# CTKAN: Making Kolmogorov–Arnold Networks Think over Spline Activities for Medical Image Segmentation

**Anonymous authors**
Paper under double-blind review

## Abstract

Neural computation can benefit from evolving internal dynamics rather than relying solely on a single static mapping. Although Kolmogorov–Arnold Networks (KANs) have shown promise in medical image segmentation through learnable spline functions, most KAN-based models evaluate spline responses once and aggregate them immediately, preventing basis-wise contributions from being reconsidered before the layer output is formed. We argue that KANs should think over their spline activities by repeatedly refining basis contributions before aggregation, thereby strengthening nonlinear representation. We therefore propose CTKAN, a Continuous-Thought Kolmogorov–Arnold Network that introduces neural-dynamics-inspired recurrent evolution into the spline activity space. CTKAN maintains a sample-conditioned state for each output–input–basis triplet and updates these states over multiple thought ticks. Normalized pairwise coactivities provide relational feedback, while terminal states generate residual, sample-adaptive gates over shared spline coefficients, allowing each sample to induce a distinct mixture of spline bases. We integrate CTKAN at two deployment scales into a U-shaped segmentation backbone. Experiments on ultrasound, endoscopic, histopathological, and cardiac MRI benchmarks show that both variants outperform U-KAN, with CTKANMax improving the four-dataset macro-average by 1.29 IoU and 1.03 Dice points. These results show that making KANs think over their own spline activities improves segmentation performance through sample-adaptive spline modulation.

## 1 Introduction

Medical image segmentation has progressed from convolutional encoder–decoders such as U-Net (Ronneberger et al., 2015) to nested decoders, attention, Transformers, and state-space models (Zhou et al., 2018; Oktay et al., 2018; Chen et al., 2021; Ma et al., 2024). These architectures primarily improve how spatial context is encoded and fused. Their local nonlinear mappings, however, are still commonly implemented by convolutions or multilayer perceptrons with fixed-form activations.

Kolmogorov–Arnold Networks (KANs) offer a different primitive: learnable univariate functions, usually expanded with B-spline bases, are placed on network edges (Liu et al., 2024). U-KAN demonstrated that these edge functions can serve as nonlinear visual operators in a tokenized U-shaped architecture (Li et al., 2025). Subsequent medical variants combine KANs with global modeling, semi-supervision, dual encoders, attention, or uncertainty estimation (Wu et al., 2024; Ye et al., 2025; Alshami et al., 2025; Chen et al., 2025; Zhang et al., 2026b). Yet the spline computation inside these models remains largely one-shot: basis responses are evaluated, multiplied by shared coefficients, and summed immediately.

This aggregation creates a structural bottleneck. A KAN spline branch exposes only the sum over input dimensions and bases; distinct basis-wise activity patterns may therefore yield the same output token. Recurrence or gating placed after a KAN layer can refine the resulting token features or predictions, but it cannot selectively reconsider the hidden output–input–basis contributions unless that decomposition is reconstructed. We instead place dynamics before aggregation, where each evolving state remains aligned with the coefficient it controls.



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.

This design is motivated by neural-dynamics models, in which computation emerges from the evolution and coordination of internal states (Elman, 1990; Maass et al., 2002; Darlow et al., 2025). It is also distinct from iterative segmentation, which recurrently updates feature maps or masks (Wang et al., 2019; Zhang et al., 2019; Tang et al., 2021), and from temporal or adaptive KANs that model observed sequences or predict content-dependent parameters without repeatedly evolving the basis decomposition of one static sample (Genet & Inzirillo, 2024; Lu et al., 2025; Zhang et al., 2026a).

We propose **CTKAN**, a Continuous-Thought Kolmogorov–Arnold Network. CTKAN summarizes the token responses of every spline basis, constructs a sample-conditioned activity state for each output–input–basis triplet, and evolves these states over latent thought ticks. Fixed sampled pairs read signed co-activities from the current memory; a factorized projection writes this relational context back to the activity tensor. The terminal memory then generates a residual gate for every spline coefficient before aggregation. This recurrent, history-conditioned refinement is the operational meaning of *thought* in CTKAN.

Our contributions are threefold. First, we identify one-shot spline aggregation as an operator-level limitation and introduce stateful refinement inside the spline activity space. Second, we develop sampled co-activity feedback, exponential memory, and residual coefficient modulation that preserve the original KAN bases and base branch. Third, we evaluate two deployment scales on four medical segmentation benchmarks, report operator-aware arithmetic and hardware cost, and provide controlled ablations and visual diagnostics of the learned spline modulation.

## 2 Related Work

**Visual and adaptive KANs.** KAN replaces fixed node-wise activations with learnable edge functions (Liu et al., 2024). U-KAN introduced tokenized KAN blocks into a U-shaped segmentation and generation framework (Li et al., 2025); TransUKAN combined KAN transformations with global modeling (Wu et al., 2024); and later medical variants emphasized semi-supervision, anatomy-aware encoders, attention, and uncertainty (Ye et al., 2025; Alshami et al., 2025; Chen et al., 2025; Zhang et al., 2026b). More closely related adaptive designs include TKAN for temporal forecasting and GDKansformer for pathology-aware fitting (Genet & Inzirillo, 2024; Lu et al., 2025). PAKAN generates content-conditioned spline-summation weights, whereas LG-KAN applies input-conditioned routing over local spline bases (Zhang et al., 2026a; Cheng & Liu, 2026). CTKAN instead maintains recurrent activity memory aligned with each output–input–basis triplet and repeatedly revises it through cross-basis co-activity before aggregation.

**Neural dynamics and iterative segmentation.** Neural-dynamics models regard the temporal evolution of internal states as an essential part of computation rather than an incidental intermediate process (Elman, 1990; Maass et al., 2002). The Continuous Thought Machine further develops this perspective through neuron-level temporal processing and synchronization (Darlow et al., 2025). A related but task-level approach appears in iterative segmentation: recurrent U-Net, CANet, and recurrent mask refinement repeatedly update feature maps, contextual representations, or predicted masks (Wang et al., 2019; Zhang et al., 2019; Tang et al., 2021). Although these methods share the principle of history-conditioned refinement, their recurrent states operate on aggregated features or task outputs. CTKAN instead introduces internal evolution before KAN aggregation, where output–input–basis spline activities remain individually identifiable and can be repeatedly refined before forming the layer output.

## 3 CTKAN

### 3.1 Standard KAN mapping and spline-activity construction

We first introduce the standard KAN mapping inherited by CTKAN and identify the spline decomposition on which the proposed recurrent refinement operates. Let $X \in \mathbb{R}^{B\times N\times I}$ contain $N$ tokens with $I$ input channels. In a standard KAN layer, each edge combines a base activation with $K$ learnable B-spline basis functions, producing

$$y_{bno}^{\mathrm{KAN}} = \sum_i W_{oi}^{\mathrm{base}}\sigma(x_{bni}) + \sum_{i,k} W_{oik}B_k(x_{bni}), \tag{1}$$

