DEFORMABLE KOLMOGOROV-ARNOLD NETWORKS FOR MEDICAL IMAGE SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Medical image segmentation remains difficult when boundaries are faint and anatomy varies across modalities. U-Net variants replace the bottleneck with a transformer, an MLP, a state-space model, or a Kolmogorov-Arnold network (KAN). These variants still evaluate that operator on the encoder's fixed, downsampled grid. We present UKAD, a deformable KAN encoder-decoder. Its Kolmogorov-Arnold Displacement (KAD) bottleneck predicts a content-adaptive grid, resamples the bottleneck features on that grid, and then applies a learnable activation. That activation is a low-degree group-rational function, and we share it within each channel group, so its parameter count does not grow with network width. UKAD also uses it in the skip gates and in a hybrid decoder. We train UKAD in one stage, without pretraining or distillation, and with one hyperparameter setting across datasets, resolutions, and widths. We retrain fourteen U-Net, MLP, state-space, and KAN baselines on four public datasets under one protocol. UKAD-L has the highest mean IoU and Dice, 83.11 and 90.25. AdaKAN reaches 82.36 and 89.78, and Rolling-UNet reaches 82.25 mean IoU. UKAD-B reaches 82.59 mean IoU with 45% fewer parameters than AdaKAN. We release the code at https://anonymous.4open.science/r/UKAD.

1 INTRODUCTION

Medical image segmentation delineates lesions, polyps, glands, and other structures that clinicians use in computer-aided diagnosis and image-guided intervention. The U-Net (Ronneberger et al., 2015) established the standard design: a downsampling encoder, an upsampling decoder, and skip connections that carry encoder features to the matching decoder stage. Later work changes one part of that design at a time. Some papers change the skip pathway (Zhou et al., 2018; Oktay et al., 2018). Others change the operator that mixes bottleneck features (Valanarasu & Patel, 2022; Liu et al., 2024a; Ma et al., 2024b). Self-configuring pipelines tune the training recipe around the same encoder-decoder (Isensee et al., 2021). Promptable foundation models train one prompt-conditioned model across modalities (Ma et al., 2024a).

Three lines of work replace the bottleneck operator, each at a different cost. Transformer bottlenecks widen the receptive field, at a cost quadratic in the number of tokens (Chen et al., 2021; Cao et al., 2021; Hatamizadeh et al., 2022). State-space bottlenecks replace attention with sequence-style feature mixing (Ma et al., 2024b; Ruan et al., 2024). Tokenized and axis-wise MLP blocks reduce that cost, but they mix features with fixed nonlinearities (Valanarasu & Patel, 2022; Liu et al., 2024a). In all three, the activation stays fixed, and the features stay on the encoder's downsampled grid.

Kolmogorov-Arnold Networks (KANs) change the activation (Kolmogorov, 1957; Liu et al., 2025). They place a learnable univariate function on each connection instead of a fixed activation on each node. Segmentation networks have adopted that idea in several forms. U-KAN puts tokenized KAN blocks in the deepest encoder and decoder stages (Li et al., 2025). UKAGNet uses convolutional KAN layers (Drokin, 2024). U-FunKAN uses a Hermite basis (Penkin & Krylov, 2025). AdaKAN and UKAN+ pair KAN layers with attention (Alzu'bi et al., 2026; Tian et al., 2026). These models learn the activation, but they evaluate it on the fixed encoder grid. In the protocol-matched comparison in Section 4, UKAGNet reaches 76.24 mean IoU and U-FunKAN reaches 75.20, both below the 79.97 of a plain U-Net. Yang & Wang (2025) report a related result when a KAN layer replaces the MLP in a transformer. We ask which design choices make a learnable activation useful in a U-Net.

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.

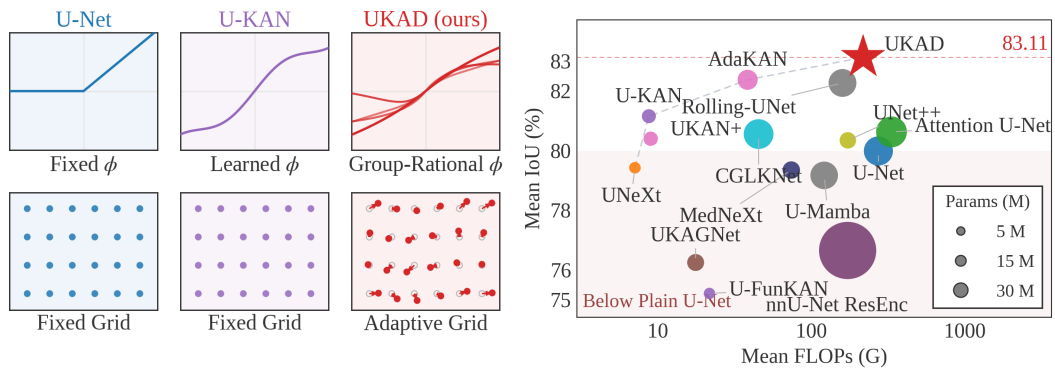

Figure 1: Design contrast and the trade-off between accuracy and compute. The left panel contrasts the fixed-grid U-Net and KAN operators with the adaptive grid and group-rational ϕ of UKAD. The right panel plots four-dataset mean IoU against FLOPs, and bubble size denotes parameters.

To answer that question, we present UKAD, a U-shaped encoder-decoder whose bottleneck is the Kolmogorov-Arnold Displacement (KAD) block. UKAD is deformable because the grid on which it evaluates the learnable activation moves with the image. Figure 1 shows the fixed grid that prior KAN U-Nets use.

The first choice is the sampling grid. Most current KAN U-Nets still sample on the encoder's downsampled grid, so the sampling locations cannot follow the image. Learned resampling moves a grid, but it applies a fixed activation to the resampled features (Jaderberg et al., 2015; Dai et al., 2017; Hu et al., 2026). The KAD block predicts a displacement from the bottleneck map and resamples that map before the learnable activation. At 256×256 input resolution, the $H/16$ bottleneck permits shifts of up to 48 input-image pixels. A displacement branch runs beside a context branch and a local depthwise branch. Section 3 defines the gate that scales the two adaptive branches. A boundary head refines the mask where the prediction is uncertain.

The second choice is the basis. A spline or a high-degree polynomial needs parameters for every input-output channel pair, and it needs a stable initialization (Yang & Wang, 2025). U-GRKAN adopts a group-rational basis but keeps the fixed grid (Wu et al., 2025). UKAD shares a low-degree rational function within each channel group, and each instance has 48 parameters at any width.

The third choice is which layers use the activation. The attention gate that passes encoder features to the decoder uses ReLU (Oktay et al., 2018), so the step nearest the final mask boundary does not use a learnable activation. UKAD uses the same group-rational activation in those skip gates and in a hybrid decoder of residual and KAN convolutions. UKAD trains in a single stage, with no pretraining and no distillation.

We compare UKAD with fourteen U-Net and KAN baselines on BUSI (Al-Dhabyani et al., 2020), CVC-ClinicDB (Bernal et al., 2015), GlaS (Sirinukunwattana et al., 2017), and ISIC 2018 (Codella et al., 2019; Tschandl et al., 2018), under one shared split and training protocol. Section 4 reports that comparison. The right panel of Figure 1 summarizes the same comparison: it plots four-dataset mean IoU against FLOPs, and the bubble size is the parameter count. UKAD-L is the highest point, at 83.11 mean IoU. UKAD-B reaches 82.59 mean IoU at 7.77 M parameters, which is 45% fewer than AdaKAN at 82.36 mean IoU, while its FLOPs are higher because the displacement and the hybrid decoder run on the spatial grid. UKAD-S, at 1.49 M parameters, lies above the parameter-matched UNeXt.

The contributions of this paper are threefold. First, we identify three choices that most current KAN U-Nets leave fixed: where they evaluate the learnable activation, which basis they use, and which parts of the network use it. Second, we present UKAD, a single-stage encoder-decoder. Its KAD block resamples features on a predicted displacement field before a group-rational activation, and the skip gates and a hybrid decoder reuse that activation. Third, we retrain fourteen U-Net, MLP, state-space, and KAN baselines under one protocol on four datasets, with paired significance tests and component ablations.

2 Related work

Encoder-decoder networks for medical segmentation. Later U-Net variants densify the skips (Zhou et al., 2018), gate them (Oktay et al., 2018), or replace the deep blocks with an MLP, a state-space model, or a transformer (Valanarasu & Patel, 2022; Liu et al., 2024a; Ma et al., 2024b; Ruan et al., 2024; Chen et al., 2021; Cao et al., 2021). These variants keep a fixed activation on a fixed sampling grid. UKAD replaces the bottleneck operator and the skip-gate nonlinearity of that family.

Kolmogorov-Arnold networks for segmentation. KANs place a learnable univariate function on each connection (Liu et al., 2025). KAN U-Nets differ in where that function sits and in which basis it uses. The bases include splines, wavelets, and polynomials (Li et al., 2025; Drokin, 2024), Hermite functions (Penkin & Krylov, 2025), and attention hybrids (Alzu'bi et al., 2026; Tian et al., 2026). U-GRKAN (Wu et al., 2025) replaces the B-splines of U-KAN with the group-rational functions of Yang & Wang (2025). Each of these models evaluates the learnable activation on the encoder's downsampled grid. UKAD keeps a group-rational activation of this kind, but it evaluates that activation on a grid that moves with the image.

Content-adaptive sampling. A spatial transformer predicts a displacement from the feature map and resamples the map with a differentiable bilinear sampler (Jaderberg et al., 2015). Deformable convolution predicts an offset at each sampling location (Dai et al., 2017), and its second version adds a learned modulation scalar (Zhu et al., 2019). InternImage predicts offsets and modulation for each channel group (Wang et al., 2023). SFNet warps decoder features along a predicted flow field (Li et al., 2020). CGLKNet places a large-kernel deformable convolution in the decoder (Hu et al., 2026). None of these methods applies a learnable activation to the resampled features.

UKAD applies a content-predicted displacement and then a group-rational activation. It reuses that activation in the skip gates and in the decoder. We evaluate the displacement and the activation jointly and individually.

3 Deformable Kolmogorov-Arnold Network

3.1 Overall architecture

UKAD keeps the convolutional encoder-decoder of U-Net and replaces the operator at its bottleneck. Where U-Net mixes bottleneck features with a convolutional block and a fixed activation, UKAD applies the Kolmogorov-Arnold Displacement (KAD) block,

$$h = \mathrm{ConvBlock}(h_0), \quad \text{[U-Net]} \qquad\qquad h = \mathrm{KAD}(h_0), \quad \text{[UKAD]} \tag{1}$$

with h_0 the encoder output at the bottleneck. The KAD block predicts a grid from h_0, resamples h_0 on that grid, and applies a group-rational activation ϕ to the resampled features. One gate, predicted from h_0, scales the displacement branch and a parallel context branch. A local branch stays open. The same ϕ drives the skip gates and the decoder, and a boundary head then refines locations where the coarse mask is uncertain. Figure 2 shows the full graph. Section 3.2 presents the group-rational activation, Section 3.3 specifies the content-adaptive grid, Section 3.4 combines them in the KAD block, and Section 3.5 presents the skip gates and hybrid decoder.

For UKAD-B, the five-stage encoder-decoder uses widths $(32, 64, 128, 256, 512)$. UKAD-S and UKAD-L change only these widths. The encoder output h_0 is at $H/16$, and KAD maps it to the decoder input h. Decoder level 4 is the stage nearest the bottleneck, and level 1 is the finest scale.

3.2 Group-rational activation

Following Yang & Wang (2025), we parameterize ϕ as a low-degree rational function, and we share it across groups of channels, in the safe Padé form of Molina et al. (2020). A ratio of low-degree polynomials requires only a few multiplications. When we share the function within a group, the parameter count stays independent of channel width. We split a feature map $u \in \mathbb{R}^{B\times C\times H'\times W'}$ into G groups $u = (u^{(1)}, \ldots, u^{(G)})$ with C divisible by G. Group g uses coefficients $a^{(g)} \in \mathbb{R}^{p+1}$

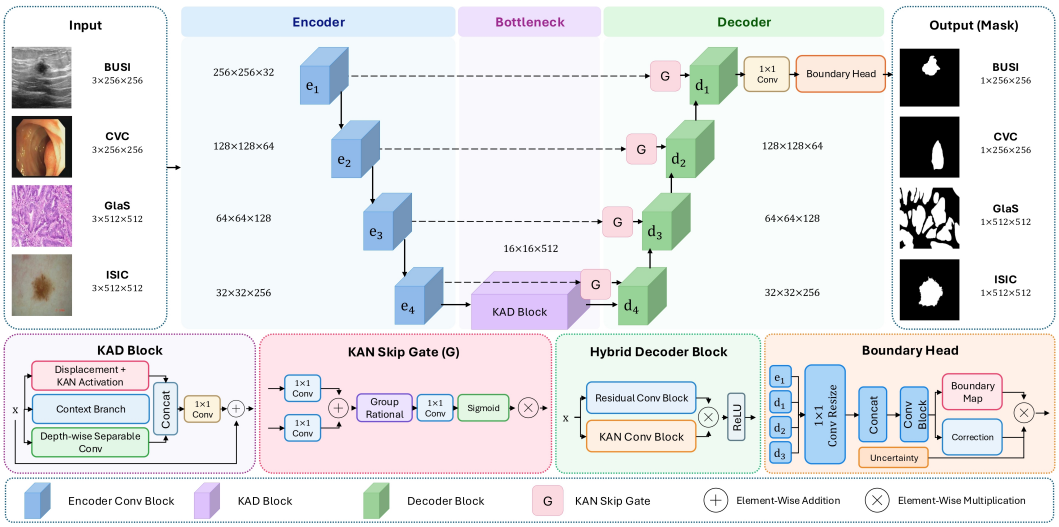

Figure 2: UKAD architecture. KAD resamples bottleneck features before it applies the group-rational activation ϕ. The skip gates and the hybrid decoder use the same ϕ.

and $b^{(g)} \in \mathbb{R}^q$,

$$\phi(u)^{(g)} = \frac{P_g\big(u^{(g)}\big)}{1 + \big|Q_g\big(u^{(g)}\big)\big|}, \qquad P_g(t) = \sum_{k=0}^{p} a_k^{(g)} t^k, \qquad Q_g(t) = \sum_{k=1}^{q} b_k^{(g)} t^k. \tag{2}$$

The denominator is at least one, so ϕ has no poles. We use $p = 3$, $q = 2$, and $G = 8$. We initialize ϕ near the identity. The bottleneck, the skip gates, and the hybrid decoder use this form with separate coefficients. Each instance adds $G(p+q+1) = 48$ parameters, independent of width or spatial size, and UKAD-B has 11 instances (528 parameters). By comparison, a spline KAN layer with $M = 5$ intervals and order $K = 3$ at 512 channels, which also replaces the channel-mixing weights that UKAD keeps in a 1×1 convolution, would learn $512^2(M + K + 3) + 512 \approx 2.9\,\text{M}$ parameters in one layer. A KAN layer in UKAD is a group-shared rational activation followed by a 1×1 convolution, as in Yang & Wang (2025). It does not learn a separate function on each edge, as in Liu et al. (2025).

