merely because they contain more points.

Cost and parallelism. Selective-scan work is linear in the number of routed samples: ordinary and terminal ray samples, two directions for each contour pass, tree-segment tokens, and final leaf tokens. It is not necessarily linear in image pixels. Route construction adds saliency, Sobel, contour-tree, and final collision-query costs; for V route-grid vertices, standard contour-tree construction is typically $O(V \log V)$ before output-sensitive refinements [4]. Many short irregular sequences, padding, and tree synchronization may reduce GPU utilization despite linear token complexity.

A fair comparison must match token width and training recipe, include raw-pixel or token-matched baselines, and report deterministic route construction in end-to-end latency. Routed-token count and asymptotic complexity alone do not predict wall-clock speed.

7 Discussion and Limitations

The initial frequency-tuned field measures bottom-up contrast, not task relevance. Low-to-high therefore means saliency order rather than known semantic importance. Section 8 outlines a trainable CNN replacement for the route field, while keeping a clear gradient boundary around hard topology.

The geometric defaults are intentionally simple: one heatmap smoothing step, one circular Sobel-strength smoothing step, a small-loop filter, and no persistence simplification. Fine branches can remain in the full topology without receiving scan tokens. This saves computation but may discard useful detail. The padded rectangle makes a contour tree valid; masked domains with holes require a Reeb graph, and plateaus require deterministic symbolic tie handling.

Direct RGB sampling removes a CNN feature stem, but it also removes built-in local texture aggregation. The model sees only pixels reached by its rays and contours. Sparse routes can miss evidence between them, and a capped ray sample count can reduce spatial resolution on long rays. Patch sampling, a shallow feature adapter, and extra local scans are therefore useful baselines or extensions rather than hidden parts of the initial model.

Tagged y is a compact interface, not proof of complete memory. Compressing a long branch into one vector may lose evidence, and copying it at a split does not create different prior histories. Terminal contours read both sides, but their rays can overlap in bowls or nonconvex regions. The shared Sobel rule also gives strong transitions both denser samples and a position near the tagged contour endpoint. Ablations should separate these effects.

Finally, irregular short sequences can be less efficient than a regular raster despite Mamba's linear scan. Saliency computation, contour construction, collision queries, empty-route frequency, padding overhead, and actual latency are necessary for any empirical comparison. The contribution here is the complete route construction and structural analysis, not measured gains in accuracy, robustness, memory, or throughput. The human-reading analogy is a motivation, not a biological claim.

8 Extensions

The following variants extend the initial architecture:

- **Trainable structural map:** an end-to-end or separately trained convolutional module outputs an input-sized structural-intensity map.
- **Dominant-direction ring scan:** replace each normal ray with concentric rings around its contour sample. A local dominant direction fixes their common start, return, and aggregation point. Bidirectional Mamba scans each ring, then aggregates the ring states from outside to inside at the center.
- **Inter-level dense scan:** horizontal and vertical scans between adjacent contours cover regions missed by sparse normals, and their states are aggregated into nearby contour samples. Each contour is also sampled pixel by pixel.
- **Dense level sampling:** use gap-free successive contour levels, remove normal scans, and scan only densely placed contour samples.

9 Conclusion

TopoContour-Mamba turns a deterministic saliency field into image-specific visual pseudo-time while reading content directly from the RGB image. One Sobel field selects a loop-wide low-saliency side, redistributes a perimeter-fixed point budget, and fixes the shared contour anchor. Reverse normal scans and bidirectional contour loops encode each contour independently; terminal contours combine both normal sides through a second loop pass. A sparse augmented contour tree then aggregates tagged contour outputs from low to high, using copied split summaries and permutation-invariant joins. Final leaves are ordered once by saliency on their actual reads. The method differs from adaptive permutations or spatial partitions by preserving level-set split/join evolution and by separating local reading from global topology. This manuscript fully defines the mechanism without claiming empirical gains. Learned saliency, richer raw-image sampling, branch summaries, and tree-parallel execution are optional extensions.

References

[1] Radhakrishna Achanta, Sheila S. Hemami, Francisco Estrada, and Sabine Süsstrunk. Frequency-tuned salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1597–1604, 2009.

[2] Diego Adame, Jose A. Nunez, Fabian Vazquez, Nayeli Gurrola, Huimin Li, Haoteng Tang, Bin Fu, and Pengfei Gu. Topo-vm-unetv2: Encoding topology into vision mamba unet for polyp segmentation. *arXiv preprint arXiv:2505.06210*, 2025.

[3] Ali Behrouz and Farnoosh Hashemi. Graph mamba: Towards learning on graphs with state space models. *arXiv preprint arXiv:2402.08678*, 2024.

[4] Hamish Carr, Jack Snoeyink, and Ulrike Axen. Computing contour trees in all dimensions. *Computational Geometry*, 24(2):75–94, 2003.

[5] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling*, 2024.

[6] Charles Gueunet, Pierre Fortin, Julien Jomier, and Julien Tierny. Task-based augmented contour trees with fibonacci heaps. *IEEE Transactions on Parallel and Distributed Systems*, 30(8):1889–1905, 2019.