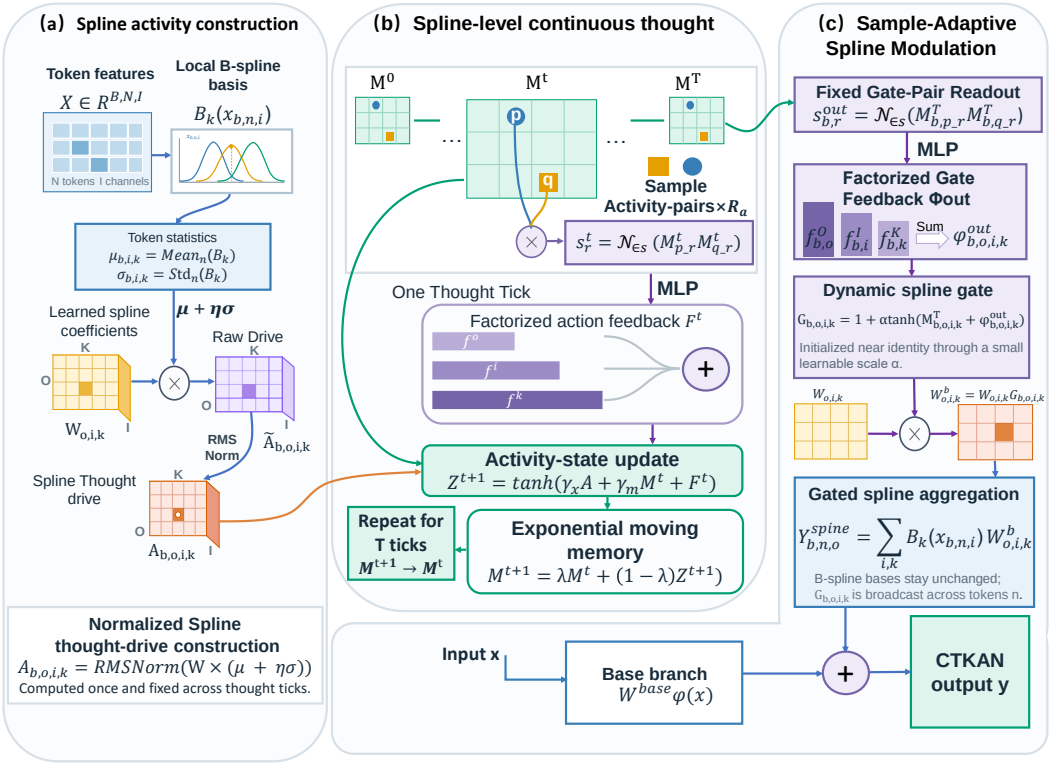



Figure 1: CTKAN overview. Token statistics and shared coefficients form a normalized spline drive (left). Sampled co-activities are repeatedly read from memory and projected back as factorized feedback (middle). The terminal memory produces residual, sample-adaptive coefficient gates before spline aggregation (right).

where $W_{oik}$ includes the learnable spline coefficient and edge-wise scaler. Equation 1 describes the original KAN computation rather than CTKAN. CTKAN retains this mapping structure, including the base branch and B-spline functions, but introduces recurrent refinement before the spline coefficients participate in the final aggregation. Specifically, the token-resolved contribution of basis $k$ on edge $(o, i)$ is

$$C_{bnoik} = B_k(x_{bni}) W_{oik}. \tag{2}$$

Standard KAN immediately sums these contributions over $i$ and $k$, after which their individual basis identities are no longer explicitly available. CTKAN therefore constructs a sample-global internal activity state aligned with each $(o, i, k)$ triplet before this aggregation occurs. The resulting gate is later broadcast over tokens, so CTKAN modulates how each sample uses the shared spline coefficients without maintaining a separate recurrent state for every spatial token.

For each input–basis pair, we compute token mean $\mu_{bik}$ and standard deviation $\sigma_{bik}$ of $B_k(x_{bni})$. The raw drive and its sample-wise RMS-normalized form are

$$\widetilde{A}_{boik} = W_{oik}(\mu_{bik} + \eta\sigma_{bik}), \qquad A_{boik} = \frac{\widetilde{A}_{boik}}{\max\left(\sqrt{\frac{1}{OIK}\sum_{o,i,k}\widetilde{A}_{boik}^2}, \epsilon_A\right)}. \tag{3}$$

Here $\eta = 0.5$ and $\epsilon_A = 10^{-6}$. The mean describes basis occupancy across tokens, while the dispersion retains information about heterogeneous responses that mean pooling alone can suppress. The drive is computed once and remains fixed during thought.

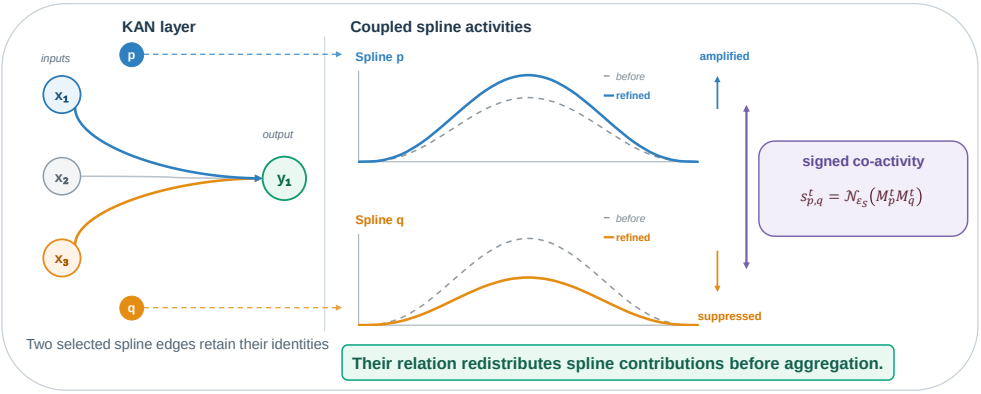



Figure 2: Conceptual view of spline co-activity refinement. Two selected spline edges retain their identities before aggregation. Their signed co-activity is interpreted through learned recurrent feedback and terminal gating, allowing one spline contribution to be amplified while another is suppressed for the current sample. The curves are schematic.

### 3.2 Recurrent co-activity refinement

The memory starts from the current sample with a weak learned prior:

$$M^0_{boik} = \tanh(\gamma_x A_{boik} + \rho P_{oik}), \qquad P = \tanh(z_{\text{init}}), \quad \rho = 0.05. \tag{4}$$

Flattening $(o, i, k)$ gives $D = OIK$ activity units. Reading all $D^2$ relations is impractical, so CTKAN samples $R_a$ coordinate pairs $\mathcal{P}^a = \{(p_r, q_r)\}_{r=1}^{R_a}$ once at initialization and reuses them at every tick. The set combines self-pairs, which measure local activity energy, and cross-pairs, which measure signed coordination between distinct units. At tick $t$,

$$s^t_b = \mathcal{N}_{\epsilon_S}\left(\left[M^t_{b,p_r} M^t_{b,q_r}\right]_{r=1}^{R_a}\right), \tag{5}$$

where $s^t_b$ is the normalized signed co-activity pattern of the sampled spline-activity pairs at tick $t$: positive and negative products indicate same- and opposite-sign activation, while magnitude reflects joint strength. The operator $\mathcal{N}$ centers and RMS-normalizes each sample, preserving relative coordination while suppressing common offset and scale. Fixed pair coordinates maintain temporal correspondence.

Figure 2 illustrates the role of relational readout. The product in Equation 5 provides a signed observation of coordination. The feedback MLP learns from the segmentation objective how each co-activity pattern should modify the evolving triplet states, and the terminal readout converts those states into residual coefficient gates. Consequently, spline responses that support the current prediction can be strengthened, whereas competing responses can be attenuated before aggregation. Relational feedback lets each triplet adapt to the states of other splines while preserving its own history.

The complete descriptor $s^t_b$ is processed by a two-layer MLP that outputs $O + I + K$ values. These are split into output-, input-, and basis-axis feedback and broadcast to form

$$[f^{O,t}_b, f^{I,t}_b, f^{K,t}_b] = \text{Split}(\text{MLP}_a(s^t_b)), \qquad F^t_{boik} = f^{O,t}_{b,o} + f^{I,t}_{b,i} + f^{K,t}_{b,k}. \tag{6}$$

Here $f^{O,t}_b$, $f^{I,t}_b$, and $f^{K,t}_b$ indicate how the current co-activity pattern adjusts the output dimensions, input dimensions, and spline bases, respectively. Their sum gives each triplet $(o, i, k)$ a structured local signal. Thus, from a neural-dynamics perspective, $F^t$ can be interpreted as a structured coupling field that converts population-level co-activity into triplet-level feedback; the next state update combines it with $A$ and $M^t$, closing a read–decode–write recurrent loop.

For hidden width $H$, this axis-wise decoder uses $H(O+I+K)$ rather than $H(OIK)$ output weights. It imposes an additive axis-separable bias that reduces parameters and regularizes feedback.