3.3 Content-adaptive displacement

A 3×3 convolution of the bottleneck map predicts, for each group, raw offsets $(\Delta y, \Delta x)$ and a gate logit ζ. The applied displacement in pixels is

$$\delta_y = r_{\max}\, \sigma(\zeta)\, \tanh(\Delta y), \qquad \delta_x = r_{\max}\, \sigma(\zeta)\, \tanh(\Delta x), \tag{3}$$

with $r_{\max} = 3$ and $\gamma := \sigma(\zeta) \in (0,1)^{B \times G \times H' \times W'}$. At $H/16$, a 256×256 image becomes a 16×16 grid. One bottleneck position then spans 16 image pixels, and $r_{\max} = 3$ permits a displacement of up to 48 image pixels. We bilinearly sample each channel group at its displaced coordinates with reflection padding (Jaderberg et al., 2015; Dai et al., 2017). As in the modulation scalar of Zhu et al. (2019), γ controls adaptive sampling. Here γ scales the whole displacement branch for one channel group and one spatial position, not a single sampling point. Equation (5) uses the same γ to scale the context branch. Offset weights and biases start at zero, and the gate bias starts at 4. Training therefore begins from the fixed-grid bottleneck, and the gate can close the branch when resampling does not help. Appendix E visualizes the resulting fields.

3.4 KAD block

Resampling changes the sampling location. It does not enlarge the receptive field, so KAD pairs the displacement branch with a context branch and with a local depthwise-separable branch that stays open. KAD concatenates the three branches, mixes them, and adds the result to h_0.

The displacement branch is

$$\tilde{h} \;=\; \mathrm{Warp}\big(\mathrm{BN}(h_0);\delta\big), \qquad h_{\mathrm{kad}} \;=\; \phi\big(W_{1\times1}\,\phi(\tilde{h})\big). \tag{4}$$

The context branch is a stack of dilated depthwise convolutions (Yu & Koltun, 2016). The gate γ from Equation (3) scales both adaptive branches and does not scale the local branch. The block output is the residual sum

$$h \;=\; h_0 + s \odot W_{1\times1}\big[\,\bar{\gamma}\odot h_{\mathrm{kad}} \,\|\, \bar{\gamma}\odot h_{\mathrm{ctx}} \,\|\, h_{\mathrm{dsc}}\,\big], \tag{5}$$

where h_{ctx} and h_{dsc} are the context and local branch outputs, $\bar{\gamma}$ expands the group gate γ to all C channels, $\|$ denotes channel concatenation, $\odot$ denotes elementwise multiplication, and $s \in \mathbb{R}^C$ is a learned per-channel scale that we initialize at 1. At UKAD-B width, the displacement and context branches add 0.60 M and 0.56 M parameters, respectively, as Table 21 reports.

3.5 KAN SKIP GATES AND HYBRID DECODER

Each skip uses the attention-gate form of Oktay et al. (2018), with ϕ in place of ReLU. Levels 4 and 3, the two deepest decoder stages, combine a residual convolution with a two-stage depthwise-pointwise KAN convolution. The KAN path starts weakly, with its gate at $\sigma(-2) \approx 0.12$. Levels 2 and 1 are convolutional. Levels 3 and 2 also add an adaptive-context mixer, a residual path across dilations $(1, 2, 4)$ that starts disabled. Appendix A.8 gives both blocks in full.

3.6 UNCERTAINTY-GATED BOUNDARY HEAD

A head on the level-1 decoder map d_1 produces coarse logits z. A multi-scale refinement predicts a boundary map b and a correction. It weights the correction by $\sigma(b)$, applies that correction only near uncertain pixels through $4\sigma(z)(1-\sigma(z))$, and outputs the refined logits z'. The final layer of the head starts at zero. Appendix A.8 gives the full specification.

3.7 TRAINING OBJECTIVE

The objective is

$$\mathcal{L} \;=\; \mathcal{L}_{\mathrm{Dice}}(z', Y) + 0.5\,\mathcal{L}_{\mathrm{BCE}}(z', Y) + 0.05\,\mathcal{L}_{\partial}(b, Y_{\partial}) + \lambda(t)\big(0.1\,\mathcal{L}^{(3)} + 0.2\,\mathcal{L}^{(2)}\big). \tag{6}$$

Here Y_{∂} is a 3×3 morphological boundary band. $\mathcal{L}^{(3)}$ and $\mathcal{L}^{(2)}$ are the losses of the deep-supervision heads on decoder levels 3 and 2. $\lambda(t)$ is 1 for the first 60% of training and then falls linearly to 0. Every dataset and every model size uses these loss weights, which Appendix A lists.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate on four public binary segmentation datasets, one per imaging modality. BUSI (Al-Dhabyani et al., 2020) contains 647 benign and malignant breast ultrasound images. We remove the normal class. CVC-ClinicDB (Bernal et al., 2015) contains 612 colonoscopy frames with polyp masks. Tables and figures abbreviate CVC-ClinicDB as CVC and ISIC 2018 as ISIC. GlaS (Sirinukunwattana et al., 2017) contains 165 hematoxylin-eosin histology images with gland masks. We pool its official train, testA, and testB parts before the split. ISIC 2018 (Codella et al., 2019; Tschandl et al., 2018) contains 2,594 dermoscopy images with lesion masks from Task 1. Input resolution follows U-KAN (Li et al., 2025) and Rolling-UNet (Liu et al., 2024a): 256×256 for BUSI and CVC-ClinicDB, and 512×512 for GlaS and ISIC 2018. The train and validation counts are 517 and 130 for BUSI, 489 and 123 for CVC-ClinicDB, 132 and 33 for GlaS, and 2,075 and 519 for ISIC 2018.

Protocol. We followed the public U-KAN split and training recipe (Li et al., 2025). Appendix A gives that same recipe for the reproduced KAN baselines. For each dataset we drew an 80/20 train/validation split with sklearn. The split seeds are 2981, 6142, and 1187, and the model and training RNG is 1029 for every run. We split CVC-ClinicDB by frame, not by source video. We

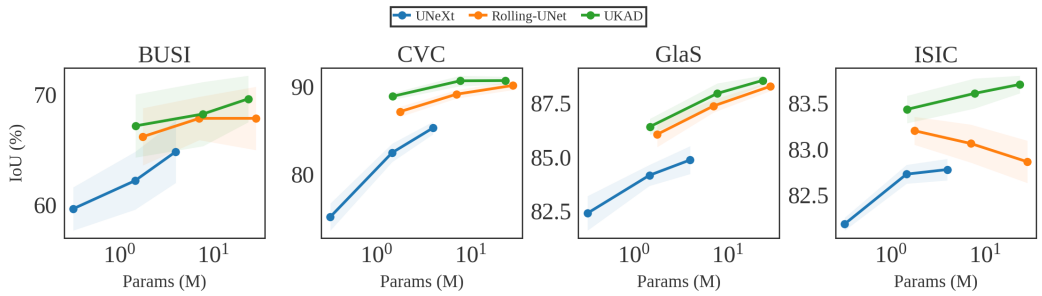

Figure 3: IoU versus parameters across three widths of UNeXt, Rolling-UNet, and UKAD on each dataset.

retrained all fourteen baselines on the same split manifests, with the same preprocessing, schedule, and checkpoint rule. We import no published numbers. We score each method on the validation partition that selects its checkpoint. The tables compare models under one protocol. Every model uses one training seed. We chose the UKAD design on these same splits. Appendix B.4 reports the official ISIC 2018 test set in addition.

We trained every model for 400 epochs with Adam at learning rate 10^{-4} and weight decay 10^{-4}, cosine annealing to 10^{-5}, and batch size 8. The exception is a model whose official recipe sets a grouped learning rate, as U-KAN does (Li et al., 2025). We selected the checkpoint with the highest validation IoU at threshold 0.5, reloaded it, and computed MedPy IoU and Dice as in the released evaluator (Liu et al., 2024b). We average the reported IoU and Dice over batches, and we weight each batch by its size. We computed HD95 on each image with MedPy. An image where exactly one of the prediction and the target is empty scores the image diagonal, and an image where both are empty scores 0. We report the mean and the sample standard deviation over the three data splits. Appendix B adds recall, precision, specificity, Hausdorff distance, and every published width.

Baselines. We compare against fourteen published methods: U-Net (Ronneberger et al., 2015), Attention U-Net (Oktay et al., 2018), UNet++ (Zhou et al., 2018), UNeXt (Valanarasu & Patel, 2022), MedNeXt (Roy et al., 2023), U-Mamba (Ma et al., 2024b), nnU-Net ResEnc (Isensee et al., 2024), Rolling-UNet (Liu et al., 2024a), UKAGNet (Drokin, 2024), U-KAN (Li et al., 2025), U-FunKAN (Penkin & Krylov, 2025), CGLKNet (Hu et al., 2026), UKAN+ (Tian et al., 2026), and AdaKAN (Alzu'bi et al., 2026). Where a public author implementation exists, we use it. Otherwise we follow the published specification in Appendix A.9. We keep each published architecture and change only the training recipe to the one above. Every baseline trains with a Dice plus 0.5 BCE loss, which is also the main term of Equation (6). Published numbers use other splits and other recipes, so we compare only the runs that we trained under this protocol. We release our code, configurations, and split manifests. UNeXt and Rolling-UNet each publish several widths. In the tables, UNeXt means UNeXt-B and Rolling-UNet means Rolling-UNet-L. Figure 3 reports every width.

We trained every model from scratch, in one stage, on 2D binary masks. We therefore exclude pretrained transformer backbones (Chen et al., 2021; Cao et al., 2021), 3D networks (Hatamizadeh et al., 2022), and prompt-conditioned foundation models (Ma et al., 2024a). U-Mamba represents the state-space line (Ruan et al., 2024).

UKAD configuration. The group-rational activation uses $p = 3$, $q = 2$, and $G = 8$ groups. The displacement radius is $r_{\max} = 3$ bottleneck positions. The context branch uses dilations $(1, 3, 5)$. UKAD-S, UKAD-B, and UKAD-L differ only in the five channel widths. UKAD-B is the default. We did not give any dataset its own UKAD hyperparameters. Every dataset, resolution, and width uses the values above and the loss weights of Equation (6).

Cost. We profile one image with THOP for the parameter count and the multiply-accumulate count, and we multiply that count by two to obtain FLOPs. The tables average the four dataset input sizes, two at 256 and two at 512.

Table 1. Segmentation accuracy (IoU/Dice, %, mean ± std over three splits). Parameters and FLOPs average the four input sizes. We bold the best value and underline the second-best value.

			BUSI		CVC		GlaS		ISIC 2018		Avg.
Method	#Param.	FLOPs	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice	IoU
U-Net	31.04	273.69	62.97±3.35	76.25±2.81	88.80±0.16	94.02±0.11	87.59±0.82	93.32±0.42	80.53±0.45	89.01±0.30	79.97
Attention U-Net	34.88	333.08	63.83±2.69	76.91±2.39	89.79±0.57	94.57±0.34	87.78±0.97	93.43±0.52	81.04±0.51	89.34±0.35	80.61
UNet++	9.16	173.10	64.14±2.84	77.14±2.48	89.26±0.43	94.30±0.25	87.10±0.59	93.03±0.32	80.82±0.43	89.20±0.33	80.33
UNeXt	1.47	2.87	62.17±4.53	75.65±3.69	82.42±1.80	90.21±1.10	84.14±0.84	91.32±0.51	82.72±0.18	90.40±0.12	77.86
MedNeXt	10.45	74.15	64.03±4.15	77.24±3.48	86.89±1.50	92.87±0.93	83.35±0.83	90.84±0.48	83.07±0.31	90.58±0.20	79.33
U-Mamba	27.77	121.41	65.52±5.64	78.02±4.51	87.41±2.21	93.15±1.39	80.96±1.26	89.34±0.83	82.74±0.43	90.38±0.31	79.16
nnU-Net ResEnc	122.97	172.56	64.13±5.20	77.03±4.16	80.74±1.66	89.16±1.03	80.44±1.34	89.04±0.84	81.25±0.68	89.46±0.40	76.64
Rolling-UNet	28.31	159.78	67.82±4.97	79.78±3.96	90.07±0.87	94.72±0.50	88.26±0.37	93.70±0.22	82.86±0.40	90.43±0.24	82.25
UKAGNet	9.81	17.67	55.76±3.36	70.53±3.35	85.83±1.35	92.28±0.82	84.58±0.81	91.56±0.42	78.78±0.40	87.93±0.28	76.24
U-KAN	6.36	8.76	67.49±5.12	79.56±4.03	87.28±1.05	93.13±0.62	86.79±0.48	92.86±0.31	83.00±0.47	90.56±0.33	81.14
U-FunKAN	3.78	21.73	62.42±4.98	75.92±3.96	78.44±0.28	87.66±0.17	78.02±1.81	87.53±1.13	81.91±0.71	89.87±0.51	75.20
CGLKNet	32.49	45.41	65.87±5.57	78.52±4.32	86.20±1.23	92.46±0.76	86.70±0.95	92.81±0.54	83.39±0.36	90.77±0.27	80.54
UKAN+	6.72	9.01	66.20±5.72	78.69±4.50	87.17±1.07	93.09±0.63	85.06±1.22	91.87±0.73	83.10±0.25	90.62±0.18	80.38
AdaKAN	14.21	38.54	69.24±5.83	80.72±4.64	89.40±0.99	94.36±0.57	87.09±0.74	93.05±0.43	83.69±0.11	**90.99±0.04**	82.36
UKAD-S	1.49	20.36	67.14±4.95	79.39±3.84	88.83±0.65	94.04±0.39	86.39±0.64	92.62±0.38	83.43±0.26	90.79±0.17	81.45
UKAD-B	7.77	74.46	68.22±5.04	80.16±3.88	90.60±1.02	95.02±0.58	87.93±0.81	93.51±0.45	83.60±0.28	90.90±0.18	82.59
UKAD-L	23.53	217.67	**69.59±3.65**	**81.10±2.54**	**90.63±0.78**	**95.03±0.44**	**88.53±0.39**	**93.88±0.24**	**83.70±0.17**	**90.99±0.13**	**83.11**

Table 2. Per-image HD95 (pixels, lower is better, mean ± std over three splits) for the Table 1 models. An image with exactly one empty mask scores the image diagonal. We bold the best value and underline the second-best value.