[7] Linfeng He, Meiqin Liu, Qi Tang, Chao Yao, and Yao Zhao. Mambavsr: Content-aware scanning state space model for video super-resolution. *arXiv preprint arXiv:2506.11768*, 2025.

[8] Wenchong He, Arpan Man Sainju, Zhe Jiang, and Da Yan. Deep neural network for 3d surface segmentation based on contour tree hierarchy. *arXiv preprint arXiv:2008.11269*, 2020.

[9] Yi-Kuan Hsieh, Jun-Wei Hsieh, Xin Li, Ming-Ching Chang, and Yu-Chee Tseng. Partial ring scan: Revisiting scan order in vision state space models. *arXiv preprint arXiv:2602.04170*, 2026.

[10] Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. Localmamba: Visual state space model with windowed selective scan. In *European Conference on Computer Vision Workshops*, 2024.

[11] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[12] Tanzhe Li, Caoshuo Li, Jiayi Lyu, Hongjuan Pei, Baochang Zhang, Taisong Jin, and Rongrong Ji. Damamba: Vision state space model with dynamic adaptive scan. In *Advances in Neural Information Processing Systems*, 2025.

[13] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. In *Advances in Neural Information Processing Systems*, 2024.

[14] Mohit Sharma and Vijay Natarajan. On-demand augmentation of contour trees. In *11th Indian Conference on Computer Vision, Graphics and Image Processing*, pages 104:1–104:8, 2018.

[15] Fei Xie, Weijia Zhang, Zhongdao Wang, and Chao Ma. Quadmamba: Learning quadtree-based selective scan for visual state space model. In *Advances in Neural Information Processing Systems*, 2024.

[16] Chenhongyi Yang, Zehui Chen, Miguel Espinosa, Linus Ericsson, Zhenyu Wang, Jiaming Liu, and Elliot J. Crowley. Plainmamba: Improving non-hierarchical mamba in visual recognition. In *British Machine Vision Conference*, 2024.

[17] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

A Reference Configuration

Table 2 fixes one reproducible starting point, not tuned optima.

Table 2: Route defaults for a 224×224 image and 56×56 route grid.

Item	Default
Saliency	input-scale frequency-tuned Lab; 1–99 percentile clip, then normalize
Smoothing / resize	Gaussian $\sigma = 3$ input pixels; area-resize to 56×56
Sobel	3×3 on smoothed S; bilinear pilot lookup
Boundary / cells	below-range frame / fixed NW–SE split
Retained levels	$\lambda_j = j/13$, $j = 1, \ldots, 12$; deterministic critical-value nudge
Loop filter	route-grid perimeter ≥ 12, area ≥ 8 cells; no persistence filter
Point budget	$N_C = \max(16, \mathrm{round}(P_C/2))$; no upper cap
Pilot count	$2N_C$ uniform points
Ordinary side	lower side from mean signed normal Sobel response
Strength smoothing	one circular $(1, 2, 3, 2, 1)/9$ convolution
Density weight	$\mathrm{clip}(\bar{g}/(\mathrm{median}^{+}\, \bar{g} + 10^{-6}), 1, 3)$
Anchor p_0	maximum smoothed unclipped $\bar{g}$; (y, x) tie break
Collision set	retained polylines plus padded frame; final points only
Normal samples	$R_p = \mathrm{clip}(\lceil d_p/2 \rceil + 1, 2, 32)$; include q_p, p
Terminal score	arc-weighted saliency mean over both actual ray sides

Here d_p is the collision-to-contour length in input pixels; perimeter and area thresholds use route-grid units. Dynamic power-of-two buckets continue beyond 256, and exact chunking never drops samples.

Table 3: Initial model defaults.

Item	Default
Token width / Mamba core	192; Mamba-1 with state 16, expansion 2, convolution 4
Input embedding	type-specific pointwise Linear–LayerNorm, shared over positions; no CNN stem
Normal input	RGB, level/difference, (x, y), progress, ray length
Contour input	normal summary, RGB, (x, y), level, arc support, ray geometry
Normal / Contour / Tree blocks	1 / 2 / 1
Direction parameters	shared; concatenate, project, LayerNorm
Cross-segment state	tagged y only; never h or cache
Join prior	$\frac{1}{2}\log(1 + Q_i)$; zero-initialized learned gate
Tags	learned normal, contour, segment, join, leaf, final, empty vectors
Length buckets	dynamic powers of two; optional exact chunking
Empty route	projected RGB channel mean and variance
Stochastic depth	0 in the reference model

Tags are added at true endpoints before padding; padding follows the endpoint and is masked. Block counts are per sequence pass.

Evaluation protocol. Recompute saliency and routes after every spatial augmentation. Fixed-path baselines should match input projection, token width, Mamba blocks, optimizer schedule, and full-grid or routed-token count. Report at least three seeds. Ablate route/leaf order, sampling, terminal reads, local input, tree joins, tags, and hard/soft routing; report accuracy, tokens, route construction and model latency, padding, ray overlap, and empty-route rate.