The candidate state and exponential moving memory are

$$z^{t+1} = \tanh(\gamma_x A + \gamma_m M^t + F^t), \qquad M^{t+1} = \lambda M^t + (1-\lambda) z^{t+1}, \quad \lambda = e^{-1/L}. \tag{7}$$

The transition parameters are shared for $T$ ticks. Hence $T$ controls the number of relational read–write cycles, while $L$ controls the memory time scale without introducing a queue or changing parameter count.

### 3.3 Terminal coefficient modulation

An independently sampled fixed pair set reads the terminal memory. A separate projection $\text{MLP}_o$ produces factors whose broadcast sum is $\Phi^{\text{out}}$. CTKAN then generates

$$G_{boik} = 1 + \alpha \tanh(M^T_{boik} + \Phi^{\text{out}}_{boik}), \qquad W^b_{oik} = W_{oik} G_{boik}, \tag{8}$$

where the learnable residual scale is initialized as $\alpha_0 = 0.05$. Near-identity initialization protects the pretrained form of the KAN mapping while allowing modulation to grow during training. The final spline branch is

$$Y^{\text{spline}}_{bno} = \sum_{i,k} B_k(x_{bni}) W^b_{oik}, \tag{9}$$

and is added to the unchanged base branch in Eq. 1. Because $G$ has no token index, CTKAN is a sample-global coefficient modulator informed by token statistics rather than a token-resolved recurrent operator.

**Synchronization and its ablation.** We define *synchronization* as the complete relational path that forms the normalized pair products in Eq. 5 and decodes them into recurrent and terminal co-activity feedback. The *w/o synchronization* variant removes both pair readouts and their projection MLPs: the recurrent feedback $F^t$ is omitted from Eq. 7, and the terminal feedback $\Phi^{\text{out}}$ is omitted from Eq. 8. It retains the same drive, learned prior, recurrent self-history, number of ticks, EMA coefficient, residual gate scale, deployment layout, and training protocol. This ablation therefore isolates cross-triplet relational feedback from recurrent self-history and temporal memory.

### 3.4 Segmentation instantiation

As shown in Fig. 3, a CTKAN residual token block applies layer normalization before a three-unit hybrid spline layer under the `first` layout. Only the first spline-linear projection uses CTKAN, while the second and third projections remain Static KAN. Each projection is followed by depth-wise convolution, batch normalization, and ReLU. DropPath regularizes the transformed branch before it is added to the identity shortcut. CTKAN modifies the first projection's internal spline computation within the three-unit token-block configuration established by U-KAN.

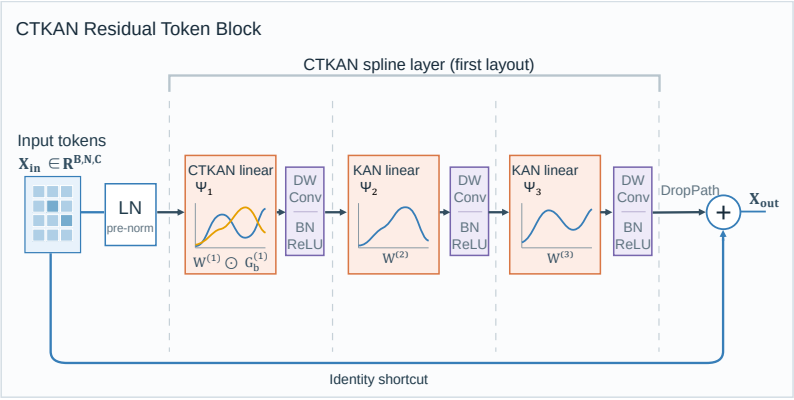



Figure 3: CTKAN residual token block under the `first` layout. The first spline-linear projection uses CTKAN, while the second and third remain Static KAN; all three are followed by depth-wise convolution, batch normalization, and ReLU, with DropPath and an identity shortcut.

We replace selected token-stage KAN blocks in U-KAN while retaining its convolutional stages, skip connections, and resolution transitions. Figure 4 shows the two deployment scales used throughout the experiments. **CTKANLight** places CTKAN at the deepest encoder token stage and the first decoder token stage, while **CTKANMax** applies it at all four token stages. The two variants therefore differ in deployment scale rather than in the proposed spline-level recurrent mechanism. Both are trained end-to-end with the task-appropriate segmentation objective described in Section 4; its gradients propagate through the gated spline aggregation into the recurrent states and feedback projections, without an auxiliary loss on the internal thought trajectory.

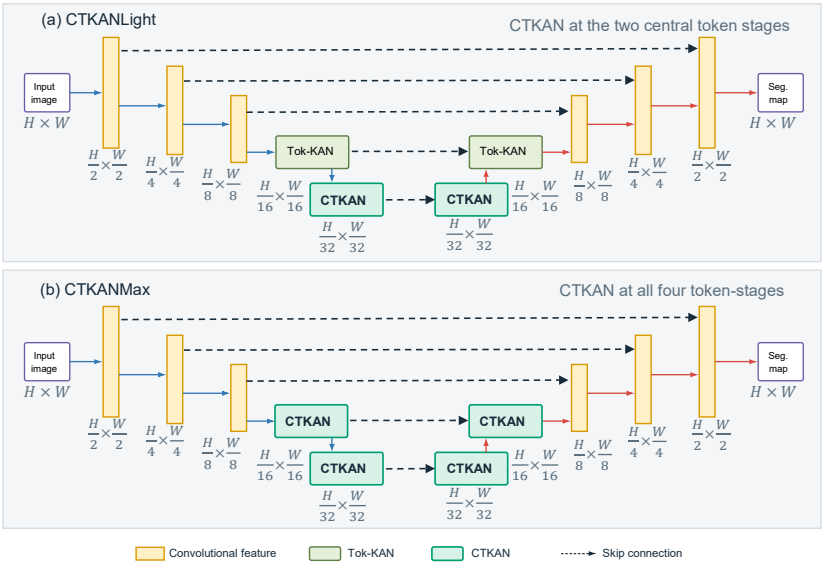



Figure 4: CTKAN deployment scales in the shared U-shaped segmentation backbone. CTKANLight replaces two token-stage KAN blocks, whereas CTKANMax replaces all four; the remaining convolutional stages, skip connections, and resolution transitions are shared.

# 4 Experiments

## 4.1 Setup

We evaluate BUSI malignant-tumor ultrasound segmentation (Al-Dhabyani et al., 2020), CVC-ClinicDB polyp segmentation (Bernal et al., 2015), GlaS gland segmentation (Sirinukunwattana et al., 2017), and ACDC multi-class cardiac-structure segmentation. The test set remains sealed during model and checkpoint selection. We train every model from scratch with five initialization seeds on the same fixed split and report mean $\pm$ sample standard deviation over the five test scores. Four-dataset macro-averages are first computed within each matched seed and then summarized across seeds. The reported dataset-macro score is the unweighted average across the four datasets: BUSI, CVC-ClinicDB, and GlaS use foreground IoU and Dice, while ACDC first averages IoU and Dice over the three foreground structures.

ACDC, BUSI, CVC-ClinicDB, and GlaS are each partitioned at the image level into 70% training, 15% validation, and 15% test subsets. All architecture and dynamics hyperparameters, including $T$, $L$, $R_a$, and $R_o$, are selected exclusively on the validation sets before final test evaluation.