Method	BUSI	CVC	GlaS	ISIC 2018
U-Net	36.40±8.30	6.18±2.01	13.98±7.96	20.31±3.63
Attention U-Net	32.68±11.00	5.34±2.43	11.25±4.46	18.12±1.76
UNet++	31.10±7.97	4.02±1.03	11.29±4.31	17.15±3.41
UNeXt	32.46±7.80	11.29±3.57	13.03±4.14	12.71±1.60
MedNeXt	20.42±4.48	5.13±1.46	18.36±3.06	12.36±1.07
U-Mamba	19.33±4.75	4.49±1.52	21.96±2.19	10.75±0.85
nnU-Net ResEnc	21.55±6.22	11.92±1.48	22.97±3.52	12.31±0.94
Rolling-UNet	19.79±5.56	4.38±1.08	13.33±9.97	11.79±0.22
UKAGNet	51.77±3.17	7.60±2.73	17.64±11.75	20.53±3.12
U-KAN	26.47±7.15	11.63±3.39	11.72±2.64	11.44±1.27
U-FunKAN	36.91±8.97	16.39±5.77	23.40±9.11	12.50±1.75
CGLKNet	18.80±4.08	7.87±2.89	16.90±12.23	11.70±1.93
UKAN+	23.99±8.56	5.81±2.03	13.74±4.01	10.31±0.77
AdaKAN	**17.43±5.95**	4.66±2.25	10.46±2.82	**9.98±0.48**
UKAD-S	27.05±7.97	4.31±1.68	12.43±4.16	11.07±1.17
UKAD-B	22.64±5.34	4.23±1.94	11.57±5.09	10.39±0.49
UKAD-L	19.26±5.94	**3.30±1.65**	**8.32±1.80**	9.99±0.90

4.2 SEGMENTATION ACCURACY

IoU and Dice. Table 1 reports IoU and Dice. UKAD-L has the highest IoU on every dataset and the highest Dice on three datasets. It ties AdaKAN on ISIC 2018 Dice at 90.99. AdaKAN is second on IoU for BUSI and for ISIC 2018, UKAD-B is second on CVC-ClinicDB, and Rolling-UNet is second on GlaS. Mean IoU over the four datasets is 83.11 for UKAD-L, 82.59 for UKAD-B, 82.36 for AdaKAN, 82.25 for Rolling-UNet, and 79.97 for U-Net. Mean Dice is 90.25 for UKAD-L, 89.90 for UKAD-B, and 89.78 for AdaKAN. Table 3 is the primary comparison. It pairs models on the 12 matched dataset-split runs and adds a per-image bootstrap for U-KAN, AdaKAN, and Rolling-UNet. Table 20 gives the full matrix. UKAD-L wins 11 of those 12 runs against Rolling-UNet and all 12 against U-KAN ($p_{\mathrm{Holm}} = 0.007$ for both). Per image, the bootstrap gives UKAD-L a gain of 0.79 IoU over AdaKAN ($p < 0.001$). That bootstrap holds each checkpoint fixed and measures variation across images. On the official ISIC 2018 test set, UKAD-B has the highest IoU and Dice, at 77.99 and 87.24. Table 13 places UKAD-L fourth, at 77.61 IoU.

Other KAN U-Nets. UKAGNet reaches 76.24 mean IoU, and U-FunKAN reaches 75.20. Those scores are 3.73 and 4.77 points below the plain U-Net at 79.97. U-KAN reaches 81.14, and UKAD-B reaches 82.59, at 6.36 M and 7.77 M parameters. These models differ in more than the activation. The ordering agrees with Yang & Wang (2025): a KAN layer in a transformer trails the MLP it replaces until the basis, the sharing, and the initialization change together.

Per-image HD95. Table 2 reports HD95 from the same checkpoints as Table 1. Each value is the mean over images, in pixels, and a lower value is a closer boundary. UKAD-L has the lowest HD95 on CVC-ClinicDB and GlaS, at 3.30 and 8.32 pixels, and it is second on ISIC 2018. AdaKAN is lowest on BUSI and on ISIC 2018. Over the four datasets, the mean rank is 1.75 for UKAD-L and 2.75 for AdaKAN. Appendix B.5 gives the batch-level HD95 for the same models.

Table 3. Primary IoU comparison on 12 matched dataset-split runs. Wins is the number of runs in which UKAD has the higher IoU, and ΔIoU is UKAD minus the baseline. The image columns are a paired bootstrap with each checkpoint held fixed. Table 20 gives the full matrix.

	UKAD-L				UKAD-B			
	Primary runs		Supporting images		Primary runs		Supporting images	
Baseline	Wins	p_{Holm}	ΔIoU	p_{boot}	Wins	p_{Holm}	ΔIoU	p_{boot}
Rolling-UNet	11/12	0.007	+0.63	0.004	9/12	0.532	+0.11	0.617
U-KAN	12/12	0.007	+2.53	<0.001	12/12	0.007	+2.01	<0.001
AdaKAN	9/12	0.092	+0.79	<0.001	7/12	0.569	+0.27	0.253

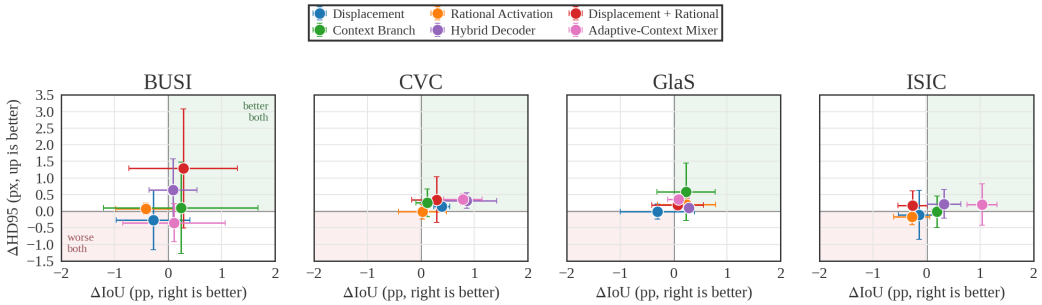

Figure 4: Paired effect of keeping each UKAD-B component. Points are the mean ± standard deviation over three matched splits. Right is a higher IoU, and up is a lower HD95. The upper-right region means keeping the component improves both.

CVC-ClinicDB. UKAD-B has its largest IoU margin on CVC-ClinicDB. The margin is 0.53 over Rolling-UNet, the next best model on that dataset, and 3.32 over U-KAN. On BUSI and on ISIC 2018 the ordering is close.

4.3 Accuracy against model size

Among models near 7.77 M parameters, UKAD-B has a higher mean IoU than U-KAN (6.36 M), UKAN+ (6.72 M), Rolling-UNet-M (7.10 M), UNet++ (9.16 M), and AdaKAN (14.21 M). The gaps to AdaKAN and to U-KAN are 0.23 and 1.45 mean IoU. At the small end, UKAD-S (1.49 M) exceeds the parameter-matched UNeXt (1.47 M) by 3.59 mean IoU. UKAD-S reaches 81.45 mean IoU, against 81.14 for U-KAN, with 77% fewer parameters.

UKAD-B uses 74.46 GFLOPs, against 8.76 GFLOPs for U-KAN and 9.01 GFLOPs for UKAN+, because displacement sampling and the hybrid decoder run at spatial resolution. UKAD-B uses fewer FLOPs than U-Net, Attention U-Net, UNet++, and Rolling-UNet, and $8.5\times$ more than the tokenized KAN bottlenecks. On one A100, UKAD-B takes 11.31 ms against 7.06 ms for U-KAN at 256×256, and 39.62 ms against 22.93 ms at 512×512. Table 9 reports peak memory of 141 and 449 MB for UKAD-B, against 59 and 135 MB for U-KAN. UKAD suits a GPU or offline setting that can exchange throughput and memory for overlap and HD95.

Figure 5 shows one validation case from each dataset on split 2981. The columns compare Rolling-UNet, U-KAN, AdaKAN, and UKAD-L with the image and the ground truth. Green marks agreement with the ground truth, blue marks missed foreground, and red marks a false positive.

4.4 Ablation study and analysis

Every arm uses the three split manifests of UKAD-B. Figure 4 plots the paired effect of keeping each of six components. Positive values favor keeping the component. Removing the displacement and the group-rational activation together lowers mean IoU from 82.59 to 82.49. The same removal raises batch-level HD95 on all four datasets, by 1.29, 0.35, 0.19, and 0.17 pixels.

The displacement arm, the group-rational arm, the joint-removal arm, and the default form a 2×2 comparison. When we keep the group-rational activation, displacement adds 0.39 IoU on CVC-

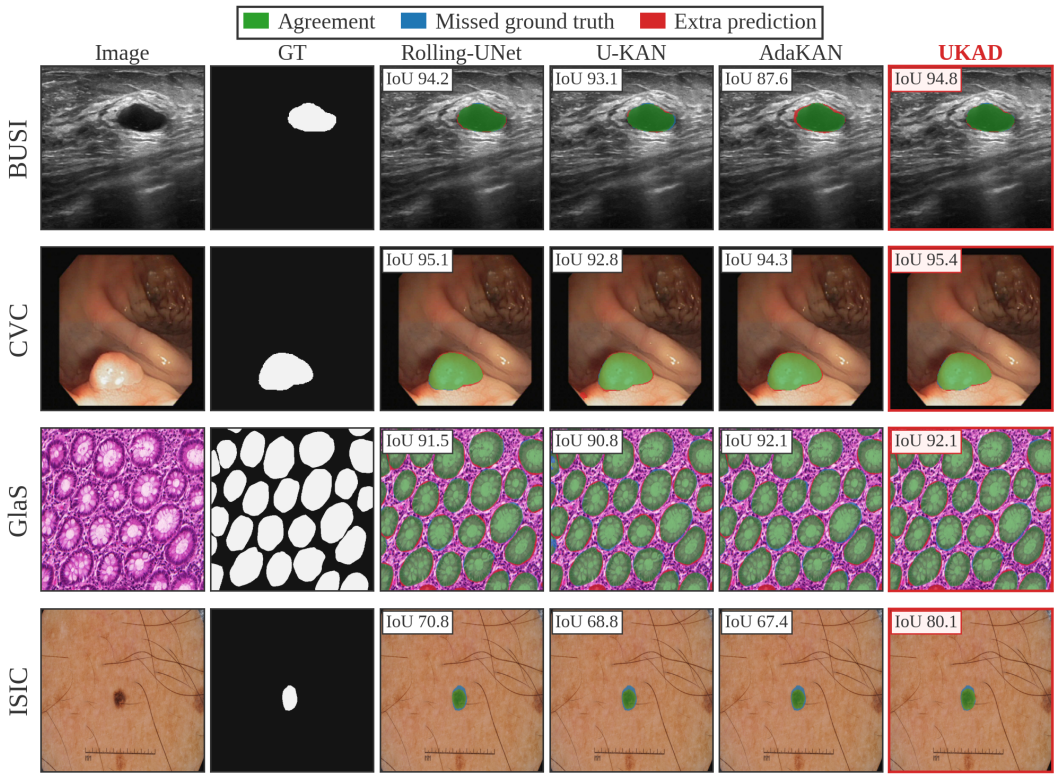

Figure 5: One validation case per dataset, from split 2981. Green marks agreement with the ground truth, blue marks missed foreground, and red marks a false positive. Each panel reports IoU (%).

ClinicDB. The interaction is the default, minus the arm without displacement, minus the arm without the group-rational activation, plus the arm without both. That term is $+0.12$ IoU on CVC-ClinicDB. The adaptive-context mixer is positive on CVC-ClinicDB and on ISIC 2018, by 0.78 and 1.04 IoU. Removing the KAN skip gates lowers IoU on all four datasets and lowers mean IoU from 82.59 to 82.30, with 1.13 M fewer parameters.

5 Conclusion

UKAD is a deformable KAN encoder-decoder for medical image segmentation. Its KAD bottleneck predicts a content-adaptive grid, resamples the bottleneck features on that grid, and then applies a group-rational activation. We share that activation within each channel group, so its parameter count does not grow with network width, and the skip gates and the hybrid decoder reuse it. UKAD trains in one stage, without pretraining or distillation, and with one hyperparameter setting across datasets, resolutions, and widths. In the comparison with fourteen U-Net, MLP, state-space, and KAN baselines, UKAD-L has the highest mean IoU and Dice. UKAD-B reaches 82.59 mean IoU with 45% fewer parameters than AdaKAN. Together, these results show that, for KAN segmentation, where the network evaluates the learned function can matter as much as the basis that defines it.

REFERENCES

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

Dalia Alzu'bi, Deep Bhattacharyya, Ali Ayub, and Abdessamad Ben Hamza. AdaKAN: A dual-branch adaptive Kolmogorov-Arnold network for medical image segmentation. *Pattern Recognition Letters*, 208:27–33, 2026. doi: 10.1016/j.patrec.2026.07.014.

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.

Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-Unet: Unet-like pure transformer for medical image segmentation. *arXiv preprint arXiv:2105.05537*, 2021.

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. *arXiv preprint arXiv:2102.04306*, 2021.

Noel Codella, Veronica Rotemberg, Philipp Tschandl, M. Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, Harald Kittler, and Allan Halpern. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (ISIC). *arXiv preprint arXiv:1902.03368*, 2019.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 764–773, 2017.

Ivan Drokin. Kolmogorov-Arnold convolutions: Design principles and empirical studies. *arXiv preprint arXiv:2407.01092*, 2024.

Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger R. Roth, and Daguang Xu. UNETR: Transformers for 3D medical image segmentation. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022.

Cong Hu, Zhengpeng Li, Jun Hu, Kunyang Wu, Bin Yang, Jiansheng Wu, and Yuanyuan Zhang. Medical image segmentation model for complex boundary features: Large kernel deformable convolution and gated feature preservation. *Expert Systems with Applications*, 296:128801, 2026. doi: 10.1016/j.eswa.2025.128801.

Fabian Isensee, Paul F. Jäger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein. nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 2021.

Fabian Isensee, Tassilo Wald, Constantin Ulrich, Michael Baumgartner, Saikat Roy, Klaus Maier-Hein, and Paul F. Jäger. nnU-Net revisited: A call for rigorous validation in 3D medical image segmentation. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2024.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015.

Andrey N. Kolmogorov. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Doklady Akademii Nauk SSSR*, 114 (5):953–956, 1957.

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*, volume 39, pp. 4652–4660, 2025.

Xiangtai Li, Ansheng You, Zhen Zhu, Houlong Zhao, Maoke Yang, Kuiyuan Yang, Shaohua Tan, and Yunhai Tong. Semantic flow for fast and accurate scene parsing. In *European Conference on Computer Vision (ECCV)*, pp. 775–793, 2020.