Models are trained for 300 epochs with Adam, batch size 4, initial learning rate $10^{-4}$, cosine annealing, and a task-appropriate segmentation objective. For the three binary datasets, we use binary cross-entropy plus soft Dice. For ACDC, the four output channels correspond to background, right ventricle (RV), myocardium (MYO), and left ventricle (LV); we use multi-class cross-entropy together with a soft Dice term averaged over the three foreground structures, excluding background. CTKAN dynamics use learning rate $10^{-3}$ without weight decay; the matched control variants (PAKAN-style global coefficient pools, LG-KAN-style basis routing weights, and Post-KAN recurrence gates) re-

Table 1: Four-dataset macro-average accuracy and standardized model efficiency. Accuracy is mean ± sample SD over five seeds (%); FPS is mean ± sample SD over three batch-4 repeats. GFLOPs/image uses batch size 1. Best accuracy means are bold. Full dataset-wise results and profiling details are in Appendix B.

| Method | Macro IoU | Macro Dice | Params (M) | GFLOPs | FPS |
|---|---|---|---|---|---|
| CTKANLight | 81.34±0.32 | 89.16±0.27 | 6.73 | 5.48 | 270.27±6.46 |
| CTKANMax | **81.66±0.51** | **89.44±0.44** | 7.07 | 5.51 | 215.15±5.29 |
| U-KAN | 80.37±0.25 | 88.41±0.13 | 6.36 | 5.43 | 361.92±5.22 |
| U-Mamba | 80.09±0.27 | 88.58±0.25 | 47.51 | 52.47 | 450.22±1.01 |
| MedNeXt | 80.87±0.42 | 88.71±0.28 | 10.71 | 35.86 | 368.72±0.19 |
| U-Net | 78.71±0.55 | 87.37±0.49 | 31.04 | 96.33 | 606.99±0.04 |
| Attention U-Net | 79.29±0.41 | 87.69±0.30 | 31.39 | 98.49 | 544.60±0.11 |
| U-Net++ | 78.37±0.28 | 86.56±0.30 | 36.62 | 275.65 | 188.12±0.28 |
| UNeXt | 76.67±0.50 | 86.22±0.45 | 1.47 | 1.12 | 956.15±25.95 |
| Rolling-UNet | 78.50±0.38 | 87.25±0.44 | 7.10 | 16.43 | 48.04±0.51 |

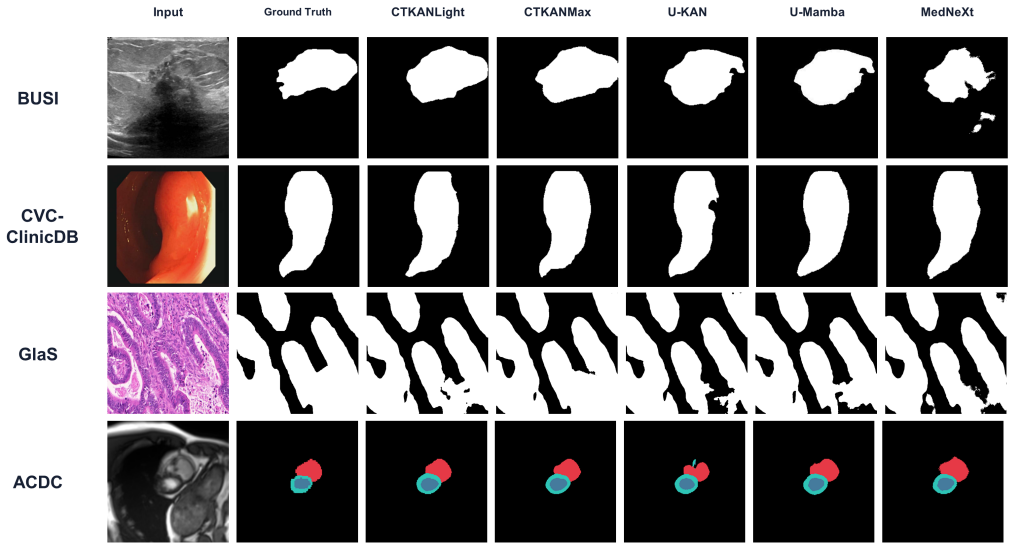



Figure 5: Representative BUSI, CVC-ClinicDB, GlaS, and ACDC predictions. Columns show input, ground truth, CTKANLight, CTKANMax, U-KAN, U-Mamba, MedNeXt, and U-Net. Additional baseline predictions are provided in Appendix B.

ceive the same dynamics learning rate and weight-decay treatment. This matched optimization isolates mechanism differences. Images are resized to $256^2$ for BUSI/CVC/ACDC and $512^2$ for GlaS. Training augmentations are right-angle rotations and horizontal/vertical flips. All reported CTKAN results use the validation-selected configuration $T = 5$ and $L = 5$, with $R_a = R_o = 1024$ and 32 self-pairs in each pair set. Full split, preprocessing, and profiling details are in Appendix A.

### 4.2 Comparison with segmentation baselines

Table 1 shows that CTKANMax achieves the best four-dataset macro-average, with CTKANLight ranking second. Relative to the shared U-KAN backbone, CTKANLight gains 0.97 IoU and 0.75 Dice points, while CTKANMax gains 1.29 IoU and 1.03 Dice points. Both variants add modest operator-counted arithmetic over U-KAN, but recurrent read–update–write reduces realized throughput. CTKANLight therefore offers the preferred accuracy–efficiency balance, whereas CTKANMax prioritizes peak accuracy.

Figure 5 focuses on the two CTKAN variants, their shared U-KAN baseline, representative modern competitors, and U-Net. Across binary and multi-class cases, CTKAN more closely follows fine boundary variations while reducing local under-segmentation, leakage, and fragmented predictions. This behavior is consistent with pre-aggregation refinement of basis contributions, which enables more sample-adaptive nonlinear fitting of local anatomical structure. To keep the main figure readable, predictions from the remaining baselines are moved to Appendix B. We also consider transfer beyond medical imagery on the Freiburg Forest semantic-segmentation benchmark; its dedicated material is separated into Appendix C.

### 4.3 Mechanism ablations

We compare CTKAN with two matched non-recurrent adaptive KAN controls. PAKAN-style global coefficient pooling aggregates tokens and applies a one-shot factorized spline-coefficient gate; LG-KAN-style basis routing predicts token-conditioned basis weights before aggregation (Zhang et al., 2026a; Cheng & Liu, 2026). Every ablation variant is constructed within the same U-KAN backbone using the matched first-projection placement: the first projection in each deep KAN block is replaced by the designated operator, while the second and third projections remain Static KAN. The operator at this matched position is the only variable across ablation rows. Both controls retain the standard per-token Cox–de Boor basis evaluation and spline-coefficient contraction; their global coefficient gate and local basis router add 4.17 and 1.84 MFLOPs/image, respectively. Consequently, all three variants round to $5.43$ operator-aware GFLOPs/image under the same convention. Post-KAN and the remaining removals isolate recurrence location, repeated updates, pairwise relational feedback, and EMA within the same constructor. As defined above, w/o synchronization removes both recurrent and terminal pairwise co-activity feedback while retaining recurrent self-history, repeated updates, and EMA memory.

Appendix E isolates the location constraint with paired synthetic activities: aggregate-only MLP and recurrent predictors remain exactly at chance, whereas predictors that retain the pre-aggregation coordinates recover the label.

Table 2: Four-dataset macro-average CTKANLight mechanism ablations (mean $\pm$ sample SD, %, five matched seeds). All variants use the matched `first` layout: only the first projection in each deep KAN block is replaced, while the other two remain Static KAN. Params are trainable; GFLOPs/image uses batch size 1, and batch-4 FPS is mean $\pm$ sample SD over three repeats. Dataset-wise accuracy is in Appendix B.

| Variant | Dataset-macro IoU | Dataset-macro Dice | Params (M) | GFLOPs/image | FPS |
|---|---|---|---|---|---|
| Static KAN | $80.22\pm0.88$ | $88.28\pm0.60$ | 6.36 | 5.43 | $362.40\pm2.39$ |
| PAKAN-style global coeff. | $80.42\pm0.33$ | $88.50\pm0.48$ | 6.54 | 5.43 | $347.38\pm4.11$ |
| LG-KAN-style basis route | $80.67\pm0.41$ | $88.86\pm0.62$ | 6.36 | 5.43 | $331.29\pm4.02$ |
| Post-KAN recurrence | $80.50\pm0.30$ | $88.58\pm0.34$ | 6.42 | 5.43 | $318.84\pm0.40$ |
| CTKAN ($T=1$) | $80.28\pm0.74$ | $88.34\pm0.25$ | 6.73 | 5.45 | $316.86\pm3.41$ |
| w/o synchronization | $80.55\pm0.25$ | $88.67\pm0.18$ | 6.73 | 5.47 | $326.49\pm6.95$ |
| w/o EMA | $81.01\pm0.42$ | $88.83\pm0.14$ | 6.73 | 5.47 | $270.03\pm6.13$ |
| Full CTKAN | $\mathbf{81.47\pm0.15}$ | $\mathbf{89.63\pm0.50}$ | 6.73 | 5.48 | $274.96\pm0.23$ |

The two controls improve over Static KAN by 0.20/0.22 and 0.45/0.58 dataset-macro IoU/Dice points. Full CTKAN remains 1.05/1.13 and 0.80/0.77 points ahead, indicating an additional contribution from recurrent co-activity evolution beyond one-shot coefficient adaptation and local basis routing. It also exceeds Post-KAN by 0.97/1.05; using one tick or removing synchronization/EMA costs 1.19/1.29, 0.92/0.96, and 0.46/0.80 points. Under the unified protocol, throughput falls from $362.40\pm2.39$ to $274.96\pm0.23$ FPS despite only a small GFLOP increase, reflecting sequential-state overhead.

In the ablation study, the "Full CTKAN" and "Static KAN" rows use exactly the same architectures as CTKANLight and U-KAN in the main table, respectively, and differ only in their independent random-seed sets and evaluation batches.

The larger Dice gain is consistent with the more coherent foreground recovery in Figure 7 of Appendix B; dataset-wise results are in Appendix B. Internal diagnostics show an update ratio decreasing from $0.998$ at $t=1$ to $0.285$ at $t=5$, basis-specific amplification and suppression, and progressive

basis-specific memory growth (Figure 9(a–c) in Appendix D). The mean absolute terminal-gate deviation is $0.058$ over all 42 BUSI validation samples in the first CTKAN projection. Together, these diagnostics confirm non-identity, recurrent, and basis-selective coefficient modulation.

To verify that these gaps exceed seed-level noise, we apply the same two-level group-paired bootstrap (five matched seeds, identical test-group draws, 10,000 replicates) to Full CTKAN minus each control using the group identifiers in the supplementary reproducibility package. Every interval excludes zero: vs. LG-KAN-style basis route $0.80$ ([0.46, 1.14]), vs. PAKAN-style global coefficient $1.05$ ([0.70, 1.40]), vs. Post-KAN recurrence $0.97$ ([0.62, 1.32]), vs. w/o EMA $0.46$ ([0.14, 0.78]), vs. w/o synchronization $0.92$ ([0.58, 1.26]), and vs. CTKAN ($T$=1) $1.19$ ([0.83, 1.55]). The gap against the strongest one-shot control (LG-KAN-style basis routing) is therefore 0.80 IoU with an interval excluding zero; the smallest ablation gap is 0.46 versus w/o EMA, whose interval also excludes zero, and every Full-minus-control difference is positive on all four datasets.

### 4.4 Thought–memory coupling and efficiency

Figure 10 in Appendix D shows non-monotonic coupling between $T$ and $L$: equal $T/L$ need not behave equally because co-activity and candidate states are recomputed at every tick. The validation means at $(5, 5)$ and $(20, 3)$ differ by only 0.06 IoU and 0.07 Dice points. We select $(5, 5)$ because it remains within 0.1 percentage points of the best observed mean while providing higher throughput than $(20, 3)$ ($275.50\pm0.39$ versus $182.36\pm0.90$ FPS). Appendix D also reports activity-pair sensitivity.

## 5 Conclusion

CTKAN makes spline computation stateful before aggregation: sample-conditioned triplet states evolve through co-activity feedback and exponential memory, then modulate shared spline coefficients through residual gates. Across four medical image segmentation benchmarks, CTKAN achieves positive mean improvements over the base U-KAN backbone on every dataset. Controlled ablations further support the effectiveness of spline-space placement, recurrence, relational feedback, and memory. This study uses token-global dynamics and evaluates ACDC, BUSI, CVC-ClinicDB, and GlaS at the 2D slice level. Future work will incorporate cross-token and cross-scale context and extend validation beyond medical image segmentation.

### AI use statement

Generative AI assisted language editing, code organization, LaTeX migration, and figure preparation. The authors verified all content against source materials and take responsibility for the final paper and artifacts.

### Ethics statement

This study uses public, de-identified benchmarks, collects no new patient data, and makes no clinical decisions. Clinical use would require independent validation, subgroup analysis, and institutional and regulatory review.

### Reproducibility statement

Section 4 and Appendix A specify the full protocol. Anonymized code will be released.

## References

Walid Al-Dhabyani, Mohammed Gomaa, Hussein Khaled, and Aly Fahmy. Dataset of breast ultrasound images. *Data in Brief*, 28:104863, 2020. doi: 10.1016/j.dib.2019.104863.

Ali Alshami, Caspar Domingi, Alexander Kudryavtsev, Dmitry Onishchenko, Liam Maguire, T. Martin McGinnity, and Jim Harkin. DE-KAN: A kolmogorov-arnold network with dual encoder for 2d teeth segmentation. SSRN preprint, 2025.

Jorge Bernal, F. Javier Sánchez, Gloria Fernández-Esparrach, Debora Gil, Cristina Rodríguez, and Fernando Vilariño. WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. *Computerized Medical Imaging and Graphics*, 43:99–111, 2015. doi: 10.1016/j.compmedimag.2015.02.007.

Hang Chen, Wenjia Yang, Eigil Samset, and Ziyu Wang. MM-UKAN++: Multi-dimensional attention enhanced U-KAN with frequency, channel, and spatial perception for medical ultrasound segmentation. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2025. doi: 10.1109/TUFFC.2025.3588181.

Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L. Yuille, and Yuyin Zhou. TransUNet: Transformers make strong encoders for medical image segmentation, 2021.

Zhe Cheng and Bin Liu. Mitigating distal interference in continual learning through locally gated kolmogorov–arnold networks. *Neurocomputing*, 699:134392, 2026. doi: 10.1016/j.neucom.2026.134392.

Luke Darlow, Ciaran Regan, Sebastian Risi, Jeffrey Seely, and Llion Jones. Continuous thought machines, 2025.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. doi: 10.1207/s15516709cog1402_1.

Remi Genet and Hugo Inzirillo. TKAN: Temporal kolmogorov–arnold networks, 2024.

Chenxin Li, Xinyu Liu, Wuyang Li, Cheng Wang, Hengyu Liu, Yifan Liu, Zhen Chen, and Yixuan Yuan. U-KAN makes strong backbone for medical image segmentation and generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. KAN: Kolmogorov-arnold networks, 2024.

Xiaoyan Lu, Xun Gong, and Yuanyuan Chen. GDKansformer: A group-wise dynamic kolmogorov–arnold transformer with multi-view gated attention for pathological image diagnosis. *Expert Systems with Applications*, 286:127978, 2025. doi: 10.1016/j.eswa.2025.127978.

Jun Ma, Feifei Li, and Bo Wang. U-Mamba: Enhancing long-range dependency for biomedical image segmentation, 2024.

Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14 (11):2531–2560, 2002. doi: 10.1162/089976602760407955.

Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y. Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention U-Net: Learning where to look for the pancreas, 2018.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volume 9351 of *Lecture Notes in Computer Science*, pp. 234–241. Springer, 2015. doi: 10.1007/978-3-319-24574-4_28.

Korsuk Sirinukunwattana, Josien P. W. Pluim, Hao Chen, Xiaojuan Qi, Pheng-Ann Heng, Yun Bo Guo, Li Yang Wang, Bogdan J. Matuszewski, Elia Bruni, Urko Sanchez, Anton Böhm, Olaf Ronneberger, Bassem Ben Cheikh, Daniel Racoceanu, Philipp Kainz, Michael Pfeiffer, Martin Urschler, David R. J. Snead, and Nasir M. Rajpoot. Gland segmentation in colon histology images: The GlaS challenge contest. *Medical Image Analysis*, 35:489–502, 2017. doi: 10.1016/j.media.2016.08.008.