Yutong Liu, Haijiang Zhu, Mengting Liu, Huaiyuan Yu, Zihan Chen, and Jie Gao. Rolling-Unet: Revitalizing MLP's ability to efficiently extract long-distance dependencies for medical image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 3819–3827, 2024a. doi: 10.1609/aaai.v38i4.28173.

Yutong Liu, Haijiang Zhu, Mengting Liu, Huaiyuan Yu, Zihan Chen, and Jie Gao. Rolling-Unet: Official PyTorch implementation. `https://github.com/Jiaoyang45/Rolling-Unet`, 2024b. Evaluation script `val.py` and `metrics.py`.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. KAN: Kolmogorov-Arnold networks. In *International Conference on Learning Representations (ICLR)*, 2025.

Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15:654, 2024a.

Jun Ma, Feifei Li, and Bo Wang. U-Mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024b.

Alejandro Molina, Patrick Schramowski, and Kristian Kersting. Padé activation units: End-to-end learning of flexible activation functions in deep networks. In *International Conference on Learning Representations (ICLR)*, 2020.

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. *arXiv preprint arXiv:1804.03999*, 2018.

Maksim Penkin and Andrey Krylov. FunKAN: Functional Kolmogorov-Arnold network for medical image enhancement and segmentation. *arXiv preprint arXiv:2509.13508*, 2025.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, 2015.

Saikat Roy, Gregor Koehler, Constantin Ulrich, Michael Baumgartner, Jens Petersen, Fabian Isensee, Paul F. Jäger, and Klaus H. Maier-Hein. MedNeXt: Transformer-driven scaling of ConvNets for medical image segmentation. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2023.

Jiacheng Ruan, Jincheng Li, and Suncheng Xiang. VM-UNet: Vision Mamba UNet for medical image segmentation. *arXiv preprint arXiv:2402.02491*, 2024.

Korsuk Sirinukunwattana, Josien P. W. Pluim, Hao Chen, Xiaojuan Qi, Pheng-Ann Heng, Yun Bo Guo, Bogdan J. Matuszewski, Elia Bruni, Urko Sanchez, Anton Böhm, Olaf Ronneberger, et al. 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.

Lele Tian, Xinglong Wu, Yapeng Wang, Xiayu Xu, Tao Tan, Xu Yang, Zefeng Li, Mini Han Wang, Jucheng Song, Sio-Kei Im, Zhantu Liang, Bailin Chen, and Xuming Tong. UKAN+: Enhancing medical image segmentation via fusion attention with Kolmogorov-Arnold networks. *Biomedical Signal Processing and Control*, 119:109789, 2026. doi: 10.1016/j.bspc.2026.109789.

Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5: 180161, 2018. doi: 10.1038/sdata.2018.161.

Jeya Maria Jose Valanarasu and Vishal M. Patel. UNeXt: MLP-based rapid medical image segmentation network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 23–33, 2022.

Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, Xiaogang Wang, and Yu Qiao. InternImage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14408–14419, 2023.

Xiaohong Wu, Sheng Ji, Jie Tao, and Yonggen Gu. U-GRKAN: An efficient and interpretable architecture for medical image segmentation. *Journal of Imaging Informatics in Medicine*, 39(4): 3532–3547, 2025.

Xingyi Yang and Xinchao Wang. Kolmogorov-Arnold transformer. In *International Conference on Learning Representations (ICLR)*, 2025.

Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations (ICLR)*, 2016.

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 (DLMIA)*, pp. 3–11, 2018.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable ConvNets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9308–9316, 2019.

REPRODUCIBILITY STATEMENT

Every result in this paper comes from one pipeline. Appendix A.3 gives the split protocol, the three split seeds, and the fixed model seed. Appendix A.4 gives the preprocessing and augmentation, including the double division by 255 that the released U-KAN dataset classes apply. Appendix A.5 gives the training recipe and the single baseline whose official recipe departs from it. Appendix A.6 and Appendix A.7 give the loss and every UKAD hyperparameter, and Appendix A.9 pins each baseline to the public source we built it from. Appendix A.2 defines every metric and names the two IoU implementations the pipeline uses. The pipeline writes the split manifests to disk before training and reads them back at evaluation, so the reported runs and any rerun see the same images in the same order.

ETHICS STATEMENT

This work uses four public, de-identified medical image datasets (Al-Dhabyani et al., 2020; Bernal et al., 2015; Sirinukunwattana et al., 2017; Codella et al., 2019; Tschandl et al., 2018), each released for research by its original providers and used here under its own terms. We collected no new patient data and attempted no re-identification. The results use the protocol-matched validation splits of Section 4.1.

THE USE OF LARGE LANGUAGE MODELS

We used a large language model as a general-purpose assistant. Its use fell into three areas. The first was editing prose that we drafted, for grammar, concision, and consistency of terminology. The second was routine software engineering, including writing and refactoring plotting, evaluation, and table-generation scripts. The third was checking the manuscript against the tables, and this check found several cross-reference and arithmetic errors. The research idea, the architecture, the experimental design, and every interpretation of the results are our own. The model did not generate research ideas and produced none of the reported numbers, all of which come from the training and evaluation runs described above. We checked all text and code it contributed and take full responsibility for the content of this paper.

APPENDIX OVERVIEW

The main paper makes four claims. This appendix gives the detail needed to check each one. Appendix A documents the split, the preprocessing, the training, the metrics, and the baseline sources. The comparison uses one shared protocol. Appendix B places the overlap results next to parameter count, FLOPs, latency, and memory. Appendix C tests which of the headline differences are significant. Appendix D gives every ablation arm and every metric. Appendix E shows representative masks and the learned fields, maps, and gates. Appendix F reads those results.

A PROTOCOL, IMPLEMENTATION, AND REPRODUCIBILITY DETAILS

This section specifies the protocol behind every reported table. It defines the notation and the metrics, then the data splits, the preprocessing, the training, the UKAD objective, the UKAD hyperparameters, and the source of each reproduced baseline. Together, these details set the comparison as a matched reproduction of the released recipes.

A.1 NOTATION AND TERMS

Table 4 lists the symbols and names that the paper uses.

Table 4. Notation.

Symbol or Name	Meaning
x	Input image, shape $B \times 3 \times H \times W$
B, C, H, W	Batch size, channels, height, width
e_i, d_i	Encoder and decoder feature maps at level $i \in \{1, 2, 3, 4\}$
ϕ	Group-rational activation, Eq. (2)
G	Number of channel groups; default $G = 8$
p, q	Numerator and denominator degrees of ϕ; default $p = 3$, $q = 2$
a_k, b_k	Learned rational coefficients, shared inside a group
$r_{\max}$	Maximum Displacement in bottleneck pixels; default 3
$(\Delta y, \Delta x), \zeta$	Raw offset and gate logit predicted by a 3×3 Convolution
(δ_y, δ_x)	Applied Displacement, Eq. (3)
$\gamma, \bar{\gamma}$	Group gate $\sigma(\zeta)$ and its expansion to C channels
u	A feature map passed to ϕ or to Warp
(ξ_j, η_i)	Identity sampling coordinates on the bottleneck grid
$\hat{d}_\ell$	Upsampled decoder map entering the skip gate at level ℓ
Warp	Bilinear sampling at the displaced grid
$D_{k,d}$	Depthwise Convolution, kernel k, dilation d
s	Residual per-channel scale in the KAD block
$F_{\mathrm{KAN}}, F_{\mathrm{conv}}$	KAN Convolution and residual Convolution
σ	Sigmoid
α_ℓ	Skip-connection attention mask at level ℓ
θ	Per-channel logit that mixes the KAN branch of the hybrid decoder
z, z'	Coarse and refined logits
b, c	Boundary map and correction from the boundary head
$Y_\partial, \lambda(t)$	Morphological boundary target and deep-supervision anneal
KAD	Kolmogorov-Arnold Displacement bottleneck block
DSC	Depthwise-separable Convolution (local branch)
UKAD	U-shaped network with a KAD block
UKAD-S/B/L	Width variants of the same architecture
IoU	Intersection over union (Jaccard index)
Dice	Dice coefficient (set overlap)
HD, HD95	Hausdorff distance and its 95th-percentile form

Two names in Table 4 mean something else in other papers, so we fix the meaning here. Dice in the last row is the set-overlap score. Appendix A.6 defines the training Dice loss, which is a different quantity. A hard gate is a straight-through estimator. The forward pass thresholds at 0.5, and the backward pass uses the sigmoid. Appendix D.1 names the change in each ablation arm.

A.2 METRIC DEFINITIONS

Let $\hat{Y}, Y \in \{0,1\}^{H\times W}$ be a binarized prediction and a ground-truth mask. Write $\mathrm{TP} = \sum_{ij} \hat{Y}_{ij} Y_{ij}$, $\mathrm{FP} = \sum_{ij} \hat{Y}_{ij}(1 - Y_{ij})$, $\mathrm{FN} = \sum_{ij}(1 - \hat{Y}_{ij}) Y_{ij}$, and $\mathrm{TN} = \sum_{ij}(1 - \hat{Y}_{ij})(1 - Y_{ij})$. We binarize the network output by $\hat{Y} = \mathbf{1}[\sigma(z) > 0.5]$.

Overlap. Intersection-over-union (IoU, Jaccard) and Dice are

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \qquad \mathrm{Dice} = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \tag{7}$$

They are related by $\mathrm{Dice} = 2\,\mathrm{IoU}/(\mathrm{IoU} + 1)$ when we compute both on the same sets.

Pixel classification. Recall (sensitivity), precision, and specificity are

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \qquad \mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \qquad \mathrm{Specificity} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}. \tag{8}$$

Boundary. Let $S_{\hat{Y}}$ and S_Y be the sets of foreground-boundary pixels. The directed distance from A to B is the largest gap from a point in A to the nearest point in B. HD is the larger of the two directed distances. HD95 is the 95th percentile of those same distances. The unit is pixels, at 256 or at 512. A lower value is a closer boundary. The appendix boundary tables follow the released evaluator (Liu et al., 2024b).[1] That code sends one thresholded batch to MedPy, not one image at a time. Table 2 computes HD95 on each image instead. If the prediction is empty and the target is not, or the reverse, that image scores the diagonal: 362 pixels at 256 and 724 pixels at 512. If both are empty, the image scores 0. Table 15 counts the images that score the diagonal.

Two IoU implementations. Checkpoint selection uses one IoU, and the tables use another. The training IoU uses threshold 0.5, sums intersection and union over the whole batch, and adds smoothing 10^{-5},

$$\mathrm{IoU}_{\mathrm{train}} = \frac{|\hat{Y} \cap Y| + 10^{-5}}{|\hat{Y} \cup Y| + 10^{-5}}.$$

MedPy then computes binary Jaccard, Dice, HD, HD95, recall, precision, and specificity on each binarized batch. The tables average those batch values with weights equal to the batch size. We compute Hausdorff distances only at evaluation, not during training. No MedPy call failed in any reported run, so no fallback value enters the tables.

Aggregation. The standard deviation in every table is the sample form over the three splits,

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}, \qquad n = 3.$$

We report IoU, Dice, recall, precision, and specificity as percentages. HD and HD95 stay in pixels.

A.3 DATASETS AND SPLITS

Table 5 lists the four datasets. The split unit is the image for BUSI, GlaS, and ISIC 2018, and the frame for CVC-ClinicDB. Following the public U-KAN recipe, we apply sklearn's unstratified `train_test_split` with `test_size=0.2` and `random_state` equal to the split seed. The split seeds are 2981, 6142, and 1187, and we draw each split independently. The model and training RNG is 1029 on every split. The remaining 20% is the validation set. We use it both to select the checkpoint and to report the metrics.

The public U-KAN code and the reproduced KAN baselines use this two-way train and validation protocol. We write the manifests before training, and every model reads those same manifests. For CVC-ClinicDB, the public protocol splits frames rather than source video sequences.

[1] `https://github.com/CUHK-AIM-Group/U-KAN` and `https://github.com/MaksimPenkin/MedicalKAN`

Table 5. Dataset counts for the protocol-matched 80/20 train/validation split. BUSI excludes normal images, and GlaS pools its official subsets before splitting.

Dataset	Split Unit	Pool	Train	Val.
BUSI	Image	647	517	130
CVC-ClinicDB	Frame	612	489	123
GlaS	Image	165	132	33
ISIC 2018	Image	2,594	2,075	519

A.4 PREPROCESSING AND AUGMENTATION

Training augmentation follows the U-KAN, UNeXt, and Rolling-UNet pipeline (Li et al., 2025; Valanarasu & Patel, 2022; Liu et al., 2024a). In order, it applies `RandomRotate90` with probability 0.5, then a random flip with probability 0.5. The flip is horizontal, vertical, or both. It then resizes BUSI and CVC-ClinicDB to 256×256, and GlaS and ISIC 2018 to 512×512. The last step is ImageNet normalization, with mean $(0.485, 0.456, 0.406)$ and standard deviation $(0.229, 0.224, 0.225)$. Augmentation runs before the resize. Validation uses only the resize and the normalization.

The dataset classes released with that pipeline divide the already-normalized image by 255 a second time. The effective map is $((x/255 - \text{mean})/\text{std})/255$. We keep that scaling because the released baseline code applies it. `albumentations` version 2 and later removed `A.Flip`. We implement the three flip combinations as a `OneOf`.

A.5 TRAINING AND EVALUATION

Section 4.1 gives the shared recipe. This subsection records the one exception and the steps of the evaluation pass.

U-KAN is the only baseline whose official recipe uses a grouped learning rate (Li et al., 2025). Parameters whose names contain both `layer` and `fc` use learning rate 10^{-2}. The remaining parameters use 10^{-4}. We keep that grouping for U-KAN. Every other model, including UKAD, uses the common Adam recipe.

Checkpoint selection uses the training IoU defined in Appendix A.2, not the MedPy IoU. Evaluation reloads the selected checkpoint and runs the validation loader once. It writes the MedPy metrics of each split to one JSON file per experiment. We build the tables from these files. We trained every model, baselines and UKAD alike, on one NVIDIA A100-SXM4-80GB. Appendix B.2 uses that same GPU for the latency and memory measurements.

A.6 UKAD LOSS

The main loss is Dice plus binary cross-entropy, and the cross-entropy weight is 0.5. The Dice loss takes logits and uses smoothing 10^{-5}. When we enable the boundary head, we add a morphological-boundary term with weight 0.05. The boundary target is the mask after a 3×3 dilation, minus the mask after a 3×3 erosion. The boundary term is the mean of a boundary Dice loss and a weighted boundary BCE loss. The positive weight is $\text{clip}(N_{\text{neg}}/N_{\text{pos}}, 1, 20)$.

Deep supervision adds heads on decoder levels 3 and 2, with weights 0.1 and 0.2. We compare each head with the mask at the input resolution. Those weights stay at their initial values for the first 60% of training and then fall linearly to zero. The total training objective is

$$\mathcal{L} = \mathcal{L}_{\text{Dice}} + 0.5\,\mathcal{L}_{\text{BCE}} + 0.05\,\mathcal{L}_{\text{boundary}} + \lambda(t)\,(0.1\,\mathcal{L}_{\text{aux3}} + 0.2\,\mathcal{L}_{\text{aux2}}),$$

where $\lambda(t) = 1$ for the first 60% of epochs and then falls linearly to 0.

A.7 UKAD HYPERPARAMETERS

Table 6 lists the settings that are specific to UKAD. The optimizer, the schedule, the epoch count, and the batch size are the shared settings in Section 4.1.

Table 6. UKAD-specific hyperparameters, shared by every dataset and every width.

Setting	Value
Encoder/decoder widths (S)	$(16, 32, 64, 112, 192)$
Encoder/decoder widths (B)	$(32, 64, 128, 256, 512)$
Encoder/decoder widths (L)	$(56, 112, 224, 448, 896)$
Groups G	8
Rational degrees (p, q)	$(3, 2)$
Max Displacement $r_{\max}$	3 pixels at bottleneck resolution
Context dilations	$(1, 3, 5)$
Decoder hybrid levels	4 and 3
Hybrid decoder gate init θ	-2
Displacement / gate Convolution	3×3, zero weights
Gate bias at init	4 ($\sigma(4) \approx 0.98$)
Deep-supervision loss weights	$(0.1, 0.2)$
Deep-supervision anneal start	0.6 of training
Boundary loss weight	0.05
Boundary kernel	3×3
Refinement channels	16

A.8 Hybrid decoder, adaptive-context mixer, and boundary head

The KAN convolution of the hybrid decoder has two depthwise-pointwise stages, each followed by batch normalization and ϕ,

$$F_{\mathrm{KAN}}(x) = \phi\big(\mathrm{BN}(W_2 D_2 u)\big), \qquad u = \phi\big(\mathrm{BN}(W_1 D_1 x)\big),$$

where D_1 and D_2 are 3×3 depthwise convolutions and W_1 and W_2 are 1×1 convolutions. Decoder levels 4 and 3, the two deepest decoder stages, add F_{KAN} to a residual convolution block F_{res} through a per-channel gate,

$$d = \mathrm{ReLU}\big(F_{\mathrm{res}}(x) + \sigma(\theta) \odot F_{\mathrm{KAN}}(x)\big),$$

and we initialize $\theta \in \mathbb{R}^C$ at -2. F_{res} is two 3×3 convolutions with batch normalization and a ReLU between them, plus a 1×1 shortcut when the width changes.

The adaptive-context mixer on decoder levels 3 and 2 applies three 3×3 depthwise convolutions $D^{(r)}$ with dilations $r \in \{1, 2, 4\}$ to $\hat{d} = \mathrm{BN}(d)$ and weights them with a router,

$$\alpha = \mathrm{softmax}\big(W_b\, \mathrm{ReLU}(W_a\, \mathrm{GAP}(\hat{d}))\big), \qquad d' = d + s_{\mathrm{mix}} \odot \mathrm{BN}\Big(W_f \sum_r \alpha_r D^{(r)} \hat{d}\Big),$$

where GAP is global average pooling. W_a maps C channels to $\max(C/8, 8)$ channels, W_b maps that vector to the three dilations, and W_f is a 1×1 convolution. The scale $s_{\mathrm{mix}} \in \mathbb{R}^C$ starts at zero, so the mixer adds nothing at initialization. Training can open it.

The boundary head takes the coarse logits z and four feature maps: the level-1 encoder map e_1 and the decoder maps d_1, d_2, and d_3. A 1×1 convolution, followed by batch normalization and ReLU, projects each map to 16 channels. Bilinear interpolation then resizes each map to the resolution of z. A separable residual block fuses the concatenated 64 channels into a map f with 32 channels. This block is a 1×1 convolution, a 3×3 depthwise convolution, and a 1×1 convolution, each with batch normalization, plus a 1×1 shortcut. Two small heads read f. The boundary head H_b is a 3×3 convolution to 16 channels, batch normalization, ReLU, and a 1×1 convolution to one channel. Its output is the boundary logits b. The correction head H_c is a 3×3 depthwise convolution, batch normalization, ReLU, and a 1×1 convolution to one channel. Its output is the correction c. The refined logits are

$$z' = z + 4\,\sigma(\bar{z})\big(1 - \sigma(\bar{z})\big) \odot \sigma(b) \odot c,$$

where $\bar{z}$ is z after we stop its gradient. The last convolution of H_c starts at zero, so $z' = z$ at initialization. Appendix A.6 supervises b with the morphological boundary of the mask.

A.9 Baseline implementations

Table 7 records the source and the retained setting for every baseline. Every baseline uses the shared splits, preprocessing, loss, schedule, and checkpoint rule from Section 4.1. The grouped learning rate of U-KAN is the only training setting we keep from an official recipe.

Table 7. Baseline fidelity. We retain each architecture from the listed author source or published specification. All baselines use 256-pixel BUSI and CVC inputs and 512-pixel GlaS and ISIC inputs. The shared protocol changes all training settings except U-KAN's grouped learning rate.

Method	Public author source	Retained setting
U-Net	`github.com/milesial/pytorch-unet`	Original U-Net
Attention U-Net	`github.com/ozan-oktay/Attention-Gated-Networks`	Attention-gated U-Net
UNet++	`github.com/MrGiovanni/UNetPlusPlus`	$[32, 64, 128, 256, 512]$ grid
UNeXt	`github.com/jeya-maria-jose/unext-pytorch`	Published UNeXt-B
MedNeXt	`github.com/MIC-DKFZ/MedNeXt`	MedNeXt-B 2D, kernel 3
U-Mamba	`github.com/bowang-lab/U-Mamba`	2D plan $(32, 64, 128, 256, 512, 512)$
nnU-Net ResEnc	`github.com/MIC-DKFZ/nnUNet`	ResEnc-M 2D plan, 7 stages at 256 and 8 at 512
Rolling-UNet	`github.com/Jiaoyang45/Rolling-Unet`	Published Rolling-UNet-L
UKAGNet	`github.com/IvanDrokin/torch-conv-kan`	Plain non-bottleneck model
U-KAN	`github.com/CUHK-AIM-Group/U-KAN`	Published architecture and grouped learning rate
U-FunKAN	`github.com/MaksimPenkin/MedicalKAN`	Published architecture and resolution-matched ϕ
CGLKNet	`github.com/HUCONGG/-CGLKNet`	Published CGLKNet
UKAN+	Published specification (Tian et al., 2026)	Published UKAN+
AdaKAN	`github.com/bhattacharyyadeep/KAT-Project`	Published AdaKAN

B Extended benchmark results and cost evidence

The main paper reports the compact comparison. This section adds the remaining metrics, every published width, and the compute measurements that sit next to an overlap result.

B.1 Cost and four-dataset summary

The value returned by THOP `profile()` is a multiply-accumulate count. We store it as `macs` and report FLOPs $= 2 \times$ MACs. FLOPs scale with spatial size. We use a four-dataset average that is 2.5 times the 256×256 count, because two datasets use 256 and two use 512. Table 8 lists both columns. A comparison with a profile that uses only 256×256 should read the 256 column.

U-FunKAN uses a 16×16 bottleneck operator at 256 input and a 32×32 operator at 512 input. Its parameter count is 3.64 M at 256 and 3.93 M at 512. Table 1 reports the average of the two counts, 3.78 M.

B.2 Inference latency and GPU memory

A FLOP count leaves out memory-bound operations such as bilinear feature resampling. Table 9 measures the full forward pass, including resampling. The measurement excludes data loading, preprocessing, and host-to-device transfer. Section 4.3 compares the UKAD-B latencies with U-KAN. Peak allocated GPU memory is 141 MB for UKAD-B at 256 and 449 MB at 512, against 59 MB and 135 MB for U-KAN. UKAD-B is slower than the tokenized U-KAN bottleneck and faster than Rolling-UNet-L and CGLKNet, at both resolutions. Table 9 adds latency and memory to the FLOP counts above. It does not replace those counts.

B.3 Dice and HD95 of the width variants

Figure 3 and Table 10 give the IoU of the nine width variants. Tables 11 and 12 give Dice and batch-level HD95 for the same rows.

Dice follows the IoU ordering at every width. Batch-level HD95 does not. In Table 12 a lower HD95 is a closer boundary. For Rolling-UNet-S, M, and L, BUSI HD95 is 8.15, 8.50, and 8.76, and ISIC

Table 8. Cost and four-dataset summary. FLOPs at 256 use one image, and avg. FLOPs average 256 and 512 inputs. We bold the best summary value and underline the second-best summary value.

Method	Params (M)	FLOPs 256 (G)	FLOPs avg. (G)	Mean IoU	Mean Dice
U-Net	31.04	109.48	273.69	79.97	88.15
Attention U-Net	34.88	133.23	333.08	80.61	88.56
UNet++	9.16	69.24	173.10	80.33	88.41
UNeXt-S	0.32	0.23	0.56	74.82	84.87
UNeXt-B	1.47	1.15	2.87	77.86	86.89
UNeXt-L	3.99	2.84	7.11	79.41	87.94
MedNeXt	10.45	29.51	74.15	79.33	87.88
U-Mamba	27.77	48.56	121.41	79.16	87.72
nnU-Net ResEnc	105.92 / 140.02	68.69	172.56	76.64	86.17
Rolling-UNet-S	1.78	4.20	10.51	80.62	88.69
Rolling-UNet-M	7.10	16.26	40.64	81.82	89.42
Rolling-UNet-L	28.31	63.91	159.78	82.25	89.66
UKAGNet	9.81	7.07	17.67	76.24	85.58
U-KAN	6.36	3.51	8.76	81.14	89.03
U-FunKAN	3.64 / 3.93	8.69	21.73	75.20	85.24
CGLKNet	32.49	18.19	45.41	80.54	88.64
UKAN+	6.72	3.60	9.01	80.38	88.57
AdaKAN	14.21	15.42	38.54	82.36	89.78
UKAD-S	1.49	8.14	20.36	81.45	89.21
UKAD-B	7.77	29.78	74.46	<u>82.59</u>	<u>89.90</u>
UKAD-L	23.53	87.07	217.67	**83.11**	**90.25**

Table 9. Batch-one FP32 inference on one A100-SXM4-80GB. Latency is the mean ± std over 200 timed passes after 50 warm-ups. Peak memory includes weights and inference allocations.

Method	Input 256×256		Input 512×512	
	Latency (ms)	Peak memory (MB)	Latency (ms)	Peak memory (MB)
U-Net	6.88±1.38	256	22.18±8.24	664
Attention U-Net	7.53±0.99	278	36.23±4.09	705
UNet++	11.34±0.78	177	37.44±4.22	601
UNeXt-S	1.00±0.45	35	1.28±0.02	34
UNeXt-B	1.07±0.02	26	1.68±0.03	61
UNeXt-L	1.21±0.02	46	5.84±0.99	115
MedNeXt	6.58±0.61	124	29.10±2.84	375
U-Mamba	11.46±1.01	184	35.39±3.90	392
nnU-Net ResEnc	11.86±2.06	458	25.77±2.64	746
Rolling-UNet-S	17.30±0.84	39	24.09±1.90	112
Rolling-UNet-M	34.53±15.87	80	48.26±1.95	211
Rolling-UNet-L	71.65±13.42	199	104.33±13.41	447
UKAGNet	7.92±1.64	191	41.76±5.22	626
U-KAN	7.06±0.92	59	22.93±4.63	135
U-FunKAN	10.77±0.89	56	12.22±1.34	160
CGLKNet	46.64±1.50	198	71.19±4.07	390
UKAN+	16.62±1.18	61	34.51±4.54	136
AdaKAN	7.61±1.75	88	21.79±0.98	163
UKAD-S	8.00±1.85	93	26.91±3.95	332
UKAD-B	11.31±1.16	141	39.62±4.24	449
UKAD-L	16.71±1.52	250	62.05±6.71	702

2018 HD95 is 2.22, 2.29, and 2.99. Rolling-UNet-L is the farthest boundary of the three on those two datasets. On IoU, that same width is highest on CVC-ClinicDB at 90.07 and on GlaS at 88.26, and it ties Rolling-UNet-M on BUSI at 67.82. UKAD-L has BUSI HD95 7.62 and GlaS HD95 1.32, against 6.90 and 1.29 for UKAD-B, so its boundary is farther on those two datasets. Table 8 gives mean IoU and mean Dice for these nine models and for every other model. We do not average HD95 across datasets, because HD95 is in pixels and the datasets use two resolutions.

We take the channel lists of UNeXt-B, UNeXt-L, and all three Rolling-UNet widths from their official configurations. Valanarasu & Patel (2022) report a parameter count for UNeXt-S but not its channel list. We widened the fourth stage of the S configuration until the count matched the published 0.32 M. That row is a reconstruction.

Table 10. IoU (%, mean ± std over three splits) across published widths. We bold the best value and underline the second-best value.

Method	BUSI	CVC	GlaS	ISIC 2018
UNeXt-S	59.61±3.39	75.10±2.66	82.39±1.38	82.18±0.12
UNeXt-B	62.17±4.53	82.42±1.80	84.14±0.84	82.72±0.18
UNeXt-L	64.80±4.88	85.22±1.24	84.85±1.11	82.77±0.20
Rolling-UNet-S	66.16±4.49	87.09±1.00	86.04±1.01	83.20±0.26
Rolling-UNet-M	67.82±3.39	89.07±0.82	87.35±0.51	83.06±0.36
Rolling-UNet-L	67.82±4.97	90.07±0.87	<u>88.26±0.37</u>	82.86±0.40
UKAD-S	67.14±4.95	88.83±0.65	86.39±0.64	83.43±0.26
UKAD-B	<u>68.22±5.04</u>	<u>90.60±1.02</u>	87.93±0.81	<u>83.60±0.28</u>
UKAD-L	**69.59±3.65**	**90.63±0.78**	**88.53±0.39**	**83.70±0.17**

Table 11. Dice (%, mean ± std over three splits) across published widths. We bold the best value and underline the second-best value.