Hao Tang, Xingwei Liu, Shanlin Sun, Xiangyi Yan, and Xiaohui Xie. Recurrent mask refinement for few-shot medical image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3918–3928, 2021.

Wei Wang, Kaicheng Yu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Recurrent U-Net for resource-constrained segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2142–2151, 2019.

Yanlin Wu, Tao Li, Zhihong Wang, Hong Kang, and Along He. TransUKAN: Computing-efficient hybrid KAN-transformer for enhanced medical image segmentation, 2024.

Zanting Ye, Xiaolong Niu, Xuanbin Wu, Wenxiang Yi, Yuan Chang, and Lijun Lu. Semi-KAN: KAN provides an effective representation for semi-supervised learning in medical image segmentation, 2025.

Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. CANet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5217–5226, 2019.

Haoyu Zhang, Haojing Chen, Zhen Zhong, and Liangjian Deng. PAKAN: Pixel adaptive kolmogorov–arnold network modules for pansharpening, 2026a.

Zhen Zhang, Yifan Liu, Quande Li, Zeyu Zhang, Yonghao Liu, Yuxuan Wang, Lequan Lu, and S. Kevin Zhou. UUEKAN: Edge-enhanced and uncertainty-guided efficient kolmogorov-arnold network for medical image segmentation. *Computerized Medical Imaging and Graphics*, 124: 110649, 2026b. doi: 10.1016/j.compmedimag.2025.110649.

Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. UNet++: A nested U-Net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, volume 11045 of *Lecture Notes in Computer Science*, pp. 3–11. Springer, 2018. doi: 10.1007/978-3-030-00889-5_1.

## A Additional formulation and implementation details

**Pair normalization.** For a sampled product vector $u_b \in \mathbb{R}^R$, CTKAN uses

$$\mathcal{N}_\epsilon(u_b)_r = \frac{u_{b,r} - \bar{u}_b}{\sqrt{R^{-1}\sum_j (u_{b,j} - \bar{u}_b)^2 + \epsilon^2}}, \qquad \bar{u}_b = R^{-1}\sum_r u_{b,r}. \tag{10}$$

We use $\epsilon_S = 10^{-4}$. Action and terminal gate pairs are sampled independently at model initialization and remain fixed across samples and ticks. Unless otherwise stated, $R_a = R_o = 1024$ and each set includes 32 self-pairs. The transition and output projections are two-layer MLPs with hidden width 64 and dropout 0.2. Their final layers use truncated-normal initialization with standard deviation $10^{-3}$ and zero bias. The gains $\gamma_x$ and $\gamma_m$ are initialized to 1 and 0.5, respectively.

**Data protocol.** BUSI and CVC-ClinicDB images are resized to $256 \times 256$, GlaS images to $512 \times 512$, and ACDC images to $256 \times 256$. Each dataset uses one fixed split. All models and five initialization seeds use identical sample partitions. Checkpoints and hyperparameters are selected on the validation set, and the held-out test set is evaluated only after model selection. Training augmentation includes random right-angle rotations and horizontal and vertical flips. Images are normalized using ImageNet channel statistics. Image resizing uses bilinear interpolation, and mask resizing uses nearest-neighbor interpolation. Binary predictions use a threshold of 0.5. ACDC predictions use the channel-wise argmax, and ACDC metrics are macro-averaged over the three foreground structures.

**Optimization treatment.** For every method, the shared backbone parameters—encoder, decoder, skip connections, and Static KAN projections—use a base learning rate of $10^{-4}$ and Adam weight decay of $10^{-5}$.

**Complexity measurement.** All efficiency metrics are remeasured with a unified pipeline on an NVIDIA GeForce RTX 5090 using PyTorch 2.7.1, CUDA 12.8, FP32 eager inference, and `torch.inference_mode()`. Every measurement uses the BUSI task configuration with three input channels, one output channel, and spatial size $256 \times 256$. These are standardized profiling results rather than dataset-specific inference costs. Operator-aware counting includes convolution,

linear layers, Cox–de Boor basis evaluation, spline mappings, token statistics, pair products and normalization, factor projections, recurrent state updates, exponential moving average (EMA), and terminal gating. FPS uses batch size 4. Each repeat includes 30 warm-up inferences followed by 100 timed inferences, and we report the mean ± sample standard deviation over three repeats. Every method is measured through its own experiment constructor.

## B Full comparisons and mechanism ablations

Table 3: Dataset-wise test performance (%). Values are mean ± sample standard deviation over five runs. ACDC scores are foreground-macro averages over RV, MYO, and LV. Best means are bold.

| Method | BUSI | | CVC | | GlaS | | ACDC | |
|---|---|---|---|---|---|---|---|---|
| | IoU | Dice | IoU | Dice | IoU | Dice | Fg. mIoU | Fg. mean Dice |
| CTKANLight | 65.26±1.18 | 78.13±0.44 | 87.93±0.46 | 93.38±0.43 | 87.66±0.16 | 93.40±0.11 | 84.49±0.69 | 91.75±0.70 |
| CTKANMax | **65.78±0.48** | **78.91±0.20** | 88.15±0.82 | 93.56±0.65 | **87.79±0.29** | **93.47±0.21** | 84.94±1.07 | **91.81±2.03** |
| U-KAN | 63.17±0.26 | 76.61±0.12 | 87.15±0.40 | 92.91±0.25 | 87.48±0.14 | 93.28±0.08 | 83.69±1.11 | 90.84±0.51 |
| U-Mamba | 63.00±0.27 | 77.30±0.27 | 85.91±0.80 | 92.42±0.79 | 87.35±0.47 | 93.25±0.20 | 84.11±0.65 | 91.36±0.83 |
| MedNeXt | 62.98±1.26 | 76.31±0.83 | **88.31±0.76** | **93.72±0.60** | 87.21±0.74 | 93.13±1.10 | **84.98±1.07** | 91.66±0.51 |
| U-Net | 61.18±0.79 | 75.06±1.25 | 84.68±0.97 | 91.44±0.82 | 87.00±0.11 | 93.01±0.07 | 81.99±1.05 | 89.98±0.57 |
| Attention U-Net | 61.35±0.87 | 75.00±0.46 | 85.42±1.78 | 91.84±1.05 | 86.82±0.38 | 92.92±0.20 | 83.59±0.63 | 91.01±0.66 |
| U-Net++ | 59.80±0.46 | 72.03±0.66 | 84.97±0.77 | 91.35±1.03 | 86.21±0.40 | 92.55±0.31 | 82.48±0.63 | 90.32±0.84 |
| UNeXt | 60.77±1.81 | 75.05±1.38 | 80.36±0.61 | 88.82±0.43 | 83.35±0.41 | 90.87±0.15 | 82.20±0.98 | 90.15±0.94 |
| Rolling-UNet | 59.73±0.70 | 74.33±0.36 | 84.31±1.24 | 91.13±0.71 | 86.61±0.38 | 92.80±0.13 | 83.36±1.33 | 90.74±1.32 |

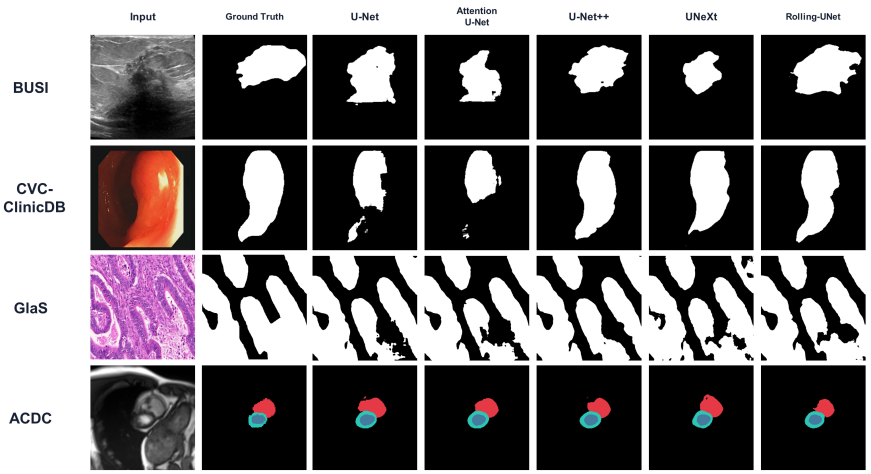



Figure 6: Additional qualitative comparisons on BUSI, CVC-ClinicDB, GlaS, and ACDC. Columns show input, ground truth, Attention U-Net, U-Net++, UNeXt, and Rolling-UNet. The complementary primary comparison is shown in Figure 5.