Method	BUSI	CVC	GlaS	ISIC 2018
UNeXt-S	73.62±3.00	85.52±1.94	90.27±0.82	90.05±0.10
UNeXt-B	75.65±3.69	90.21±1.10	91.32±0.51	90.40±0.12
UNeXt-L	77.66±3.92	91.94±0.76	91.74±0.68	90.41±0.15
Rolling-UNet-S	78.63±3.48	93.02±0.59	92.43±0.58	90.67±0.15
Rolling-UNet-M	79.76±2.79	94.15±0.50	93.20±0.33	90.57±0.23
Rolling-UNet-L	79.78±3.96	94.72±0.50	<u>93.70±0.22</u>	90.43±0.24
UKAD-S	79.39±3.84	94.04±0.39	92.62±0.38	90.79±0.17
UKAD-B	<u>80.16±3.88</u>	<u>95.02±0.58</u>	93.51±0.45	<u>90.90±0.18</u>
UKAD-L	**81.10±2.54**	**95.03±0.44**	**93.88±0.24**	**90.99±0.13**

B.4 Held-out ISIC 2018 test set

Every ISIC 2018 model trains on the official training set. We select its checkpoint on a validation split of that set. We never use the 1000 images of the official ISIC 2018 test set for training or for checkpoint selection. We evaluate those same checkpoints on this test set. Overlap uses the metrics of Table 1, and boundary error uses the per-image HD95 of Table 2. UKAD-B has the highest IoU and Dice on this test set. AdaKAN has the lowest HD95.

B.5 Recall, precision, specificity, and Hausdorff distance

Tables 16 to 19 report every metric that Table 1 leaves out, on the same checkpoints. Table 2 in the main text reports per-image HD95. Table 14 gives the batch-level HD95 of the released evaluator for the same models. Table 15 counts the images with exactly one empty mask. Batch-level HD95 puts every boundary distance in the batch into one list and then keeps the 95th percentile. A bad image can sit above that percentile, so the score ignores it. HD keeps the largest distance, so that image remains in the score. On CVC-ClinicDB, UKAD-B therefore has batch-level HD95 0.75 and HD 20.91. Per-image HD95 scores each image on its own. For the same checkpoints that score is 4.23.

The rows cover every model in the main comparison plus UNeXt, Rolling-UNet, and UKAD at three widths each.

Table 12. Batch-level HD95 (pixels, lower is better, mean ± std over three splits) across published widths. We bold the best value and underline the second-best value.

Method	BUSI	CVC	GlaS	ISIC 2018
UNeXt-S	10.27±1.71	4.42±2.15	1.74±1.02	3.04±0.32
UNeXt-B	10.32±1.01	2.50±1.10	1.66±0.98	2.42±0.28
UNeXt-L	8.16±1.76	2.06±0.87	1.67±1.19	2.86±0.36
Rolling-UNet-S	8.15±1.85	1.49±0.66	1.60±1.31	2.22±0.28
Rolling-UNet-M	8.50±0.17	1.26±1.12	1.43±1.11	2.29±0.15
Rolling-UNet-L	8.76±2.63	1.01±1.02	1.41±0.99	2.99±0.46
UKAD-S	7.05±1.76	0.95±0.42	1.68±1.20	2.45±0.08
UKAD-B	**6.90±1.19**	**0.75±0.69**	**1.29±0.97**	2.23±0.22
UKAD-L	7.62±1.03	0.76±0.53	1.32±0.98	**2.13±0.33**

Table 13. Official ISIC 2018 test set, 1000 images, with the Table 1 checkpoints. IoU and Dice are in %, per-image HD95 is in pixels, and values are mean ± std over three splits. We bold the best value and underline the second-best value.

Method	IoU	Dice	HD95
U-Net	71.34±0.86	82.65±0.56	32.69±5.27
Attention U-Net	72.35±1.00	83.33±0.67	30.21±3.28
UNet++	72.80±0.60	83.70±0.44	28.27±0.78
UNeXt	76.50±0.84	86.19±0.51	20.66±1.77
MedNeXt	76.61±0.39	86.27±0.34	17.11±0.37
U-Mamba	76.11±0.71	85.87±0.49	17.97±0.39
nnU-Net ResEnc	74.61±0.79	84.94±0.55	20.59±0.98
Rolling-UNet	76.25±0.47	85.98±0.35	19.60±1.74
UKAGNet	71.64±0.69	82.86±0.47	30.79±2.86
U-KAN	76.84±0.34	86.46±0.23	19.36±1.94
U-FunKAN	76.40±0.62	86.21±0.44	18.37±0.57
CGLKNet	76.86±0.55	86.42±0.34	19.48±1.34
UKAN+	77.77±0.69	87.01±0.43	17.99±1.16
AdaKAN	77.83±0.44	87.15±0.29	**16.74±0.99**
UKAD-S	76.91±0.29	86.49±0.20	18.92±0.79
UKAD-B	**77.99±0.53**	**87.24±0.36**	17.76±0.81
UKAD-L	77.61±0.38	86.88±0.39	18.86±2.51

Table 14. Batch-level HD95 of the released evaluator (pixels, lower is better, mean ± std over three splits) for the Table 1 models. We bold the best value and underline the second-best value.

Method	BUSI	CVC	GlaS	ISIC 2018
U-Net	11.30±2.48	1.06±0.47	1.46±0.99	3.64±0.92
Attention U-Net	10.67±2.47	0.97±0.91	1.73±0.78	3.62±0.48
UNet++	10.10±1.90	1.11±0.78	1.42±1.20	3.52±0.70
UNeXt	10.32±1.01	2.50±1.10	1.66±0.98	2.42±0.28
MedNeXt	7.83±0.43	2.08±1.56	1.63±1.14	2.20±0.39
U-Mamba	9.73±3.81	2.30±2.48	1.65±1.07	2.28±0.44
nnU-Net ResEnc	9.49±3.17	3.17±1.48	1.68±1.11	3.52±0.73
Rolling-UNet	8.76±2.63	1.01±1.02	1.41±0.99	2.99±0.46
UKAGNet	13.50±1.98	2.45±1.69	1.65±1.15	3.82±0.34
U-KAN	7.11±2.38	1.29±0.55	1.66±1.12	2.47±0.42
U-FunKAN	10.18±3.16	3.53±1.29	1.80±1.13	2.80±0.83
CGLKNet	8.04±2.75	1.92±1.05	1.53±1.24	2.16±0.28
UKAN+	7.26±1.62	1.58±0.62	1.69±1.23	**2.06±0.44**
AdaKAN	7.03±2.66	1.14±0.84	1.49±1.12	2.28±0.09
UKAD-S	7.05±1.76	0.95±0.42	1.68±1.20	2.45±0.08
UKAD-B	**6.90±1.19**	**0.75±0.69**	**1.29±0.97**	2.23±0.22
UKAD-L	7.62±1.03	0.76±0.53	1.32±0.98	2.13±0.33

Table 15. Images with exactly one empty mask in the per-image HD95 of Table 2, summed over the three splits. The three validation sets hold 390 BUSI, 369 CVC-ClinicDB, 99 GlaS, and 1557 ISIC 2018 images.

Method	BUSI	CVC	GlaS	ISIC 2018
U-Net	14	2	0	6
Attention U-Net	11	2	0	5
UNet++	7	0	0	2
UNeXt	8	4	0	2
MedNeXt	0	0	0	0
U-Mamba	0	0	0	0
nnU-Net ResEnc	0	0	0	0
Rolling-UNet	3	1	0	0
UKAGNet	16	2	0	0
U-KAN	12	7	0	0
U-FunKAN	12	6	0	1
CGLKNet	0	3	0	3
UKAN+	5	1	0	0
AdaKAN	1	0	0	0
UKAD-S	12	1	0	1
UKAD-B	7	2	0	1
UKAD-L	3	1	0	0

Table 16. Recall (%, mean ± std over three splits). We bold the best value and underline the second-best value.

Method	BUSI	CVC	GlaS	ISIC 2018
U-Net	73.21±4.47	93.81±0.88	93.72±0.64	87.91±0.53
Attention U-Net	75.74±2.08	94.56±0.85	93.58±0.65	87.84±0.17
UNet++	76.56±3.43	94.14±0.67	92.95±1.24	87.76±0.55
UNeXt-S	74.13±4.51	84.26±2.51	90.97±1.55	89.31±0.09
UNeXt-B	73.53±5.81	88.56±0.74	91.96±1.16	89.49±0.40
UNeXt-L	75.76±4.76	90.67±0.95	92.82±1.02	89.50±0.61
MedNeXt	76.80±4.21	92.40±1.20	90.99±1.49	89.99±0.70
U-Mamba	79.19±5.10	93.18±0.60	89.55±2.22	90.55±0.72
nnU-Net ResEnc	75.79±3.55	87.05±0.71	89.27±2.32	88.18±0.59
Rolling-UNet-S	79.19±5.14	92.91±0.22	92.21±1.05	90.13±0.56
Rolling-UNet-M	76.90±4.30	94.02±0.40	92.74±0.95	90.80±0.59
Rolling-UNet-L	79.48±2.59	94.56±0.52	**94.31±1.24**	89.59±0.64
UKAGNet	68.97±1.91	90.86±1.25	91.70±0.78	87.01±0.29
U-KAN	77.12±1.94	92.38±0.46	93.13±1.07	89.62±0.36
U-FunKAN	75.27±4.82	86.20±1.51	87.80±1.57	89.34±0.97
CGLKNet	77.05±6.29	91.46±0.78	92.85±1.28	90.11±0.55
UKAN+	77.45±6.43	92.71±0.88	92.72±1.60	89.92±0.75
AdaKAN	80.32±3.28	94.32±0.52	92.98±0.70	90.45±1.01
UKAD-S	76.18±4.73	93.51±0.79	92.23±1.32	90.67±0.97
UKAD-B	78.42±3.48	94.90±0.68	93.26±0.78	**90.93±1.03**
UKAD-L	**80.79±5.12**	**95.01±0.85**	93.78±0.38	90.81±0.62

Table 17. Precision (%, mean ± std over three splits). We bold the best value and underline the second-best value.

Method	BUSI	CVC	GlaS	ISIC 2018
U-Net	81.53±1.35	94.36±0.77	93.03±0.18	90.77±1.05
Attention U-Net	80.29±4.23	94.73±0.69	93.38±0.50	91.47±0.41
UNet++	79.67±2.80	94.56±1.18	93.23±0.73	91.30±1.09
UNeXt-S	74.58±1.42	87.27±2.05	89.76±0.12	91.32±0.07
UNeXt-B	79.43±1.25	92.25±2.43	90.81±0.82	91.76±0.57
UNeXt-L	81.58±3.94	93.43±0.68	90.81±0.89	91.81±0.55
MedNeXt	78.73±3.84	93.60±0.88	90.88±1.15	91.58±0.82
U-Mamba	78.50±3.99	93.36±1.93	89.23±0.61	90.80±0.63
nnU-Net ResEnc	79.80±5.23	91.74±1.39	89.01±0.84	91.31±0.61
Rolling-UNet-S	79.42±1.85	93.32±1.01	92.76±0.21	91.70±0.54
Rolling-UNet-M	83.95±1.71	94.47±1.17	93.73±0.37	90.90±0.25
Rolling-UNet-L	81.44±5.52	95.02±0.93	93.14±0.78	91.86±0.26
UKAGNet	74.47±4.76	94.00±1.44	91.55±1.07	89.51±0.31
U-KAN	83.96±6.43	94.05±1.30	92.70±0.59	91.97±0.60
U-FunKAN	78.59±2.91	89.68±1.99	87.43±1.44	90.91±0.21
CGLKNet	81.58±1.70	93.78±0.59	92.87±0.66	91.91±0.14
UKAN+	81.42±4.39	93.65±1.39	91.15±0.81	91.81±0.49
AdaKAN	82.06±5.92	94.53±1.54	93.21±0.58	**91.99±1.00**
UKAD-S	**84.32±2.56**	94.72±0.57	93.11±0.65	91.46±0.83
UKAD-B	82.91±4.48	**95.26±0.61**	93.84±0.77	91.38±0.83
UKAD-L	83.05±2.02	95.18±0.78	**94.06±0.27**	91.62±0.49

Table 18. Specificity (%, mean ± std over three splits). We bold the best value and underline the second-best value.

Method	BUSI	CVC	GlaS	ISIC 2018
U-Net	98.39±0.17	99.42±0.08	92.52±0.57	97.62±0.26
Attention U-Net	98.21±0.38	99.47±0.07	92.92±0.67	97.86±0.11
UNet++	98.15±0.17	99.45±0.08	92.75±1.12	97.77±0.30
UNeXt-S	97.55±0.26	98.74±0.16	89.07±0.90	97.71±0.16
UNeXt-B	98.20±0.09	99.25±0.21	90.04±1.77	97.75±0.29
UNeXt-L	98.23±0.43	99.35±0.03	89.94±1.82	97.82±0.14
MedNeXt	97.91±0.54	99.38±0.04	90.48±0.75	97.77±0.20
U-Mamba	97.87±0.20	99.34±0.14	88.48±0.42	97.38±0.16
nnU-Net ResEnc	98.16±0.42	99.24±0.10	88.53±0.61	97.77±0.13
Rolling-UNet-S	97.99±0.15	99.33±0.05	92.38±0.33	97.74±0.24
Rolling-UNet-M	98.53±0.32	99.45±0.07	93.33±0.17	97.42±0.23
Rolling-UNet-L	98.13±0.58	99.50±0.05	92.55±0.92	**97.88±0.05**
UKAGNet	97.75±0.27	99.42±0.11	91.19±0.81	97.34±0.06
U-KAN	98.47±0.57	99.40±0.11	92.08±0.94	97.87±0.06
U-FunKAN	97.88±0.22	98.98±0.19	86.44±1.33	97.51±0.08
CGLKNet	98.27±0.33	99.39±0.04	92.31±1.33	97.76±0.15
UKAN+	98.27±0.34	99.35±0.22	90.39±0.66	97.78±0.17
AdaKAN	98.26±0.61	99.46±0.12	92.84±0.62	97.88±0.33
UKAD-S	**98.54±0.15**	99.48±0.02	92.77±0.63	97.64±0.31
UKAD-B	98.37±0.48	**99.52±0.02**	93.45±0.40	97.61±0.30
UKAD-L	98.14±0.16	99.51±0.05	**93.66±0.39**	97.63±0.19

Table 19. Hausdorff distance (pixels, lower is better, mean ± std over three splits). We bold the best value and underline the second-best value.