Table 4: Dataset-wise CTKANLight mechanism ablations (%). Values are mean ± sample standard deviation over five runs. Binary datasets report foreground IoU and Dice; ACDC reports foreground mIoU and foreground mean Dice over RV, MYO, and LV. Best means are bold.

| Variant | BUSI | | CVC | | GlaS | | ACDC | |
|---|---|---|---|---|---|---|---|---|
| | IoU | Dice | IoU | Dice | IoU | Dice | Fg. mIoU | Fg. mean Dice |
| Static KAN | 63.26±0.95 | 76.84±0.72 | 87.26±0.88 | 92.74±0.65 | 87.02±1.02 | 92.99±1.05 | 83.34±0.98 | 90.55±0.71 |
| PAKAN-style global coeff. | 63.56±1.02 | 77.09±0.91 | 87.52±0.63 | 93.03±0.86 | 87.14±0.67 | 93.19±0.76 | 83.46±0.77 | 90.69±0.62 |
| LG-KAN-style basis route | 63.92±1.06 | 77.68±0.97 | 87.68±1.00 | 93.20±0.92 | 87.30±0.54 | 93.44±0.51 | 83.78±0.86 | 91.12±0.78 |
| Post-KAN recurrence | 63.68±0.79 | 77.22±0.79 | 87.57±0.65 | 93.06±0.53 | 87.19±0.95 | 93.25±0.75 | 83.56±0.55 | 90.79±0.60 |
| CTKAN ($T$=1) | 63.32±0.65 | 76.90±1.01 | 87.32±1.03 | 92.80±0.62 | 87.08±0.84 | 93.05±0.58 | 83.40±0.51 | 90.61±0.62 |
| w/o synchronization | 63.94±0.69 | 77.62±0.60 | 87.23±0.54 | 92.83±0.76 | 87.10±0.88 | 93.01±0.81 | 83.95±1.00 | 91.22±0.85 |
| w/o EMA | 64.67±0.65 | 77.95±0.74 | 87.82±0.53 | 92.71±1.03 | 87.35±0.80 | 93.25±0.99 | 84.20±0.82 | 91.42±0.84 |
| Full CTKAN | **65.02±0.79** | **79.00±1.01** | **88.18±0.66** | **93.55±0.77** | **87.79±0.70** | **93.96±0.98** | **84.89±1.03** | **92.00±0.82** |

Full CTKAN ranks first in both metrics on every dataset. The magnitude varies with image modality and target structure, but the direction of the complete-model gain is consistent with the macro-average reported in Table 2.

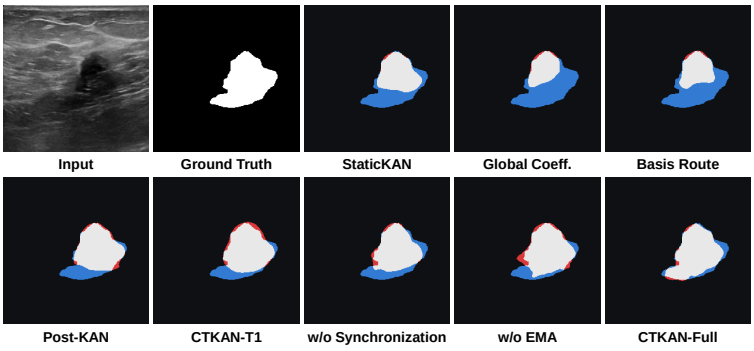



Figure 7: BUSI mechanism ablation; white, red, and blue denote true positives, false positives, and false negatives.

## C Non-medical semantic segmentation

We additionally evaluate transfer beyond medical imagery on the Freiburg Forest semantic-segmentation benchmark. Table 5 reports the available five-run results, while Figure 8 provides a representative qualitative comparison.

Table 5: Freiburg Forest semantic-segmentation results (%). Values are mean ± sample standard deviation over five runs. Best and second-best means are bold and underlined, respectively.

| Method | IoU | Dice |
|---|---|---|
| CTKANLight | <u>80.14±2.21</u> | <u>87.63±0.94</u> |
| CTKANMax | **80.97±1.17** | **88.28±2.73** |
| U-KAN | 79.34±0.98 | 86.83±0.77 |
| U-Mamba | 79.73±0.59 | 87.18±0.57 |
| U-Net | 78.87±1.11 | 86.70±1.21 |
| Attention U-Net | 79.99±1.78 | 87.40±0.72 |
| U-Net++ | 79.73±0.68 | 87.18±1.13 |
| UNeXt | 76.89±0.71 | 84.80±0.54 |
| Rolling-UNet | 71.02±1.07 | 75.23±1.55 |

On this benchmark, CTKANMax obtains the highest mean IoU and Dice (80.97 and 88.28), followed by CTKANLight (80.14 and 87.63). Relative to the shared U-KAN baseline, CTKANMax improves IoU/Dice by 1.63/1.45 points, and CTKANLight by 0.80/0.80 points. This single-benchmark, five-run evaluation provides descriptive evidence that the refinement mechanism transfers beyond medical images.

The example shows broadly consistent large-scale class layouts across methods, with local differences around the narrow side path and the vegetation–road boundary. It visualizes these local differences, while Table 5 provides the quantitative comparison.

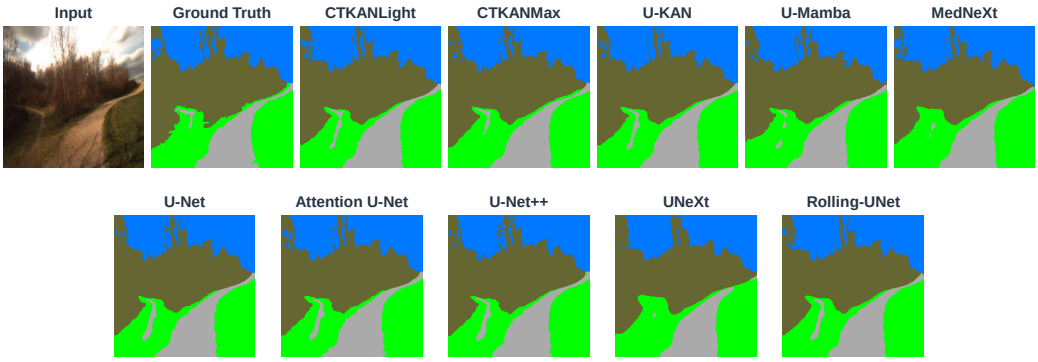



Figure 8: Representative Freiburg Forest predictions. The top row shows the input, ground truth, CTKANLight, CTKANMax, U-KAN, U-Mamba, and MedNeXt; the bottom row shows U-Net, Attention U-Net, U-Net++, UNeXt, and Rolling-UNet.

## D Parameter analysis

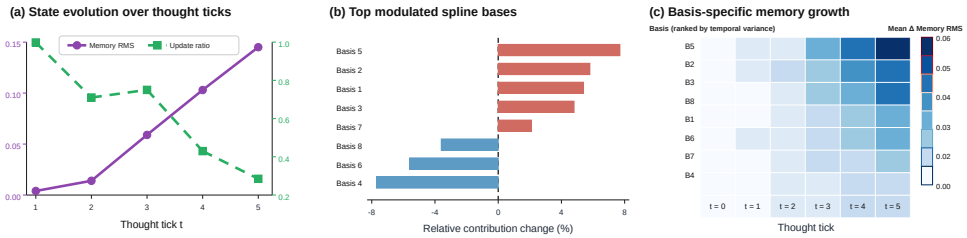



Figure 9: Mechanism diagnostics of CTKANLight ($T$=5, $L$=5) on the BUSI validation set. (a) Memory RMS and update ratio over thought ticks. (b) For a traced validation sample in the first deep-encoder CTKAN unit, absolute token-averaged contributions are averaged over the output and input dimensions for each basis, and their relative pre/post change is then reported; red and blue mark amplification and suppression. (c) Mean per-basis memory-RMS change $\Delta r_{t,k} = r_{t,k} - r_{0,k}$, averaged over six traced validation samples and the six CTKAN units; rows are ordered by temporal variance.

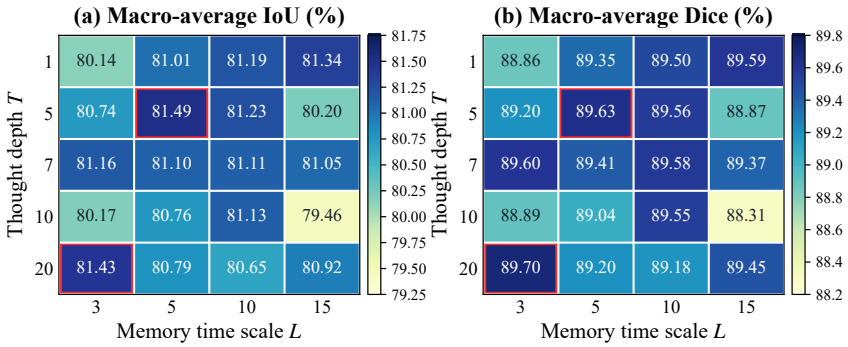



Figure 10: Joint validation sensitivity to thought depth $T$ and memory time scale $L$, averaged over five seeds. Cells report the four-dataset macro-average; red outlines mark configurations within 0.1 percentage points of the best observed mean. Hyperparameter selection uses the validation sets exclusively.

Table 6: Accuracy–cost trade-off across thought depths. Each row uses the $L$ with the highest IoU in the joint grid. All configurations have the same number of trainable parameters.

| $T$ | $L$ | IoU | Dice | Params (M) | GFLOPs/image | FPS |
|---|---|---|---|---|---|---|
| 1 | 15 | 81.34 | 89.59 | 6.73 | 5.45 | 312.26±8.28 |
| 5 | 5 | **81.49** | 89.63 | 6.73 | 5.48 | 275.50±0.39 |
| 7 | 3 | 81.16 | 89.60 | 6.73 | 5.50 | 256.91±0.24 |
| 10 | 10 | 81.13 | 89.55 | 6.73 | 5.52 | 229.98±7.31 |
| 20 | 3 | 81.43 | **89.70** | 6.73 | 5.60 | 182.36±0.90 |

The parameter count remains 6.73 M for every setting because the same CTKANLight architecture and shared recurrent transition are used throughout, while $L$ changes only the fixed EMA coefficient and introduces no trainable parameters. GFLOPs increase from 5.45 to 5.60 as $T$ grows because the shared transition is executed more times; changing $L$ leaves the computation graph unchanged. GFLOPs and FPS follow the unified protocol above: batch-1 GFLOPs and batch-4 FPS with 30 warm-up passes, 100 timed inferences, and three repeats on an RTX 5090.

Table 7: Sensitivity to the action-pair budget, with 32 self-pairs. Validation-set accuracy (IoU/Dice) guides hyperparameter selection. Efficiency is measured under the unified protocol: batch-1 GFLOPs and batch-4 FPS, with three FPS repeats.

| $R_a$ | Cross-pairs | IoU | Dice | Params (M) | GFLOPs/image | FPS |
|---|---|---|---|---|---|---|
| 256 | 224 | 80.88 | 89.37 | 6.63 | 5.48 | 277.58±0.20 |
| 512 | 480 | 81.26 | 89.59 | 6.66 | 5.48 | 277.26±0.08 |
| 1024 | 992 | **81.52** | **89.71** | 6.73 | 5.48 | 277.19±0.14 |
| 2048 | 2016 | 81.32 | 89.58 | 6.86 | 5.48 | 266.90±7.33 |

Increasing $R_a$ from 256 to 1024 improves relation coverage and raises IoU/Dice from 80.88/89.37 to 81.52/89.71, while the parameter count increases only from 6.63M to 6.73M and the measured arithmetic remains 5.48 GFLOPs/image after rounding. The batch-4 throughput is also stable from $R_a = 256$ to 1024 (277.58 to 277.19 FPS). Increasing the budget to 2048 adds parameters and produces a measurable throughput drop to 266.90 FPS, yet reduces IoU/Dice to 81.32/89.58. We therefore select $R_a = 1024$ as the best observed accuracy–efficiency operating point under this validation protocol: it provides the highest IoU/Dice without a measurable arithmetic or throughput penalty relative to $R_a = 512$.

## E Controlled aggregation-collision study

**Proposition.** Flatten the basis-wise contributions for one output into $c \in \mathbb{R}^D$ and let the fixed KAN aggregation be $A(c) = \mathbf{1}^\top c$. For every nonzero $u \in \ker A$, $A(c) = A(c+u)$. Consequently, any deterministic post-aggregation predictor $h \circ A$ produces the same prediction for $c$ and $c+u$. On a balanced dataset containing such pairs with opposite labels, it is correct for exactly one member of each pair and therefore cannot exceed $50\%$ accuracy. A pre-aggregation predictor is not subject to this identifiability constraint because it still observes the coordinates of $c$.

**Paired construction.** For each pair, we sample $s \sim \mathcal{U}(-0.5, 0.5)$ and $a \sim \mathcal{U}(1, 2)$, set $b = s/4$, and construct

$$c^{(0)} = (b+a, b+a, b-a, b-a), \qquad c^{(1)} = (b+a, b-a, b+a, b-a). \tag{11}$$

Both activities aggregate to $s$, but the noiseless statistic $c_1 c_2$ is positive for class 0 and negative for class 1. We add the same Gaussian perturbation to both members after subtracting its coordinate mean, so the perturbation is zero-sum and cannot leak the label through $A(c)$. The maximum paired aggregate discrepancy in float32 is $2.38 \times 10^{-7}$.

**Protocol and result.** We use $10{,}000/2{,}000/10{,}000$ balanced training/validation/test examples, Gaussian noise with standard deviation $0.05$, and five initialization seeds. All learned controls use

hidden width 32 and Adam with learning rate $10^{-3}$ for 80 epochs; recurrent models use $T = 5$ and $L = 5$. Aggregate-MLP and Post-Aggregation-RNN receive only $A(c)$; Pre-Aggregation-MLP receives $c$; and CTKAN-Activity is the controlled $O = I = 1, K = 4$ activity-state special case with normalized pair products, recurrent memory, terminal residual gating, and gated aggregation.

Table 8: Aggregation-collision classification accuracy (mean $\pm$ sample SD over five seeds). Aggregate-only predictors receive identical inputs within every opposite-label pair, so the impossibility result is independent of parameter count.

| Model | Observed representation | Params | Test accuracy (%) |
|---|---|---|---|
| Aggregate-MLP | $A(c)$ | 97 | $50.00 \pm 0.00$ |
| Post-Aggregation-RNN | $A(c)$ | 1,121 | $50.00 \pm 0.00$ |
| Pre-Aggregation-MLP | $c$ | 193 | $100.00 \pm 0.00$ |
| CTKAN-Activity | $c$ | 3,093 | $99.99 \pm 0.01$ |

The chance-level aggregate-only controls and near-perfect pre-aggregation models confirm the proposition and isolate the *placement* constraint: retaining basis-wise coordinates enables disambiguation of colliding activity vectors. Table 2 isolates *recurrent co-activity dynamics*: Post-KAN recurrence, a single thought tick, and removal of EMA reduce macro-IoU by 0.97, 1.19, and 0.46 points, respectively, while preserving the same first-projection spline placement.