Method	BUSI	CVC	GlaS	ISIC 2018
U-Net	47.57±1.22	27.68±5.74	11.61±2.54	65.89±2.08
Attention U-Net	43.77±3.60	26.27±5.26	12.08±3.77	66.96±2.43
UNet++	45.58±2.62	27.47±8.56	11.84±2.12	67.30±2.29
UNeXt-S	47.92±7.71	42.87±6.84	13.97±2.80	63.52±4.37
UNeXt-B	43.32±3.32	32.89±3.50	11.39±2.17	63.09±0.69
UNeXt-L	45.09±3.33	31.75±8.73	11.77±1.79	62.99±5.26
MedNeXt	48.30±6.89	27.03±7.86	14.49±4.79	69.42±9.08
U-Mamba	42.94±6.04	29.08±5.26	13.85±3.92	59.72±1.46
nnU-Net ResEnc	54.37±17.46	38.86±0.98	14.45±4.57	69.31±3.95
Rolling-UNet-S	41.39±3.16	27.96±5.01	12.10±0.80	56.21±6.21
Rolling-UNet-M	39.24±2.24	26.54±6.19	10.82±1.94	56.48±4.05
Rolling-UNet-L	39.04±2.31	22.09±7.16	10.51±1.17	58.50±1.89
UKAGNet	53.99±2.56	31.10±3.48	11.50±1.24	77.02±9.38
U-KAN	39.04±6.81	30.55±3.08	11.53±1.58	61.85±1.60
U-FunKAN	42.13±5.63	34.75±5.05	12.26±1.36	57.52±4.81
CGLKNet	40.44±4.10	24.92±3.13	11.80±1.48	**51.76±2.97**
UKAN+	39.15±6.19	23.90±5.69	10.98±1.92	55.76±4.54
AdaKAN	37.14±3.10	21.67±6.20	11.74±3.00	53.95±2.75
UKAD-S	**36.80±1.42**	24.19±5.83	12.46±2.51	56.79±2.45
UKAD-B	38.12±3.31	20.91±9.56	**9.97±2.75**	56.06±4.37
UKAD-L	37.21±3.23	**18.86±5.02**	11.76±3.31	56.53±2.84

C PAIRED SIGNIFICANCE TESTS

We test the IoU difference between UKAD and each baseline in two paired ways. Both tests use only the trained checkpoints of Table 1. The primary test pairs the two models on each of the 12 matched dataset-split runs. It is a two-sided Wilcoxon signed-rank test, with Holm correction over the fourteen baselines. The supporting test reruns inference with the same checkpoints, computes IoU on each validation image, and pairs the two models on that image. It draws 10,000 bootstrap resamples inside each dataset-split stratum and averages the strata with equal weight. Table 20 reports both tests. The image-level test varies the images and holds the checkpoints fixed. Per-image IoU is a different aggregate from the batch-level MedPy IoU in Table 1.

Table 20. Full paired IoU matrix. The primary run-level comparison reports the difference of UKAD minus the baseline, wins, and Holm-corrected Wilcoxon p. Supporting per-image evidence reports the difference, 95% bootstrap interval, and p.

	UKAD-L			UKAD-B		
Baseline	ΔIoU	Wins	p_{Holm}	ΔIoU	Wins	p_{Holm}
U-Net	+3.14	11/12	0.007	+2.61	12/12	0.007
Attention U-Net	+2.50	11/12	0.007	+1.97	11/12	0.007
UNet++	+2.78	12/12	0.007	+2.26	12/12	0.007
UNeXt	+5.25	12/12	0.007	+4.73	12/12	0.007
MedNeXt	+3.78	12/12	0.007	+3.25	12/12	0.007
U-Mamba	+3.96	12/12	0.007	+3.43	12/12	0.007
nnU-Net ResEnc	+6.47	12/12	0.007	+5.95	12/12	0.007
Rolling-UNet	+0.86	11/12	0.007	+0.33	9/12	0.532
UKAGNet	+6.88	12/12	0.007	+6.35	12/12	0.007
U-KAN	+1.97	12/12	0.007	+1.45	12/12	0.007
U-FunKAN	+7.91	12/12	0.007	+7.39	12/12	0.007
CGLKNet	+2.57	12/12	0.007	+2.05	11/12	0.007
UKAN+	+2.73	12/12	0.007	+2.20	12/12	0.007
AdaKAN	+0.75	9/12	0.092	+0.23	7/12	0.569
Image level	UKAD-L: ΔIoU [95% CI], p			UKAD-B: ΔIoU [95% CI], p		
U-Net	+3.28 [+2.73, +3.83], <0.001			+2.76 [+2.23, +3.29], <0.001		
Attention U-Net	+2.29 [+1.80, +2.81], <0.001			+1.78 [+1.28, +2.28], <0.001		
UNet++	+2.84 [+2.32, +3.38], <0.001			+2.32 [+1.83, +2.82], <0.001		
UNeXt	+5.89 [+5.23, +6.57], <0.001			+5.37 [+4.73, +6.02], <0.001		
MedNeXt	+3.24 [+2.68, +3.82], <0.001			+2.73 [+2.14, +3.33], <0.001		
U-Mamba	+3.56 [+2.97, +4.16], <0.001			+3.04 [+2.47, +3.61], <0.001		
nnU-Net ResEnc	+5.95 [+5.22, +6.71], <0.001			+5.44 [+4.71, +6.18], <0.001		
Rolling-UNet	+0.63 [+0.19, +1.06], 0.004			+0.11 [−0.32, +0.55], 0.617		
UKAGNet	+7.15 [+6.43, +7.88], <0.001			+6.63 [+5.92, +7.38], <0.001		
U-KAN	+2.53 [+2.01, +3.08], <0.001			+2.01 [+1.50, +2.54], <0.001		
U-FunKAN	+8.50 [+7.73, +9.25], <0.001			+7.98 [+7.23, +8.76], <0.001		
CGLKNet	+2.35 [+1.79, +2.90], <0.001			+1.83 [+1.27, +2.40], <0.001		
UKAN+	+3.01 [+2.49, +3.52], <0.001			+2.49 [+1.95, +3.04], <0.001		
AdaKAN	+0.79 [+0.35, +1.23], <0.001			+0.27 [−0.20, +0.73], 0.253		

D COMPONENT EVIDENCE

The main paper shows paired IoU and HD95 effects for six components. Here we give the full set of arms, with IoU, Dice, and HD95 on the same three split manifests. The comparisons test the design choices we deployed.

D.1 WHAT EACH ARM CHANGES

Every arm below starts from UKAD-B and changes one thing. The *KAD block* arms remove the content-predicted displacement, the group-rational activation, both of those together, or the context branch defined in Section 3.4. The group-rational arm replaces ϕ with ReLU in the bottleneck, in the skip gates, and in the hybrid decoder. The *decoder block* arms replace every decoder block with one type. The default they replace is a residual convolution plus a gated KAN convolution at the two deepest levels. The *design choice* arms change one decision outside the bottleneck. Those decisions are a hard straight-through gate in place of the soft gate, a separate gate for the context

branch in place of the shared displacement gate, the adaptive-context mixer on decoder levels 3 and 2, the uncertainty-gated boundary head, the boundary loss term, the KAN skip gates, and the two deep-supervision heads.

We train every arm on all three splits of all four datasets.

D.2 Results on every metric

Tables 21, 23, and 24 give IoU, Dice, and HD95 for every arm. Tables 21 and 24 are the numerical source for the paired effects in Figure 4.

A raw change is large only relative to the split standard deviation of UKAD-B. That deviation is 5.04 IoU on BUSI and 0.28 IoU on ISIC 2018. One fifth of the BUSI deviation is 1.01 IoU. No BUSI arm reaches that size. The largest BUSI move is the independent context gate, at $+0.93$. On ISIC 2018, six arms move IoU by more than 0.28. The largest move in the table is the adaptive-context arm, at -1.04 IoU. Section 4.4 therefore pairs the splits instead of comparing these columns directly. On HD95, the arms separate more than they do on the IoU reported in Section 4.4. Removing the displacement and the group-rational activation together gives the largest BUSI HD95 in the table, 8.19 pixels against 6.90 for the default. Removing deep supervision gives the largest GlaS HD95, 2.05 pixels against 1.29 for the default. On IoU, deep supervision adds 0.68 on BUSI and loses 0.39 on ISIC 2018. Dice and IoU order the arms in the same way, except where the gap is small, and those exceptions are mostly on BUSI. The identity $\mathrm{Dice} = 2\,\mathrm{IoU}/(\mathrm{IoU}+1)$ in Appendix A.2 holds for one image. It does not hold after averaging over batches and splits. The main text reports only IoU, because Dice and IoU disagree only on gaps below 0.2 IoU.

The displacement and group-rational activation. The first four rows of Table 21 form a 2×2 over the two components of the KAD block. Table 22 rearranges that comparison. Averaged over the four datasets, the arm that replaces ϕ with ReLU reaches 82.69, and the arm that removes displacement reaches 82.67, against 82.59 for the default. For the ReLU arm, BUSI is 0.41 IoU above the default and the other three datasets sum to zero, so the mean gap is 0.10. We keep both components in the default.

Table 21. Ablation IoU (%, mean ± std over three splits). Each arm changes one UKAD-B component.

Arm	Params (M)	BUSI	CVC	GlaS	ISIC 2018
UKAD-B (Default)	7.77	68.22±5.04	90.60±1.02	87.93±0.81	83.60±0.28
KAD Block					
w/o Displacement	7.17	68.50±4.36	90.21±0.89	88.23±0.19	83.74±0.28
ReLU for every ϕ	7.77	68.63±5.46	90.57±0.58	87.69±0.63	83.87±0.31
w/o displacement, ReLU for every ϕ	7.17	67.93±4.06	90.30±1.47	87.86±0.48	83.86±0.26
w/o Context Branch	7.21	67.98±3.65	90.48±0.85	87.70±0.39	83.41±0.19
Decoder Block					
Convolution	7.34	68.11±5.27	90.48±0.93	88.23±0.19	83.54±0.14
MLP	5.88	68.73±4.95	90.33±0.22	87.53±0.48	83.58±0.38
KAN	5.26	68.12±4.89	89.74±0.47	87.65±0.83	83.27±0.49
Design Choice					
Hard Gate	7.77	68.57±4.89	90.72±0.79	88.05±0.64	83.49±0.55
Independent Context Gate	7.81	69.15±5.40	90.53±1.14	87.78±0.23	83.93±0.10
w/o Adaptive Context	7.74	68.11±4.15	89.82±1.18	87.84±0.62	82.56±0.07
w/o Boundary Head	7.76	67.92±3.58	90.24±0.71	88.05±0.45	83.51±0.12
w/o Boundary Loss	7.77	68.05±4.90	90.29±0.81	88.04±0.44	83.28±0.24
w/o KAN Skip Gates	6.64	67.75±3.91	90.57±0.76	87.59±0.38	83.28±0.49
w/o Deep Supervision	7.77	67.54±5.23	90.51±1.09	87.69±0.19	83.99±0.34

Table 22. Displacement and group-rational activation 2×2 (IoU, % mean over three splits). The final column averages datasets.

Displ.	Rational	BUSI	CVC	GlaS	ISIC	Mean
		67.93	90.30	87.86	<u>83.86</u>	82.49
✓		**68.63**	<u>90.57</u>	87.69	**83.87**	**82.69**
	✓	<u>68.50</u>	90.21	**88.23**	83.74	<u>82.67</u>
✓	✓	68.22	**90.60**	<u>87.93</u>	83.60	82.59

Table 23. Ablation Dice (%, mean ± std over three splits).

Arm	Params (M)	BUSI	CVC	GlaS	ISIC 2018
UKAD-B (Default)	7.77	80.16±3.88	95.02±0.58	93.51±0.45	90.90±0.18
KAD Block					
w/o Displacement	7.17	80.43±3.28	94.81±0.51	93.70±0.07	91.00±0.20
ReLU for every ϕ	7.77	80.47±4.13	95.02±0.33	93.37±0.36	91.08±0.22
w/o displacement, ReLU for every ϕ	7.17	80.03±3.15	94.81±0.90	93.46±0.25	91.07±0.18
w/o Context Branch	7.21	79.94±2.97	94.96±0.50	93.37±0.22	90.81±0.12
Decoder Block					
Convolution	7.34	80.03±4.37	94.98±0.51	93.69±0.08	90.88±0.09
MLP	5.88	80.52±3.91	94.90±0.12	93.29±0.25	90.89±0.25
KAN	5.26	80.16±3.74	94.55±0.30	93.36±0.43	90.72±0.31
Design Choice					
Hard Gate	7.77	80.50±3.87	95.10±0.45	93.59±0.35	90.84±0.34
Independent Context Gate	7.81	80.80±4.24	94.99±0.64	93.42±0.09	91.13±0.09
w/o Adaptive Context	7.74	80.08±3.38	94.56±0.69	93.46±0.30	90.28±0.02
w/o Boundary Head	7.76	80.13±2.73	94.80±0.42	93.59±0.22	90.84±0.08
w/o Boundary Loss	7.77	80.07±3.70	94.86±0.47	93.58±0.21	90.72±0.16
w/o KAN Skip Gates	6.64	79.86±3.21	95.02±0.44	93.32±0.17	90.71±0.32
w/o Deep Supervision	7.77	79.56±4.23	94.99±0.62	93.39±0.07	91.17±0.23

Table 24. Ablation batch-level HD95 (pixels, lower is better, mean ± std over three splits).

Arm	Params (M)	BUSI	CVC	GlaS	ISIC 2018
UKAD-B (Default)	7.77	6.90±1.19	0.75±0.69	1.29±0.97	2.23±0.22
KAD Block					
w/o Displacement	7.17	6.63±1.10	0.90±0.77	1.27±0.89	2.12±0.52
ReLU for every ϕ	7.77	6.97±1.09	0.73±0.47	1.49±1.11	2.07±0.08
w/o displacement, ReLU for every ϕ	7.17	8.19±2.48	1.10±1.38	1.48±1.09	2.41±0.28
w/o Context Branch	7.21	7.00±0.29	1.00±1.10	1.87±1.00	2.22±0.27
Decoder Block					
Convolution	7.34	7.40±0.98	0.56±0.25	1.61±0.78	2.35±0.08
MLP	5.88	7.27±0.98	0.62±0.04	1.48±1.08	2.21±0.49
KAN	5.26	7.54±0.41	1.07±0.89	1.40±0.93	2.46±0.28
Design Choice					
Hard Gate	7.77	6.22±1.12	0.41±0.08	1.37±1.01	2.73±0.76
Independent Context Gate	7.81	7.61±2.64	0.72±0.61	1.79±1.11	1.95±0.10
w/o Adaptive Context	7.74	6.55±1.55	1.11±0.85	1.65±0.98	2.43±0.43
w/o Boundary Head	7.76	7.07±0.78	1.04±0.90	1.47±0.79	2.07±0.04
w/o Boundary Loss	7.77	6.80±2.00	0.53±0.17	1.31±1.04	2.37±0.07
w/o KAN Skip Gates	6.64	7.16±0.65	0.41±0.13	1.36±0.90	2.74±0.33
w/o Deep Supervision	7.77	8.03±1.49	0.56±0.26	2.05±0.97	1.95±0.25

E Qualitative evidence and learned diagnostics

This section adds validation examples to the aggregate tables and records the fields learned by UKAD-B. The mask panels show the prediction errors. The displacement, activation, and gate panels describe the learned maps.

E.1 Qualitative cases

Figure 5 in the main text uses one validation case per dataset from split seed 2981. That figure shows error types. Figure 6 adds four validation cases per dataset. It uses the same baselines, the same filled-mask projection, and the same error legend.

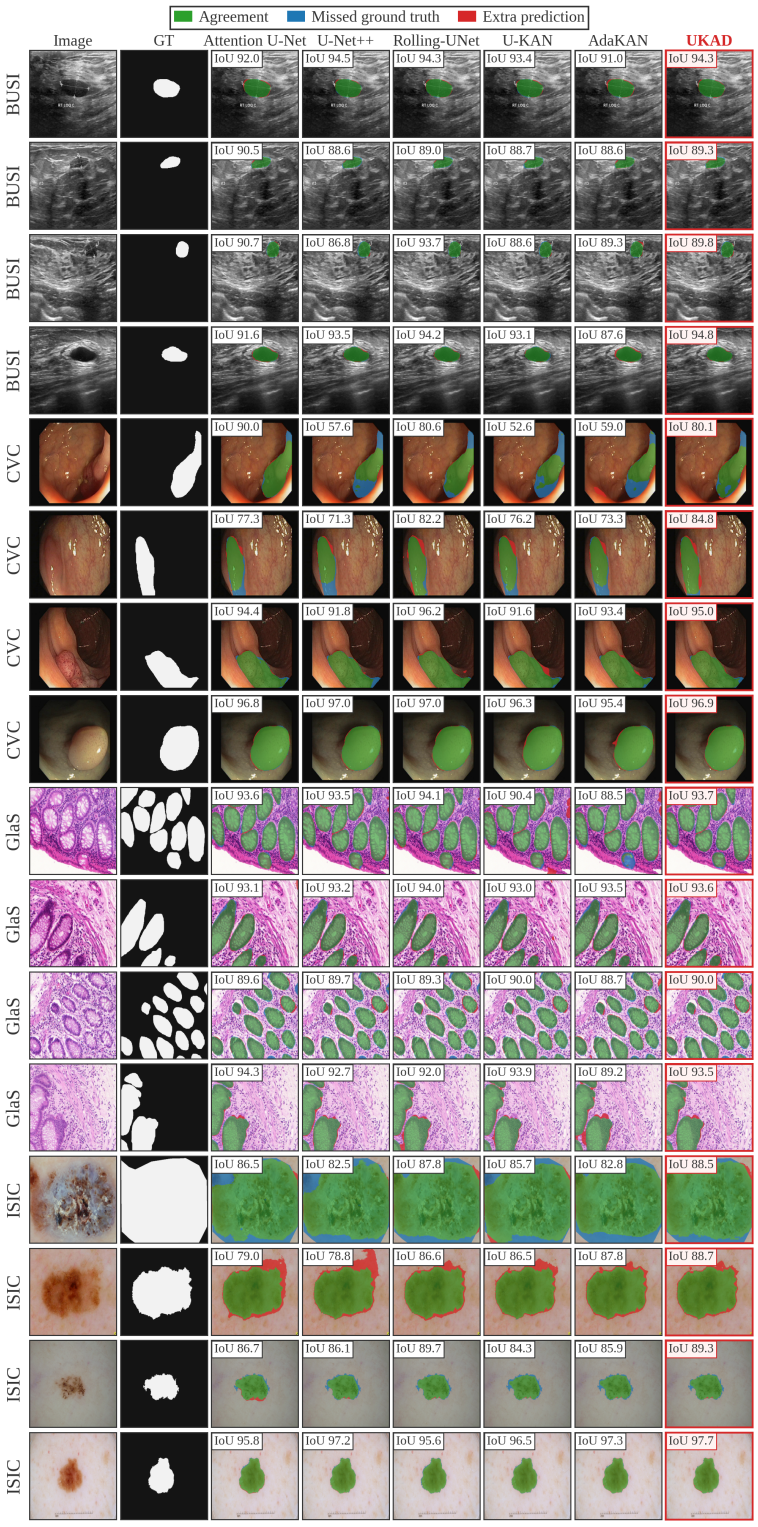

Figure 6: Additional validation cases from split 2981. Green marks agreement with the ground truth, blue marks missed foreground, and red marks a false positive. Each panel reports IoU (%).

E.2 DISPLACEMENT FIELD AND LEARNED ACTIVATIONS

Sections 3.3 and 3.2 define the displacement field and the two group-rational activations. Figures 7 to 9 show one validation case from each dataset. We export them from the UKAD-B checkpoint with the best validation IoU on split seed 2981. The displacement panels average (δ_x, δ_y) over the eight channel groups.

The fields are structured, including on uniform background. They are not boundary detectors that turn off away from an object. In these examples, the skip-gate maps α_4 and α_1 are higher near the object. α_4 is coarser because it operates at $H/8$. α_1 follows the object edge more closely. These panels record the learned maps.

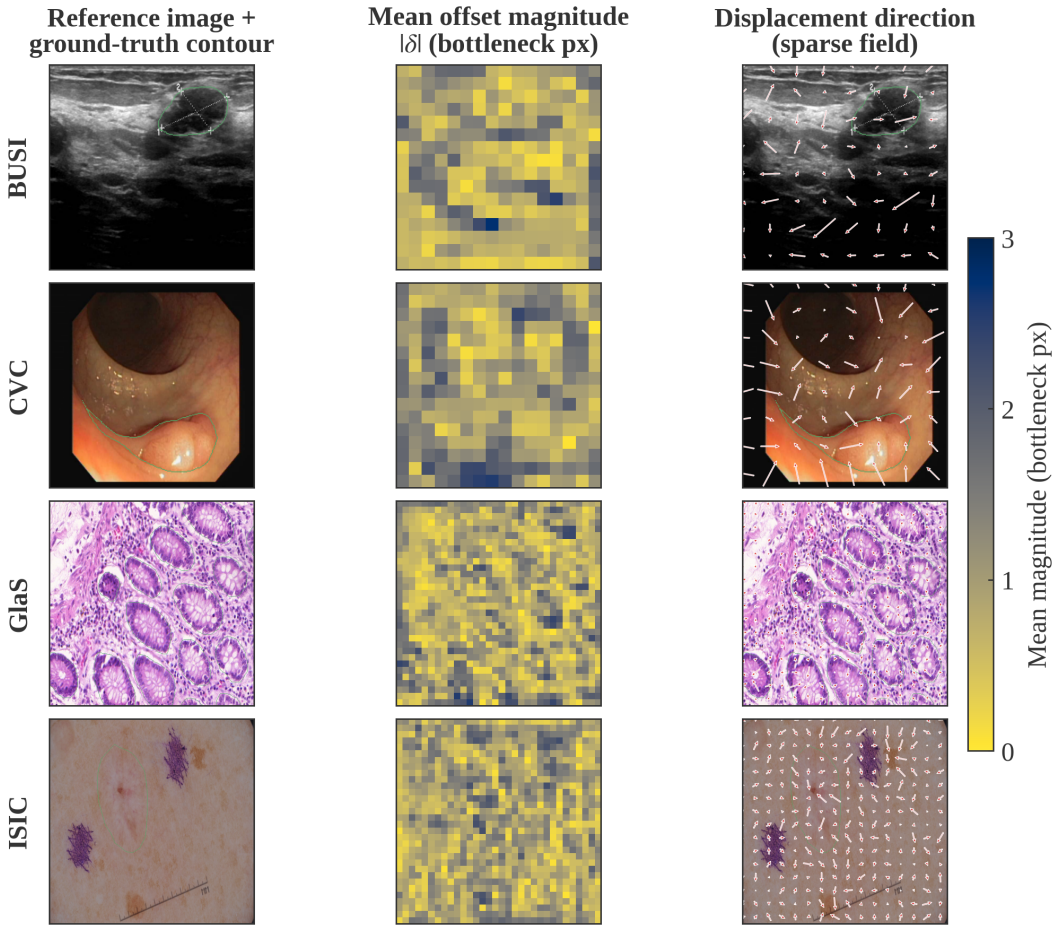

Figure 7: Predicted bottleneck displacement on one split-2981 case per dataset. Green marks ground truth, and we average the magnitude over channel groups on a shared $[0, 3]$ scale.

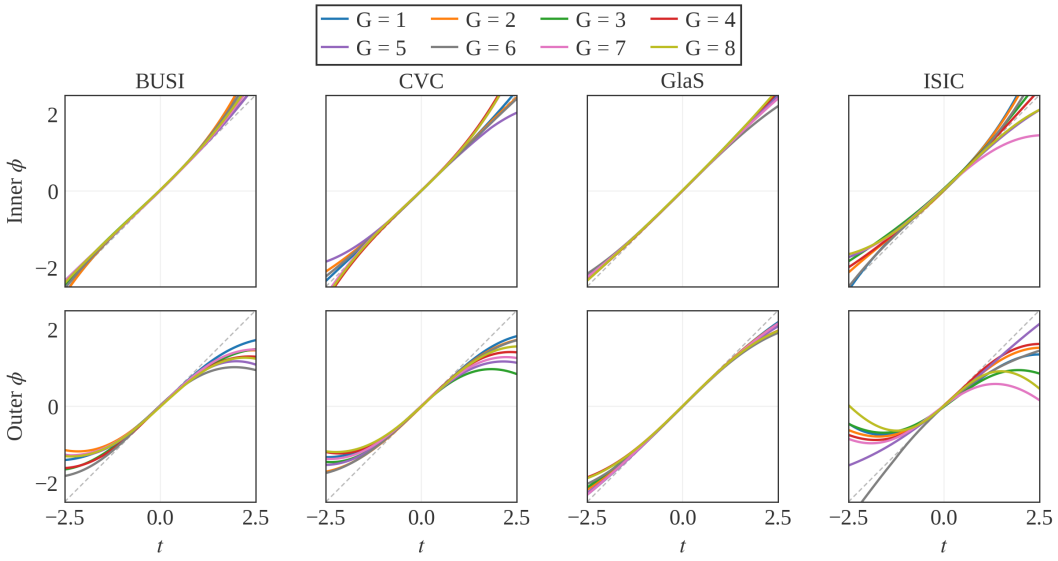

Figure 8: Learned UKAD-B bottleneck activations on split 2981. The top row shows the inner ϕ after displacement, and the bottom row shows the outer ϕ after the 1×1 mix. The dashed line is the identity.

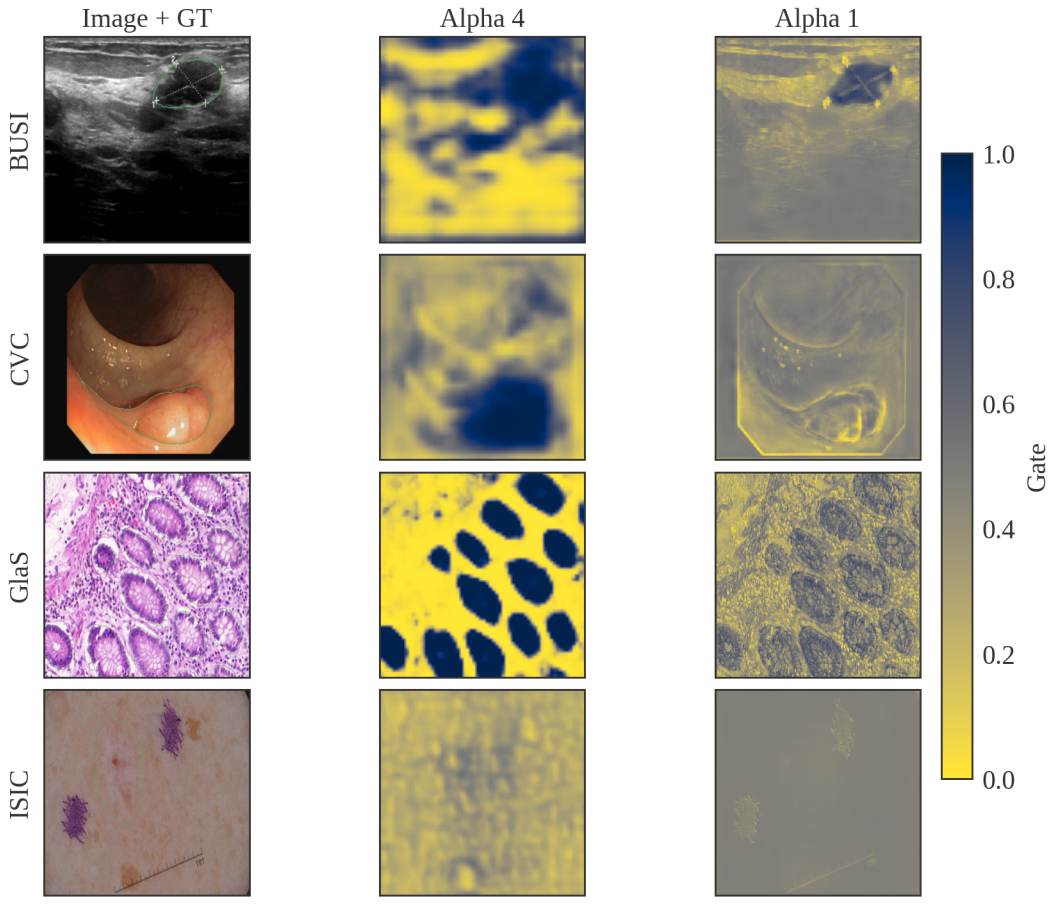

Figure 9: UKAD-B KAN skip-gate maps for the Figure 7 cases. α_4 is deepest and α_1 full resolution, and darker blue denotes a higher value.

F INTERPRETATION

F.1 HOW TO READ THE EVIDENCE

CVC-ClinicDB. CVC-ClinicDB is where UKAD-B separates most from the KAN baselines. Displacement helps on all three splits of that dataset.

The hybrid decoder. Table 21 shows that a convolutional decoder ties the hybrid decoder on mean IoU and on mean Dice. We select the decoder on the validation split. U-KAN, UNeXt, Rolling-UNet, and U-FunKAN also select their reported models on that kind of split (Li et al., 2025; Valanarasu & Patel, 2022; Liu et al., 2024a; Penkin & Krylov, 2025). Of the two options that keep ϕ in the decoder, the hybrid decoder is higher by 0.39 mean IoU than the KAN-only decoder. The convolutional decoder costs 0.43 M fewer parameters.

What the diagnostic figures show. Appendix E.2 shows a structured displacement field, an inner map near the identity, and skip gates that rise on the object. The fields are structured and correlated with tissue. We read the figures as a record of what the block converged to.

F.2 FURTHER COMPARISONS

The displacement sits only at the bottleneck, with $r_{\max} = 3$ pixels at $H/16$. Table 21 shows that a hard straight-through gate and an independent context gate are both slightly higher than the default on the four-dataset mean. We keep the default gate. Displacement helps on CVC-ClinicDB.